META-HEURISTIC PROCEDURES FOR THE MULTI-RESOURCE LEVELING PROBLEM WITH ACTIVITY SPLITTING

A THESIS IN ENGINEERING SYSTEMS MANAGEMENT
Masters of Science in Engineering Systems Management, Engineering Management Option

Presented to the faculty of the American University of Sharjah
College of Engineering
in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by
HADEEL YACOUB ALSAYEGH
B.Sc. 2004

Sharjah, UAE
April 2011

We approve the thesis of Hadeel Yacoub Alsayegh — Date of signature

Dr. Moncer Hariga
Professor
Engineering Systems Management Graduate Program
Thesis Advisor

Dr. Fouad Ben Abdelaziz
Professor
Engineering Systems Management Graduate Program
Graduate Committee

Dr. Hazim El-Baz
Associate Professor
Engineering Systems Management Graduate Program
Graduate Committee

Dr. Saleh Abu Dabous
Assistant Professor
Civil Engineering
Graduate Committee (External)

Dr. Moncer Hariga
Director
Engineering Systems Management Program Director

Dr. Hany El-Kadi
Associate Dean
College of Engineering

Dr. Yousef Al-Assaf
Dean
College of Engineering

Dr. Gautam Sen
Vice Provost, Research & Graduate Studies

META-HEURISTIC PROCEDURES FOR THE MULTI-RESOURCE LEVELING PROBLEM WITH ACTIVITY SPLITTING

Hadeel Alsayegh, Candidate for the Master of Science Degree

American University of Sharjah, 2011

ABSTRACT

The proper utilization of resources is important to achieve project success. In project management, there are two types of resource scheduling problems. The first is resource allocation in which activities are scheduled depending on the availability of limited resources to ensure that resource limitations are not exceeded in any period. The second type is resource leveling which includes moving non-critical activities within their float to improve the resource profile while not extending the project's duration.

Based on the review of related literature, resource leveling techniques can be grouped into three categories: heuristics, optimization and meta-heuristics. Most resource leveling techniques assume that activities cannot be split, meaning that once an activity starts, the work continues until the activity is completed. Activity splitting may be needed to improve resource utilization. Even with the few previous methods that incorporated activity splitting, resource leveling was accomplished using optimization techniques, which are not efficient for large size projects. A more computationally efficient approach to solve larger projects is to use meta-heuristic procedures such as Particle Swarm Optimization (PSO) and Simulated Annealing (SA). The proposed resource leveling technique is developed using Particle Swarm Optimization combined with Simulated Annealing, which assumes a time constrained project, with unlimited resources and allows for the splitting of non-critical activities.

Since there are no benchmark problems available in the literature, a set of 180 test problems are created and used as a benchmark to test the proposed methods. An optimization model is then used to determine the exact solution for these benchmark problems. Next, six PSO heuristic procedures are developed and assessed using the 180 benchmark problems. The results of these procedures are then analyzed based on the percentage difference in cost and the computational time. From the analysis, it was observed that the heuristics are becoming trapped in local optimum and are unable to find optimal solutions. Hence, the six heuristic procedures are combined with Simulated Annealing, which searches for new solutions without being trapped in local optimum, and are assessed using the benchmark problems. PSO-SA Procedure 3, which is based on Quantum theory, generated the best results with an average of 4.23% cost difference between the generated and the optimal results. Moreover, 147 out of the 180 problems had a percentage cost deviation of less than or equal to 10%. As for the computation time, the heuristic procedures generated solutions with an average reduction of 7 times for the large size problems. Furthermore, the proposed heuristic is assessed for larger problems in which a near optimum solution is reached within 25 minutes, unlike the optimal procedure which takes longer than 24 hours.

This research is an important additional step in the ongoing research on resource leveling. The proposed heuristic procedure offers several improvements over the current resource leveling techniques. The proposed procedure allows for activity splitting, which is more realistic and results in better resource profile. The new procedure takes advantage of combining Particle Swarm Optimization with Simulated Annealing to reach the optimum or near optimum solution in a short time period. The proposed procedure allows planners to consider the tradeoff between the cost of activity splitting and the cost of resource fluctuations resulting in minimum overall project cost.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Moncer Hariga for his support, intellectual advice and guidance not only during this research but also throughout my graduate studies. I have learned a lot under his tutelage both academically and professionally.

I would like to thank Dr. Fouad Ben Abdelaziz, Dr. Hazim El-Baz, and Dr. Saleh Abu Dabous for serving on my committee and spending their precious time in reviewing my work and providing inspiring and valuable suggestions. Moreover, I would like to thank the ESM faculty for teaching me during my graduate studies.

I would like to thank my family for their never ending support and love and in particular, my husband for being there for me and encouraging me.

DEDICATION

This thesis is dedicated to my husband, Sameh, my two children, Chantelle and Shadi, my parents, Yacoub and Rita, and my brother, Jeries, and sisters, Rola, and Rawan. Thank you for your continuous support, encouragement, and unconditional love.

Chapter 1: Introduction

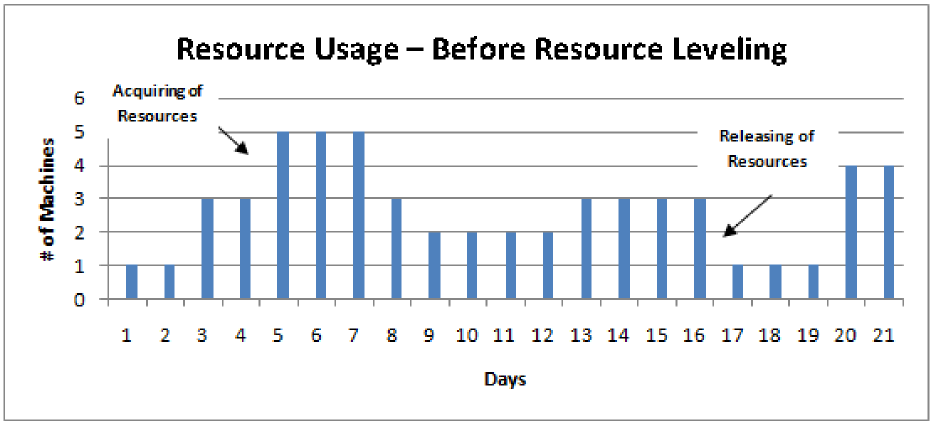
1.1 Overview

A project is formally defined as a temporary endeavor undertaken to create a unique product or service, whereas project management is the application of knowledge, skills, tools, and techniques to manage project activities in order to complete the project on time, within budget, and meet or exceed stakeholders' needs and expectations from the project [1, 2]. In today's challenging and competitive environment, time is very critical in making products available in the market ahead of competitors. Therefore, the need to properly plan and schedule projects is vital for their successful completion.

Project planning involves identifying the activities needed to complete the project and the relationships among them. In other words, determining what needs to be done and how. Project Scheduling, on the other hand, is concerned with determining the duration of each activity along with its starting and completion time as well as the project duration. One of the most traditional project scheduling techniques used in project management is the Critical Path Method (CPM), which identifies the critical path(s) of the project, by calculating the Early Start (ES), Early Finish (EF), Late Start (LS) and Late Finish (LF) times and the slack (float) of each activity. The critical path consists of activities that may not be delayed without delaying the project, and are thus known as critical activities. However, activities that can be delayed without affecting the project duration are known as non-critical activities.

Projects rely on resources in completing their activities. Examples of resources include manpower, machines, money, and materials. While scheduling the project activities using CPM, resources are assumed to be unlimited, which is not the case in most practical situations. Ignoring the availability constraints on resources while scheduling projects, could result in unrealistic schedules that cannot be achieved.

There are two categories of project scheduling problems: resource-constrained and time-constrained. With resource-constrained projects, the objective is to schedule project activities so that a particular resource does not exceed a specific limit in any given project time period, while holding the project duration to a minimum. This method is also known as resource allocation. Resource allocation usually results in extending project duration. This is especially true if the needed resources for critical activities are being utilized by non-critical activities. Some resource allocation techniques allow for the non-critical activities to be interrupted so that the needed resources are reallocated to the critical activities. However, even with allowed splitting, if the required resources are not available, extension of project duration may occur. Many optimization approaches exist for solving resource-constrained problems, some of which include Integer Programming, Dynamic Programming, and Heuristic Programming [3]. On the other hand, time-constrained projects assume that time is constrained while resources are available in unlimited quantities. Their main objective is to optimize the utilization and variation of the resources. Resource leveling is a technique used to minimize the change of the resource requirements from one period of time to the next; in other words, to minimize the peaks and valleys of resource usage. Resource leveling is accomplished by moving non-critical activities within their float. Figure 1 shows two histograms of resource utilization before and after resource leveling. After resource leveling, it can be noted that a smoother resource profile is achieved, with a gradual increase and gradual decrease in the resource usage. A more desirable distribution, but difficult to get, is the uniform distribution where the resource requirements are fixed throughout the project duration.



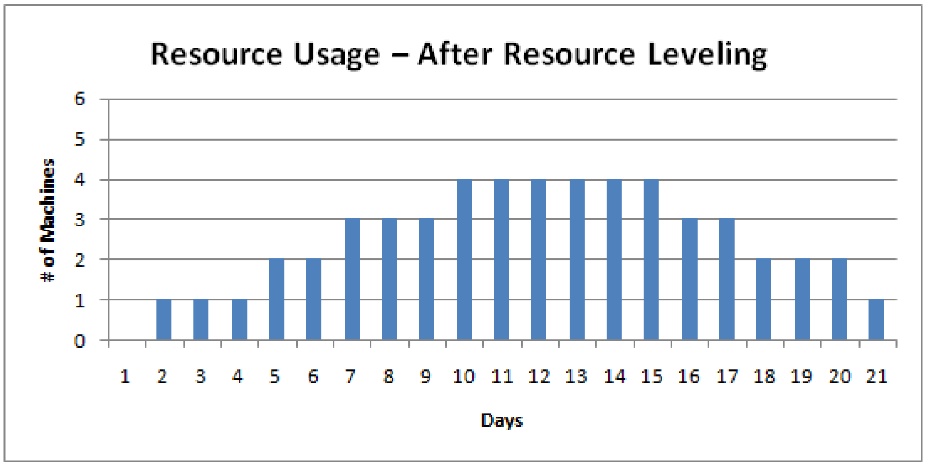


Figure 1: Resource Usage

Most resource leveling techniques assume that the activities may not be split, indicating that once an activity starts, it will continue until the work is finished. There are two main reasons behind this assumption. First, the non-splitting of activities make the mathematical computation easier and secondly, it is less costly. However, in reality, there are many projects in which the activities can be split. For example, a programmer may need to stop working on his/her current task to perform a more urgent task such as bug fixing, and then return to his/her current task once done. Another example, from the construction industry, is that during a construction of a house, the labor are working on the sidewalk are shifted to complete the fixing of the ceiling, as it is a critical activity. The workers can go back to the sidewalk once the critical activity is finished. These examples illustrate how non-critical activities are stopped to complete critical path activities.

Resource leveling with activity splitting is more mathematically complex as it introduces more decision variables and constraints [4]. Moreover, two types of costs are involved: resource dependent costs and activity dependent costs. Resource dependent costs are related to the resource itself and involve the costs of acquiring and releasing a resource from one period to the other. Activity dependent costs are costs related to the starting and stopping of the activity, such as the moving of equipment. In addition, upon splitting an activity, the learning process of the resources is affected, and it will take some time for the resources to re-achieve the learning level just prior to splitting the activity. Figure 2 shows an activity before and after splitting. The activity is split twice basically dividing the activity into three segments. For each of the two splits, there are associated stopping and starting cost.

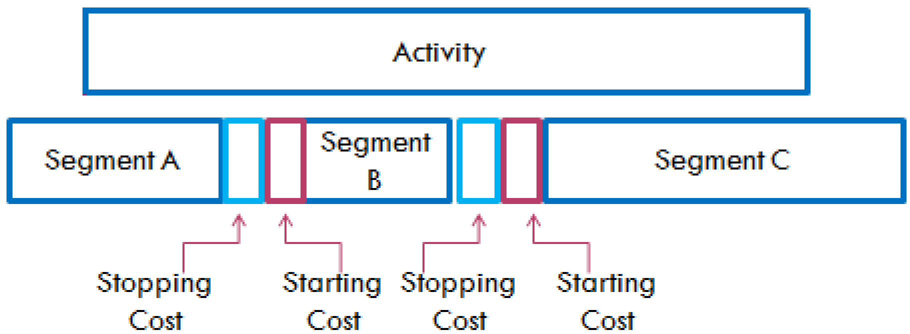


Figure 2: Activity Splitting

Although there is plenty of research done on resource leveling without splitting, very little research is done on resource leveling with activity splitting. Hariga and El-Sayegh [4] came up with a model based on a mixed integer linear programming formulation. Their model generates exact solutions to such problems; however, it is not efficient for large size projects. A more computationally efficient approach to solve larger projects is to use meta-heuristic procedures. Some examples of meta-heuristic approaches include Genetic Algorithms, Ant Colony, and Particle Swarm in solving such problems. Using meta-heuristic approaches, an optimum or "near optimum" solution is found rather than an exact solution as with exact optimization procedures. This thesis presents a new procedure for resource leveling that allows for activity splitting and uses Particle Swarm

Optimization and Simulated Annealing techniques to search for the optimum or near optimum solution.

1.2 Overview of Resource Management Categories

The techniques used to manage resources, either to allocate resources or to level the utilization of resources, are divided into three categories: heuristics, optimization, and meta-heuristics techniques. In the subsequent sub-sections, the first two categories are presented briefly, but the focus is more on the meta-heuristics since this thesis presents a resource leveling technique using two meta-heuristic approaches.

1.2.1 Heuristics

A heuristic method consists of a set of procedures that rely on common-sense ideas of how to search for fine solutions. These procedures are iterated several times until the best solution is found, hence heuristics are also known as “iterative algorithms”. Heuristic methods are procedures that are likely to discover very good feasible solutions, but not necessarily optimal solutions [5]. Each heuristic is designed for a particular type of problem. If the heuristic is well-designed, then a nearly optimal solution is obtained. Most heuristics are efficient enough to solve very large projects with numerous activities. Although heuristic methods can handle very large projects, the solutions they provide are not necessary optimum [6].

1.2.2 Optimization

Optimization techniques are techniques that rely on linear/nonlinear programming methods to determine the best outcome, where in resource management the outcome is the least cost (for resource leveling) and the shortest duration (resource allocation). An optimization technique usually consists of an objective function, which is used to minimize or maximize a variable (cost or duration) while satisfying one or more constraints. Unlike heuristics, optimization techniques provide optimum solutions; however they are limited by the project size. In other words, as the size of the problem increases, by adding more activities for example, the number of calculations also increase; and therefore, consuming a lot of time to reach the best outcome.

1.2.3 Meta-Heuristics

Complex combinatorial problems, such as the travelling salesman problem, the vehicle routing problem, the capacitated facility location-allocation problem, the resource constrained project problem, the problem addressed in this research work, and many other problems encountered in engineering economics, and industrial fields, are very hard to solve using exact optimization procedures. In fact, many of these problems are known as NP-hard, meaning that it is very unlikely to develop an algorithm capable of solving them in polynomial time.

The techniques used to solve complex combinatorial problems can be classified as either exact or approximate procedures. The former technique attempts to generate an optimal solution, which is proven to be indeed the global optimal. The branch and bound and dynamic programming are just two examples of exact procedures. However, for many problems, these procedures are computationally inefficient as the time required to obtain the optimal solution grows exponentially with the problem size. On the other hand, approximate procedures produce solutions in shorter computation time but are not guaranteed to be optimal. There are two main classes of approximate procedures: constructive and local search algorithms. The first class of algorithms, such as greedy constructive heuristics, constructs solutions from scratch by adding components to the solution one by one, whereas local search algorithms iteratively improve the current solution by moving to hopefully better neighboring solutions. The main drawback of the latter procedures is the likelihood of being stuck in local optimal solution.

Recent advances in optimization methods have shifted research attention to the development of general solution procedures, called meta-heuristics, to further improve the solution quality of local search heuristics. Because of their general structure, they can fit different kinds of optimization problems with relatively few modifications. They are based on concepts from different fields such as genetics, biology, artificial intelligence, social science, physics, and neurosciences, among others. The use of such concepts helps to create some degree of randomness in searching for optimal or near optimal solutions for well-known hard problems. Examples of common meta-heuristics include Tabu Search, Simulated Annealing, Genetic Algorithms, Ant Colony Optimization, and

Particle-Swarm Optimization. Each of these algorithms is discussed briefly in the following section; however the Particle-Swarm Optimization and Simulated Annealing are discussed more thoroughly in chapters 4 and 5, respectively.

1.2.3.1 Tabu Search

The Tabu Search (TS) is a meta-heuristic search procedure, which was initially proposed by Fred Glover in 1986. It has achieved many practical successes when applied to applications such as scheduling, routing and graph coloring [7]. The meta-heuristic maintains a tabu list with a maximum size L to keep track of the recently obtained candidate solutions. If the maximum size of the list is reached, then the oldest candidate is removed to make space for the recently improved solution. The search halts once a fixed number of iterations is reached or when no other improvement can be reached.

1.2.3.2 Simulated Annealing

Simulated Annealing (SA) is a probabilistic based search meta-heuristic to locate a good solution to a global optimization problem without being trapped in a local optimum. The concept of the simulated annealing is based on the analogy between the simulation of the annealing of the solids and the problem of solving large combinatorial optimization problems [8]. Annealing refers to the process in which the particles of a solid are randomly arranged once the solid turns to liquid at high temperatures. The procedure begins with an initial solution which is first considered as the best solution. Next, the procedure iterates until it finds a candidate solution. A solution is considered as a candidate if it is either a better solution than the one found so far or the probability of accepting it is high. This probability relies on a variable "T" which represents the temperature in an annealing process. The higher the value of T, the more chances a solution is accepted (i.e. more randomness).

1.2.3.3 Genetic Algorithms

The Genetic Algorithm (GA) technique was developed by John Holland in the 1970s. GAs are stochastic search techniques based on the natural phenomenon of "theory of evolution" formulated by Charles Darwin in the mid-19th century [5]. This phenomenon is also referred to as the "survival of the fittest". In genetic terms, each parent is represented by a chromosome. A chromosome consists of a set of genes (traits).

When two chromosome parents merge, using a crossover operation, an offspring chromosome is reproduced. During merging, some chromosomes are modified by a mutation operation. Offspring usually inherit the more fit (better) genes from each parent. Fitter chromosomes are more likely to be inherited in the next generations. A population of chromosomes is produced overtime.

Genetic algorithms are popular in the areas of optimization, scheduling, and transportation, among others. Thus, genetic algorithms can be used for resource leveling and resource allocation. A chromosome is represented by a possible sequence of activities, taking into consideration the precedence relationships among activities (i.e. all preceding activities of an activity must appear prior to the activity itself in a chromosome representation). When two parent chromosomes are merged, by randomly choosing activities from each parent, taking into consideration the precedence relationship, an offspring chromosome is reproduced. If the main objective for using genetic algorithms is to reduce costs when leveling resources, for example, then a cost is linked to each activity. The sub-sequence of activities that denote a lesser cost are more fit, to be inherited by the children. Therefore, the final result is an optimum or near-optimum chromosome (sequence of activities).

1.2.3.4 Ant Colony Optimization

Ant Colony Optimization (ACO) is a random based parallel search procedure, which was proposed by Marco Dorigo in 1992 in his PhD thesis. ACO algorithms are designed to solve specific types of combinatorial optimization problems [9]. They inspired by the behavior of real-life ant colonies that search for food by finding optimal paths.

One main feature of ants is that they leave “pheromone trails” when searching for food. Pheromone is a chemical substance that ants deposit along their way from their nest (source) to the food source (destination). When other ants are looking for food, they often look for smell pheromone paths in hope to find some food. As more ants traverse the same path, the concentration of the pheromone tends to increase. The smell of the pheromone substance tends to fade with time, thus leaving only the most traversed paths that are more probable to be chosen.

1.2.3.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithms, which were presented by Kennedy and Eberhart in 1995 [10], are based on the social behaviors of animals and insects, such as bird flocks and fish schools. Swarms, or groups, of these animals and insects, tend to self-organize themselves in optimal spatial patterns. Their behaviors, such as speed and direction, are determined based on a small number of neighboring individuals [9]. PSO algorithms conduct their search randomly by using a population (swarm) of individuals (particles) to find an optimum or near-optimum solution for hard optimization problems [11]. Each particle is a potential solution that consists of two parts representing its position in a N-dimensional space and its velocity.

The PSO was first developed to search for solutions in real space, in which the positions of the particles were represented as real numbers. However, in 1997, Kennedy and Eberhart [12] introduced a discrete binary version of the PSO where the positions of the particles held only binary values, 0 and 1. Also, the velocities of the particle represent the probability that the binary bit of the particle will change its value to one, and is restricted to [0, 1]. Therefore, a normalization function may be required to change the continuous values of the velocities to [0, 1].

1.3 Statement of the Problem

Most resource leveling techniques assume that activities are continuous and may not be split. Activity splitting may be desirable to smooth the profile of the resource utilization. In addition, most research done up until now, propose resource leveling approaches aiming at finding optimal solutions using exact optimization approaches. However, the more complex a project is and the more activities it has, the higher the computational time. Therefore, there is a clear need for a new resource leveling approach which is based on meta-heuristics, that is computationally efficient, to cater for activity splitting and handle large size projects.

Previous researches have shown that meta-heuristics algorithms are efficient in solving large size combinatorial problems in terms of computation time and quality of their solution. Therefore, the main objective of this research is to a design meta-heuristic approach based on the combination of Particle Swarm Optimization and Simulated

Annealing to generate a near optimal project schedule for the resource leveling problem with activity splitting.

1.4 Objectives

The objective of this research is to develop a meta-heuristic approach for resource leveling that is computationally efficient in finding near optimum solutions for large size projects. The proposed approach assumes a time constrained project, with unlimited resources and allow for the splitting of non-critical activities. Furthermore, the proposed approach is expected to achieve a reduction in the computational time for large size projects.

1.5 Significance

The main contributions of this research include the following:

1. Supplementing the project scheduling literature with a new resource leveling technique that can be implemented on large projects with large number of activities. This research work is the first to address heuristically the resource leveling problem with activity splitting with the objective of minimizing the cost by minimizing the fluctuations of the resources.
2. Developing heuristic procedure that reduces the computational time needed to perform resource leveling.
3. Allowing for activity splitting will ensure a better resource profile that is less costly overall.
4. Minimizing the overall project cost and reducing resource fluctuations over time.
5. Studying the possibility of linking the developed technique with other scheduling software will give better results and increase the chances of usage.

1.6 Methodology

The following steps are undertaken to achieve the research objectives:

Step 1: Review the literature related to meta-heuristic procedures with more focus on Particle Swarm Optimization.

Step 2: Develop benchmark problems to be used in the empirical computational analysis

As there are no benchmark problems available in the project scheduling literature, a set of test instances are developed to assess the solution quality of the proposed heuristics. The test problems are generated by varying the number of non-critical activities, *nn*, and number of resources, *P*. For each combination of nn ε {2, 4, 7, 8, 9, 10} and P ε {2, 4, 6}, 10 instances with different problem parameters (resource utilization rates and costs) are created. The 180 problem instances are then solved using the exact procedure proposed by Hariga and El-Sayegh [4]. The obtained optimal solutions are later used as benchmarks to evaluate the cost performance of the heuristic procedures.

Step 3: Develop and code several heuristic procedures

Six different heuristic procedures are developed. The heuristic procedures are based on Particle Swarm Optimization (PSO) and Simulated Annealing (SA) to generate near optimal solution with less computation efforts. The PSO procedures are adapted to handle binary variables as PSO is designed for continuous optimization.

Step 4: Determine the proper parameter settings for PSO and SA

This step is concerned with the tuning of the PSO and SA parameters. In other words, different values for these parameters are attempted to decide on the best ones to be employed in the assessment of the performance of the proposed procedures. As a starting point, the values found to be good in the application of PSO to other types of applications are used as the initial solution. In addition, other parameter settings are tested using different empirical experiments to ensure that the appropriate value for each parameter is used.

Step 5: Conduct computational experiments to validate the proposed heuristic procedure

The generated benchmark problems are used to assess the quality of each proposed heuristic in terms of computational time and cost performance.

Step 6: Select the best heuristic procedure

1.7 Thesis Organization

Chapter one presents the introduction to the research problem and the objectives. Chapter 2 presents the summary of the related literature review. Chapter 3 presents the test problems and the optimum solution using Hariga and El-Sayegh [4] model. Chapter 4 presents the development of six Particle Swarm Optimization models. In chapter 5, each of the PSO models presented in chapter 4 are combined with Simulated Annealing in order to further improve the results. Finally, chapter 6 concludes the thesis with the conclusion and recommendations for application.

Chapter 2: Literature Review

2.1 Introduction

Managing resources is important to the successful execution of projects as the project schedule can only be implemented after assigning the required resources for each activity of the project. Gray and Larson [13] emphasized that “project network times are not a schedule until resources have been assigned”. In most projects, resources are limited and even if they are not limited, there is still a need for their proper utilization. Resource management mainly focuses on two areas: resource allocation and resource leveling. While the former deals with the allocation of limited resources, the latter is concerned with using the required resources efficiently given fixed project duration [14].

Resource allocation refers to scheduling project activities based on the availability of resources. If the needed resources are not available, this will result in delaying that activity. For critical activities, that means the project completion time will be extended. The objective of resource allocation is to determine the minimum project duration given the resource constraints. In this category, time is assumed to be flexible and the project duration may be extended. Hegazy [15] explained that scheduling with limited resources is mathematically difficult as it is considered a large combinatorial problem.

In reality, resources are not limited, but rather adding more resources is costly. Companies are usually constrained by the project’s duration and may not be in a position to extend the project’s duration [14]. Resource leveling is concerned with the efficient utilization of resources and assumes that resources are unlimited but the project duration is fixed. Resource leveling tends to reduce the peak demands for a given resource and minimize the fluctuations in resource usage over time. Fluctuations in resource demands require extensive hiring, training, firing, moving and starting-up which is costly. Furthermore, newly hired resources are also considered as less efficient.

In the following sections, previous resource management techniques are reviewed. Resource management techniques can be grouped in three categories: Heuristics, Optimization and Meta-Heuristics.

2.2 Heuristic Procedures

One of the famous techniques of resource leveling which is developed by Harris in 1978 is the “Minimum Moment” algorithm [16]. In this technique, activities are assumed to be uninterrupted; once an activity starts, it continues until it is completed. In addition, the technique assumes that the resource consumption is constant over the duration of the activity and only one activity can be leveled at a single time. Once the project is scheduled through CPM, for example, only the non-critical activities are leveled by shifting them using their free floats. The algorithm focuses on an improvement factor which is used for the basis of leveling decisions. The improvement factor is calculated for several activities, but the activity with the largest improvement factor is chosen to be leveled.

Hiyassat [17] proposes a modification to the “Minimum Moment” approach presented by Harris [16] which uses heuristics for resource leveling. The modified method, like the traditional approach, assumes that activity splitting is not allowed, project duration is limited (time-constrained) and the resource availability is unlimited. The main objective of the proposed method is to reduce the number of calculations without compromising the results. It is achieved by changing the criteria used to select the candidate activity to be shifted. Hiyassat’s selection is based on both the value of the activity’s free float and its resource rate; while Harris’s selection is based on selecting the activity having the maximum improvement factor. An activity having an improvement factor of 0 or above indicates that the resource profile will improve by shifting the activity. The proposed method drastically reduces the number of calculations while preserving the accuracy of the results. Hiyassat [17] further notes that the main drawback of both methods (traditional and modified) is that neither of them provide the “true” minimum moment.

Hiyassat [18] presents a heuristic procedure that applies the Modified Minimum Moment Method to both methods of multiple resources leveling: Series and Combined. Hiyassat demonstrated his procedure by using the same examples that were used by Harris [16] to present the traditional minimum moment method. A comparison is made between the two, which shows that the modified method, in most cases, obtained better results than the traditional one. Furthermore, the research concludes that the Modified Minimum Moment Method is a valid heuristic method for multiple resources leveling.

Harris [19] presents another heuristic approach to level resources. This approach relies on the Critical Path Method (CPM) or Program Evaluation and Review Techniques (PERT) to determine the critical path, the duration of the project, and the critical activities. With resource leveling, it is always assumed that critical activities may not be leveled as they have zero float days and if shifted will lead to an increase to the project's duration. The resources of the critical activities are mapped onto a histogram, referred to as the base histogram. Next, the non-critical activities are queued in a priority order and added to the base histogram one-by-one. The activities are first ordered according to the decreasing resource rates, and then by the increasing total float and finally by the decreasing order of the sequence steps. Each activity is placed between its early start time and late finish time span. The activity is then assigned to the position with which the total sum of the resources is at minimum. The heuristic results in a histogram where the resources are leveled similarly as other heuristic and optimization techniques. As referred by Harris, the method is clear, logical and computationally efficient.

Zhang et al. [20] present a different heuristic approach for scheduling resource-constrained for repetitive projects that use multiple modes of resource demands. Unlike other research in which one activity is scheduled at a time, this approach considers scheduling alternative combinations of activities simultaneously. The combinations are determined through a permutation tree-based algorithm. The combination that results in the minimum project duration is selected. The researchers expect that the heuristic approach to be efficient and beneficial to others in their research and practice. Furthermore, using this heuristic, a feasible solution is always determined. This heuristic method does not take into consideration costs, and may be further extended to address resource leveling and other project uncertainties.

2.3 Optimization

One of the early models developed for scheduling multi-projects with limited resources is presented by Pritsker et al. [21]. The model is based on zero-one linear programming formulation. Some of the assumptions made in the model include limited resources, activity splitting possibilities, and precedence relations between activities. The desired objectives of the model include minimizing the total throughput time, the time by which all projects are completed, and total lateness or lateness penalty for all projects. The model results in optimal solutions for projects with fewer numbers of activities.

Easa [6] presents an integer-linear optimization model to level resources. The model levels one resource at a time and assumes that activities are uninterrupted. Unlike the previously discussed heuristic models, this model guarantees optimal leveling. The model relies on CPM to determine the critical path and the critical activities. The main objective of the model is to minimize the deviations between the actual and desirable resource rates. The model can be generalized to level multiple resources at the same time and can be extended to take into consideration the trade-off of cost scheduling. As with most optimization models, the main drawback of this model is that it can only be applied to small and medium sized projects because of the large number of calculations.

Ramlogan and Goulter [22] formulate a mixed integer model to level resources for project scheduling. As with other models, this model relies on CPM as an input to the model and uses the free float of the non-critical activities to level its resource. However, this model uses binary integer programming to ensure that activities are allocated as a whole, in which activities may not be interrupted. However, one of its objectives is to minimize the total durations of the individual activities. The other objectives include the overall resource leveling of the project and the resource leveling of the individual activities, which is known as internal leveling. The main concept of internal leveling is to try to avoid activity interruption. The objectives are integrated in the formulation by using a weighted multi-objective framework. The model can be improved by scheduling multiple resources concurrently, efficiently, and by adding priorities to each type of resource.

Bandelloni et al. [23] develop a non-serial dynamic programming approach for resource leveling. The approach results with an optimum solution to minimize the variation between the desired levels of resources and the resource requirements. The model relies on CPM to identify the critical and non-critical activities along with the project duration. The main objective of the procedure is to schedule the activities within their floats in order to achieve a rectangular-shaped histogram of the resource patterns. The model levels a single resource at a time, but may be further extended to allow for multiple resources leveling. The procedure assumes that activities are uninterrupted, with fixed durations, and constant resource rates. Like other resource leveling models, it is assumed that the project's completion date is fixed. The model looks for near-critical activities to reduce the number of interactions. Results have indicated that the procedure gives exact solutions for small and medium size projects.

Nudtasomboon and Randhawa [24] develop a zero-one integer programming model to schedule resource constrained projects. The model handles renewal and non-renewable resources, time-resource trade-offs and activity splitting. Furthermore, the model combines the three main objectives of project scheduling under one objective function. These objectives include minimum completion time, minimum project total cost, and minimum variation on resource levels. The main significance of the model is that the computational time for duration and cost problems are drastically reduced with regards the other previously presented algorithms, which don't take into consideration resource leveling.

Mattila and Abraham [25] develop an integer linear programming method to level resources for linear projects. Linear projects are characterized by having a set of activities that are repeated in different locations. Examples of linear projects include highways, pipelines, and tunnels. The method assumes that the project is planned and scheduled using a linear scheduling method. In addition, the controlling activity path is determined by using linear schedule models, and not the critical path method (CPM), because research has indicated that it is ineffective to use networks in scheduling linear projects. The formulation of the presented model relies on the ideas of rate float and activity float.

Senouci and Adeli [26] develop a mathematical model that uses the Neural Dynamic Model, developed by Adeli and Park, for resource scheduling. The model takes into consideration precedence relationships, multiple crew-strategies, and time cost trade-off, which are scheduling techniques that were ignored in prior research. In addition, the main objective of the model's formulation is to minimize the total project cost, rather than only minimizing the project duration. Moreover, the model performs resource-leveling and resource-allocation simultaneously, rather than independently, for resource scheduling. The model can be used for large projects and is very efficient for resource scheduling.

In most of the previous research, it is always assumed that an activity may not be interrupted, that is once an activity starts, it will continue until it's finished. However, in the following two papers, the researchers have taken an extra step for resource leveling in which activity splitting is allowed. In other words, an activity can be stopped and restarted during the project's duration.

Son and Matilla [27] formulate a binary linear programming model to level resources with activity splitting. The values of the decision variables, whether to split an activity or not, are only restricted to zero and one. The model relies on CPM to identify the critical and non-critical activities. Like other traditional models, this model also shifts the non-critical activities within their free-float to level resources. The main objective of the model is to measure the usage level of the resources. In addition, the model allows the practitioners to select certain activities to be split, to mimic the actual process. The model showed more realistic results when compared with results obtained from commercial software in which activity splitting was not allowed. However, their model did not consider the cost of splitting.

Hariga and El-Sayegh [4] propose an optimization model which uses a mixed binary-integer programming for resource leveling. The model takes into consideration activity splitting when leveling resource usage over the project life. This is achieved by moving the non-critical activities within their float. The objective of the paper is to minimize the costs associated with the splitting, the starting and stopping of activities, and the moving of the resources. The model assumes that the resources are unlimited; the resource rate for each activity remains constant over its duration. Although, an optimum solution is achieved through this model, large numbers of calculations are required.

2.4 Meta Heuristics

Leu and Yang [28] present a genetic algorithm based technique to schedule resource-constrained projects. The technique allocates multiple resources to a single project. Its main objective is to provide an optimum or near optimum project duration, while taking into consideration the resource constraints. One of the main drawbacks of using genetic algorithms for project scheduling is their sequencing; Leu and Yang overcome this problem by using crossover and mutation operators. Like heuristic models, genetic-based algorithms do not always provide optimum solutions.

Leu and Hung [29] establish a schedule simulation model based on genetic algorithms for scheduling resource-constrained projects. The model also takes into consideration variable activity durations because in reality activity durations are uncertain due to external environments such as weather, resource availability, and space congestion. The model uses probability distribution to come up with these uncertain activity durations. Along with providing the optimal project duration, the model is capable of providing the optimal averaged project duration and the cumulative project completion probabilities. The model may be improved to allow for resource leveling and time/cost trade-off for uncertain activity durations.

Dawood and Sriprasert [30] develop a genetic algorithm for the optimization of multi-constrained construction schedules. Examples of constraints include resource availability, execution space, physical dependency of construction products, and client instructions. The objective of the algorithm is to provide an optimum or near-optimum set of project duration, cost, and a smooth resource profile (resource-leveling). The results are achieved by altering the two sets of the chromosome string (one for the priority level assigned to each activity and the other for the options of construction methods assigned to the activity). The algorithm provides a schedule with acceptable searching time.

Montoya-Torres et al. [31] present yet another model based on genetic algorithms to schedule projects with limited resources. Unlike prior research, this research uses object-oriented programming to represent the chromosomes, which may allow the design of efficient decision support systems. Compared with other models, this model is effective and in many cases may find results in less computational time. The model does not take into consideration the activity costs. The authors suggest that further research is required to investigate the various formulation techniques for the multi-objective optimization problems.

Senouci and Eldin [32] present a model based on genetic algorithms for resource scheduling. This model performs resource leveling along with resource allocation simultaneously, unlike prior research in which they were performed independently. Moreover, this model takes into consideration the different precedence relationships (Start-Start, Start-Finish, Finish-Start, Finish-Finish), total project cost minimization, and time-cost trade-off. The model results in optimum or near optimum total costs. Also, to optimize the project schedule and the total cost, this model could be used along with CPM while performing resource leveling. The model can solve large project sizes with a large number of activities efficiently.

Christodoulou [33] presents a methodology using algorithms based on Ant Colony Optimization (ACO) artificial agents to schedule resource-constrained and resource-unconstrained projects. Also, the method can search for longest paths which are useful for solving direct network topologies. The method uses intelligent selection procedures to perform the path-route calculations. The algorithms have an advantage over the traditional CPM methods, due to its capability to calculate the shortest and longest paths. The presented methodology may further consider resource leveling and activity splitting.

Zhang et al. [11] introduce a Particle Swarm Optimization (PSO) approach to schedule resource-constrained projects. The objective of this approach is to minimize the project's duration. With PSO, a search is conducted to find the best location for each activity among a group of activities. The priorities of the activities are represented by particles and the group of activities is represented as a swarm. The solution achieved from these particles is transformed to a feasible solution using a parallel scheme. The approach takes into consideration the precedence and resource constraints. Through computational analysis, it has been shown that this approach is more efficient than genetic algorithm based approaches due to its search mechanism. The approach may be further improved to include resource leveling and multi-objectives.

Zhang et al. [34] introduce, yet another, methodology based on Particle Swarm Optimization (PSO) for scheduling resource constrained activities with activity splitting and break. The method assumes that activities are interrupted during non-working periods and are not necessarily resumed at the next working period. This because limited resources may be reallocated during their breaks and may not be back in time for the next working period. The main objective of this method is to minimize the project's duration while exploiting preemption and break of resource constrained projects. Using PSO, the solution is represented by the particle position which is then transformed into a feasible preemptive schedule using a parallel scheme. The authors verify the effectiveness of their method through computational analyses. Although this method takes into consideration activity splitting, it does not attempt to level resources, as to achieve a smooth resource profile.

Son and Skibniewski [35] develop a technique for resource leveling with the objective of minimizing the difference between the required resources and the desired resource profile. Their technique is based on local optimizers and a hybrid model. The local optimizer is developed by designing four independent algorithms, in which each algorithm uses combinations of different schemes for the order of shifting non-critical activities. A local optimal is obtained from each algorithm and the minimum optimal is chosen as the solution for the model. However, in order to enhance the performance of the local optimizers, the authors have decided to develop a hybrid model which combines the local optimizers with Simulated Annealing (SA). The model assumes that the project's duration is fixed and that once an activity starts, it should continue until it is completed. Hence, this model does not take into consideration activity splitting.

Zhang et al. [36] take a step further to present a model based on Particle Swam Optimization (PSO) for scheduling resource constrained projects using a permutation based scheme. The priority for scheduling an activity is determined by its order in the permutation. The activity must be placed after its predecessors in the permutation to take into consideration the precedent constraint. The main objective of this model is to minimize the project's duration. This permutation based model does not lead to combinatorial explosion or is problem-dependent effectiveness, and thus, it is considered more robust than heuristic methods. The model has demonstrated its effectiveness through computational experiments. This model may be further extended to include multiple objectives and to consider activity interruption.

Khanesar [37] introduces a modified discrete particle swarm optimization approach, with the aim of retaining the original definitions of the continuous PSO parameters. The velocity of the particle is interpreted as the rate at which the particle changes its bits' value. Also, the previous direction and previous state of each particle is taken into account upon updating a particle's position. The previous velocities of a particle contain information about the direction to previous local best and global bests of the particle which help in attaining better and faster solutions. This approach can be used in numerous applications which require a discrete search space.

Jun and Chang [38] present a binary mixture particle swarm algorithm to be used to solve discrete optimization problems. The algorithm combines the original binary particle swarm optimization with simulated annealing to avoid traps in the local optimal solution and improve the overall search capabilities of the algorithm. Moreover, the algorithm replaces PSOs update operation for particle velocity and position with the cross-operation of the genetic algorithm in order to simplify the algorithm's structure. The algorithm demonstrated that is it very efficient and has a fast convergence rate when tested against other PSO algorithms.

Liao et al. [39] present a discrete PSO algorithm which is applied to the flowshop scheduling problem. The proposed algorithm redefines the particle's position and velocity and utilizes an efficient approach which is based on frequency based memory to move the particle to a new position. The algorithm is tested against a continuous PSO algorithm and two genetic algorithms and has showed to be very competitive. Moreover, the authors extended their research to include PSO-LS, a local search scheme into the proposed algorithm, which performed well for the flowshop problem, but required more computational time.

Yang et al. [40] propose a discrete particle swarm optimization algorithm based on quantum theory, where the minimum unit which carries information is known as a qubit, and holds the value 0 or 1. This concept is applied to the PSO, where each bit of the particle position is either 0 or 1, and that the velocity of a particle represents the probability that the j^{th} bit of the i^{th} particle being zero. Note that the algorithm does not rely on the sigmoid function to achieve binary values, but rather generates a random number which dictates the value of the bit when compared to the velocity. The algorithm has been tested against other meta-heuristic approaches such as Genetic Algorithm and discrete PSO algorithms which use the sigmoid function. It has been proved that the proposed algorithm is simple, efficient, and converges at a fast rate.

As it can be noted from the above comprehensive literature review, resource leveling problem with activity splitting was not addressed using any heuristic approach. Most of the research tackled the resource-constrained problem. And a few took one step further to solve resource-constrained problems with activity splitting. Son and Skibniewski [35] develop a technique to minimize the difference between the required resources and the desired resources for resource leveling, as activity splitting was not taken into consideration. Moreover, none of the discrete PSO techniques developed by the other researchers tackle the resource-leveling problem.

2.5 Chapter Summary

This chapter presented a summary of the related literature that dealt with project resource leveling techniques. The techniques are grouped into three categories: heuristics, optimization and meta-heuristics. Based on this extensive review, it is clear that there is a need for a new resource leveling method that is computationally efficient and allows splitting of activities.

Chapter 3: Resource Leveling – Optimization Approach

3.1 Introduction

This chapter presents the optimization model developed by Hariga and El-Sayegh [4] to level resources with activity splitting. The model formulation is explained and a sample problem is demonstrated in the subsequent sections. Further, the model is used on several test problems, which serve as benchmarks, and are later used to assess the quality of the proposed heuristic procedure.

3.2 Optimization Model

Hariga and El-Sayegh [4] propose an optimization model for resource leveling while taking into consideration activity splitting by using a mixed binary-integer programming. In addition, the model aims at minimizing the costs associated with the shutdown and restart of the activities as well as the activities' dependent costs incurred because of the splitting. The latter type of costs is the result of work disruption such as labor inefficiency (loss of learning efficiency) due to the demobilization and subsequent remobilization [4]. As with other models, this proposed model relies on CPM to identify the critical and non-critical activities. The model assumes that the resources are unlimited, the resource rate for each activity remains constant over its duration, and that only non-critical activities can be split. Furthermore, this model allows for activity splitting and multi-resource leveling. Although, an optimum solution is achieved through this model, large numbers of calculations are required.

All optimization models consist of the following elements:

- State the assumptions
- Define the terms, parameters, and decision variables;
- Define the constraints to be met
- Set the objective function

The model considers that a project has n activities and P resource types. Each activity has a fixed duration represented by, T_j, j = 1, 2, ..., n. After performing CPM calculations, the early start (ES_j), late start (LS_j), early finish (EF_j), late finish (LF_j) as well as the total float (TF_j) for each activity is determined. The critical (nc) and non-critical (nn) activities are determined. The model reschedules the non-critical activities (*nn*) within the interval [ES_j, LF_j], thus not altering the project's duration.

The model requires that the below-mentioned set of assumptions to be satisfied at all times:

- There are *P* resource types available for running the project.
- The resource requirement rate for each activity remains constant over its duration.
- A non-critical activity can be split with an associated cost.
- The time to setup an activity prior to restarting activity is small enough such that it is carried out at the end of the split period.
- A non-critical activity resumes with the same resource requirement rate after preemption.
- The precedence relationships for activities that are split must be satisfied.

The model consists of two types of parameters: problem parameters and problem decision variables.

Problem Parameters

r_{ip} Number of units of resource type *p* ($p = 1, 2, \ldots, P$) needed to run activity $i = 1, 2, \ldots, n$.

Z_{ti} Binary parameter equal to one when critical activity *i* is active from period ES_i to period EF_i and zero otherwise, $i = 1, 2, .., nc$ and $t = 1, 2, \ldots, T$.

CI_p Cost of acquiring one unit of resource type *p* ($p = 1, \ldots, P$).

CD_p Cost of releasing one unit of resource type *p* ($p = 1, \ldots, P$).

CS_j Cost of splitting non-critical activity *j*, $j = 1, 2, .., nn$.

Problem Decision Variables

y_{tj} Binary variable equal to one when non-critical activity *j* is active (running) during period *t* and zero otherwise, $t = ES_j, ES_j+1, \ldots, LF_j$ and $j = 1, 2, .., nn$.

S_j Start time of non-critical activity j, $j = 1, 2, .., nn$.

F_j Finish time of non-critical activity j, $j = 1, 2, .., nn$.

L_{tj} Non-negative variable to determine whether activity j is split in period $t + 1$.

NL_j Number of times activity j is split, $j = 1, 2, .., nn$.

I_{tp} Number of units of resource type p $(p = 1, \ldots, P)$ acquired during period t, $t = 1, 2, \ldots, T$.

D_{tp} Number of units of resource type p $(p = 1, \ldots, P)$ released during period t, $t = 1, 2, \ldots, T$.

R_{tp} Requirement for resource type p $(p = 1, \ldots, P)$ on period t, $t = 1, 2, \ldots, T$.

The following is the cost optimization model for the resource leveling problem with allowed activity splitting:

$$Minimize \ \sum_{p=1}^{P}\left[CI_p\sum_{t=1}^{T}I_{tp} + CD_p\sum_{t=1}^{T}D_{tp}\right] + \sum_{j=1}^{nn}CS_jNL_j \qquad (1)$$

subject to:

$$R_{tp} - R_{(t-1)p} + D_{tp} - I_{tp} = 0, \qquad t = 1, 2, \ldots, T \ and \ p = 1, 2, \ldots, P \qquad (2)$$

$$R_{tp} = \sum_{i=1}^{nc} r_{ip}z_{ti} + \sum_{j=1}^{nn} r_{jp}y_{tj}, \quad t = 1, 2, \ldots, T \ and \ p = 1, 2, \ldots, P \qquad (3)$$

$$\sum_{t=ES_j}^{LF_j} y_{tj} = T_j, \quad j = 1, 2, \ldots, nn \qquad (4)$$

$$S_j = (T+1) - \max\{(T+1-t)y_{tj} : t = ES_j, ES_j + TF_j\}, \qquad j = 1, 2, \ldots, nn \qquad (5)$$

$$F_j = \max\{ty_{tj} : T = LF_j - TF_j, LF_j\}, \qquad j = 1, 2, \ldots, nn \qquad (6)$$

$$S_k \geq F_j + 1, \qquad j = 1, 2, \dots, nn \; and \; k \in Succ(j) \tag{7}$$

$$L_{tj} = Max\left(y_{tj} - y_{(t+1)j}, 0\right), \; j = 1, 2, \dots, nn \; and \; t = ES_j, LF_j \tag{8}$$

$$NL_j = \sum_{t=ES_j}^{LF_j} L_{tj} - 1, \quad j = 1, 2, \dots, nn \tag{9}$$

$$y_{tj} \in \{0, 1\}, \; j = 1, 2, \dots, nn \; and \; t = ES_j, LF_j$$

$$L_{tj} \geq 0, \; j = 1, 2, \dots, nn \; and \; t = ES_j, LF_j$$

$$I_t, D_t \geq 0, \qquad t = 1, 2, \dots, T - 1$$

$$y_{(T+1)j} = 0, \qquad j = 1, 2, \dots, n$$

In this model, there are three types of constraints: resource balance constraint, duration constraint, and network logic constraint.

Equation 3 expresses the resource requirement decision variable, R_{tp}, as a function of the binary variables, y_{tj} and z_t, which is used in the resource balance constraint which guarantees that the requirement for resource type *p* at time *t* plus the amount of the same resource acquired during period *t* is equal to the requirement for resource type *p* at time *t+1* plus the amount of the same resource released during time *t* (equation 2).

Equation 4 represents the duration constraint which is required to ensure that the total number of active periods for a non-critical activity *j* is equal to its duration T_j.

The network logic constraint is required to ensure that the network logic relationships between the non-critical activities are maintained. Since the critical activities have zero float and will not be moved, there is no need to include them in the network logic constraint.

To construct the network logic constraint, the starting time (S_j) and finishing time (F_j) for non-critical activity *j* can be determined as a function of binary variables using equations 5 and 6. Equation 7 represents the network logic constraint which ensures that the starting time of a successor activity is greater than or equal to the finishing time of its predecessor.

Finally, equation 1 represents the cost function of the optimization model, in which its objective is to minimize the sum of costs resulting from the fluctuations in resource usage over the project life as well as the total costs of splitting the non-critical activities. Note that equation 9 calculates the number of times that activity *j* is split.

As formulated, the model contains $\sum_{j=1}^{nn}(LF_j - ES_j + 1)$ binary variables and at most [2PT + $\sum_{j=1}^{nn}(LF_j - ES_j + 1)$ +2nn] continuous variables. Moreover, the model has at most [P(T-1)+nn+$\sum_{j=1}^{nn}(LF_j - ES_j + 1)$ +2$\sum_{j=1}^{nn}(TF_j + 1)$ +$\sum_{j=1}^{nn} nn_j$] constraints, where nn_j is the number of immediate successors of activity *j*. This proves that as the number of non-critical activities increase, the computational time of the model will increase as well.

3.3 Illustrative Example

In this section, an example of a project having 11 activities with 6 different resource types is illustrated using the optimization model developed by Hariga and El-Sayegh [4]. Figure 3 shows the project network for this illustrative example. After performing the CPM calculations, the *ES*, *EF*, *LS*, *LF* times and the slack for each activity are determined. Moreover, the critical and non-critical activities are identified as A, C, H, and K and B, D, E, F, G, I, and J, respectively. Note that the project's duration is 15 time periods which will be unchanged since only the non-critical activities can be delayed within their float in order to achieve smooth the resource profiles. The CPM results are taken as input in the model, along with the resource requirements, resource costs for acquiring and releasing, and splitting costs. Table 1 shows the data entered into the model with respect to the illustrative example.

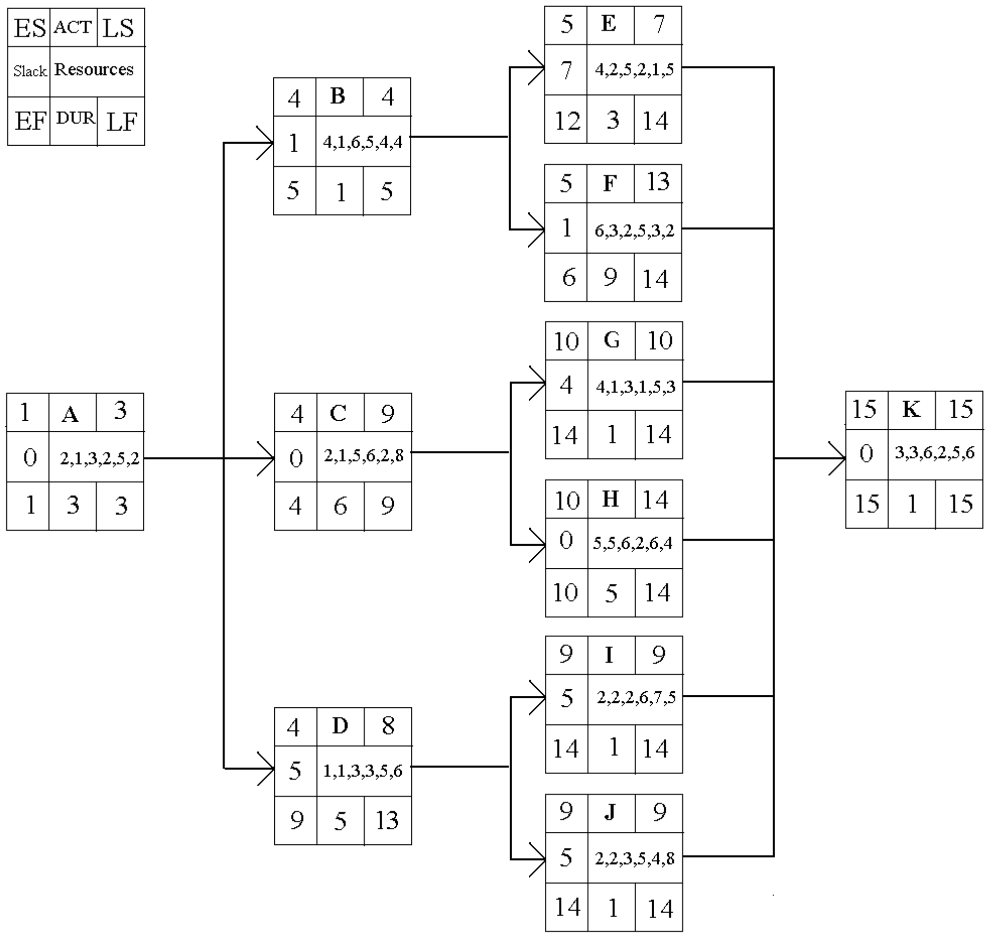


Figure 3: Project Network Example

Table 1: Data Input for the Illustrative Example

Activity	Duration	ES	LF	Slack	R1	R2	R3	R4	R5	R6	CS
A	3	1	3	0	2	1	3	2	5	2	0
B	1	4	5	1	4	1	6	5	4	4	2
C	6	4	9	0	2	1	5	6	2	8	0
D	5	4	13	5	1	1	3	3	5	6	2
E	3	5	14	7	4	2	5	2	1	5	3
F	9	5	14	1	6	3	2	5	3	2	5
G	1	10	14	4	4	1	3	1	5	3	1
H	5	10	14	0	5	5	6	2	6	4	0
I	1	9	14	5	2	2	2	6	7	5	7
J	1	9	14	5	2	2	3	5	4	8	4
K	1	15	15	0	3	3	6	2	5	6	0

Cost	
Acquired	**Required**
1	1
2	3
3	2
5	2
3	2
3	1

After inputting all the required data and running the model, the following Gantt chart is produced (Table 2). Table 2 displays the y_{tj} for each activity, which indicate the active and inactive periods and are used to in calculating the optimal cost. The chart shows that activity F is stopped (inactive) in time period 11, and that is why the y_{tj} for j = 11 is zero. Activity F resumes it work at period 12.

Table 2: y_{tj} of each activity in the Illustrative Example

Activities	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	1	1	1												
B				1	0										
C				1	1	1	1	1	1						
D				0	1	1	1	1	1	0	0	0	0		
E					0	0	0	0	0	0	1	1	1	0	
F					1	1	1	1	1	1	0	1	1	1	
G										0	0	0	0	1	
H										1	1	1	1	1	
I									0	0	1	0	0	0	
J									0	1	0	0	0	0	
K															1

Based on Table 2, the duration constraints are satisfied for all the non-critical activities, for example, for activity F,

$$\sum_{t=5}^{14} y_{tj} = (1+1+1+1+1+1+0+1+1+1) = 9$$

Once the y_{tj} values are determined, the problem decision variables can be calculated (Table 3). For example, R_{tp}, I_{tp}, and D_{tp} for resource type 2 at period 9, can be calculated as follows:

$R_{9,2} = (1 * 1) + (1 * 1) + (0 * 2) + (1 * 3) + (0 * 2) + (0 * 2) = 5$

Since $R_{10,2} - R_{9,2} = 10 - 5 > 0$, $I_{9,2} = 5$, and $D_{9,2} = 0$.

Table 3: R_{tp}, I_{tp}, D_{tp} for the Illustrative Example

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of Acquired	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
# of Released	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Total Resources 1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of Acquired	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
# of Released	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Total Resources 2	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of Acquired	1	3	7	0	1	1	1	7	0	0	1	9	0	1	
# of Released	0	0	0	7	0	0	0	0	3	1	0	0	7	0	
Total Resources 3	**1**	**2**	**5**	**12**	**5**	**6**	**7**	**8**	**15**	**12**	**11**	**12**	**21**	**14**	**15**

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of Acquired	1	3	7	0	1	1	1	7	0	0	1	9	0	1	
# of Released	0	0	0	7	0	0	0	0	3	1	0	0	7	0	
Total Resources 4	**1**	**2**	**5**	**12**	**5**	**6**	**7**	**8**	**15**	**12**	**11**	**12**	**21**	**14**	**15**

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of Acquired	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
# of Released	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Total Resources 5	**6**	**10**	**14**	**28**	**36**	**40**	**44**	**48**	**52**	**54**	**58**	**59**	**63**	**65**	**66**

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of Acquired	3	39	3	0	3	3	3	21	0	9	0	27	6	0	
# of Released	0	0	0	33	0	0	0	0	9	0	9	0	0	24	
Total Resources 6	**3**	**6**	**45**	**48**	**15**	**18**	**21**	**24**	**45**	**36**	**45**	**36**	**63**	**69**	**45**

The resource balance constraint is satisfied for all R_{tp}, for $t = 0, 1,...,T$ and $p = 1,2,\ldots,P$. For example, $R_{9,2}$ is 5 – 10 - 0 + 5 = 0.

Moreover, the network logic constraints for all the non-critical activities are satisfied. The starting times and finishing times can be easily computed using the equations mention in the previous section. For example, the starting time for activity F is

$S_j = 15 - \max\{11 \times 0, 10 \times 1\} = 15 - 10 = 5.$

$F_j = \max\{13 \times 1, 14 \times 1\} = 14.$

Figure 4 shows the resource utilization for the six different resource types used in this example before leveling while Figure 5 displays the resource utilization for the six different resource types after leveling. Each graph plots the resource requirements against the time period. Notice that the resource profile for each of the resources has improved after leveling. For resource type 2, the peak has dropped from 21 to 14 in addition to minimizing the fluctuations.

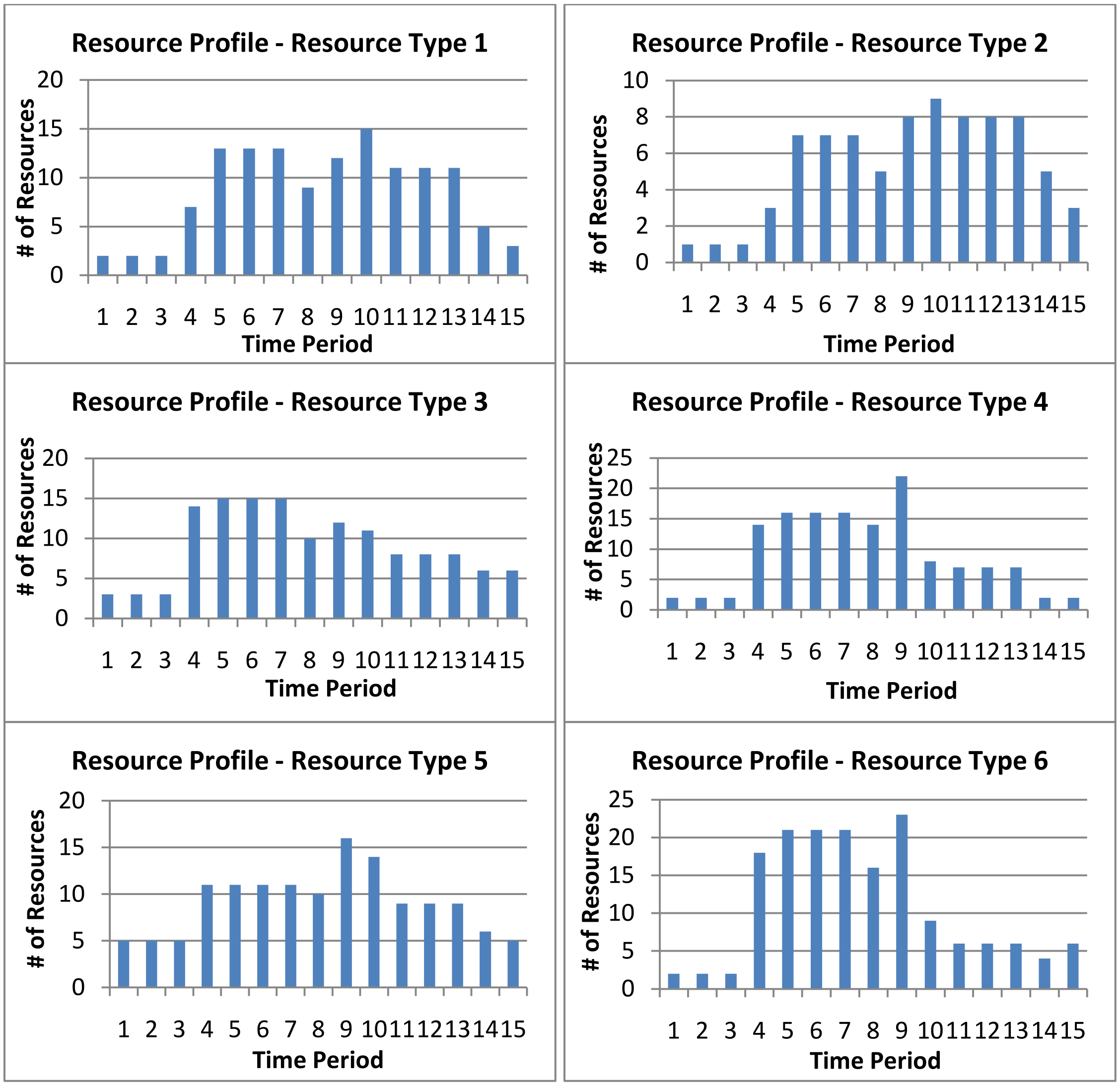


Figure 4: Resource Utilization for Illustrative Example (Before Resource Leveling)

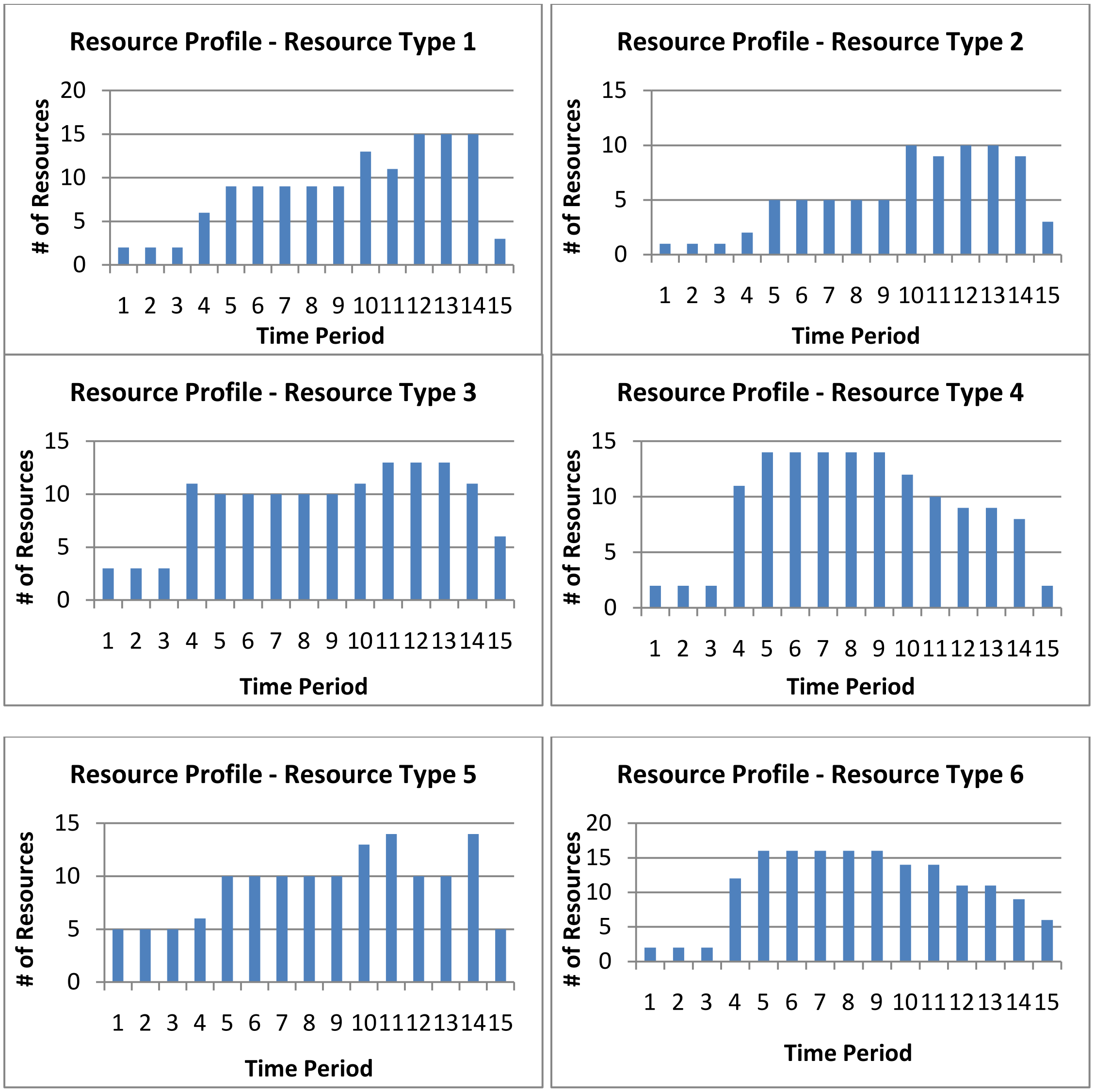


Figure 5: Resource Utilization for Illustrative Example (After Resource Leveling)

Finally, the optimal solution generated by this model has a total cost of 328 which is attained using the objective function of the model whose objective is to minimize total cost.

3.4 Benchmark Problems

As there are no benchmark problems available in the project scheduling literature, a set of test instances are developed to assess the solution quality of the proposed meta-heuristic approach. The test problems are generated by varying the number of non-critical activities, *nn*, and number of resource types, *P*. For each combination of $nn \in \{2, 4, 7, 8, 9, 10\}$ and $P \in \{2, 4, 6\}$, 10 instances with different problem parameters (i.e. varying the resource utilization rates and costs) are created. Each of the 180 problem instances are setup in Microsoft Excel with the application of the cost optimization model of Hariga and El-Sayegh[4] and are then solved using What'sBest 9 from LINDO Systems. These obtained optimal solutions are later used as benchmarks to evaluate the cost performance of the proposed heuristic procedure. Appendix A contains the network diagrams and Appendix B contains the optimal cost and the computational time for the 180 instances.

3.5 Chapter Summary

For large number of non-critical activities and long activity scheduling intervals (difference between the latest finishing time and earliest starting time), the model becomes a large mixed binary linear program requiring a large number of calculations to produce an optimum solution. Therefore, one of the objectives of this research work is to develop a heuristic procedure that is computationally efficient and generates high quality solutions. In the next two chapters, the Particle Swarm Optimization and Simulated Annealing models for leveling resources with activity splitting are presented, respectively.

Chapter 4: Particle Swarm Optimization Based Solution for the Multi-Resource Problem with Activity Splitting

4.1 Introduction

The chapter begins with an overview of the original Particle Swarm Optimization (PSO), discrete PSO and quantum discrete PSO. Next, the particle swarm optimization search procedure is implemented for the resource leveling problem with activity splitting. Six heuristic techniques based on different PSO search procedures are then presented. Subsequently, the six procedures are assessed based on the cost and time performance using the 180 test problems generated in Chapter 3. Finally, the chapter concludes with a summary of the findings.

4.2 Overview of Particle Swarm Optimization

The PSO technique consists of a population of particles whereby each particle is represented by a position in n-dimensional space and a velocity. The velocity corresponds to the speed and direction at which the particle is moving. The particles update their positions and velocities using their own previous best positions, cognitive learning, and the best previous position of all the particles, social learning, which are known as local best positions and global best position, respectively. The following subsections describe three previously developed PSO models.

4.2.1 Continuous PSO

The PSO method is first developed by Kennedy and Eberhart [10] as a continuous PSO in which the positions of the particles are represented as real numbers. The PSO model consists of a population of P particles in the swarm, where each particle is initialized with a random position and velocity. Next, the PSO procedure searches iteratively for the best position (near or optimum) by updating each particle's velocity and position using its own previous best position and best position of all particles. The local and global bests are determined through the assessment of each particle's fitness values. The search continues until convergence is attained which is either when the allowed maximum number of iterations, K, is exceeded or a relatively steady position is reached (i.e. the algorithm is trapped in one of the local optimum).

The particle's position and velocity are represented by $X_i(k)$ and $V_i(k)$, for $i = 1, 2, \ldots, P$ and $k = 1, 2, \ldots, K$. The N-dimensional position for the i^{ih} particle at the k^{th} iteration is denoted by

$$X_i(k) = [x_{i1}(k), x_{i2}(k), \ldots, x_{iN}(k)]$$

where,

$x_{ij}(k)$ represents the j^{th} coordinate of the i^{th} particle for $j = 1,2, \ldots, N$.

Similarly, the particle's velocity for the i^{th} particle at the k^{th} iteration is represented by

$$V_i(k) = [v_{i1}(k), v_{i2}(k), \ldots, v_{iN}(k)]$$

The updating mechanism of the i^{th} particle's velocity and position at the k^{th} iteration is performed using the following two equations, respectively [11]

$$V_i(k) = wV_i(k-1) + c_1 r_1 [X_i^L - X_i(k-1)] + c_2 r_2 [X^G - X_i(k-1)]$$

$$X_i(k) = V_i(k) + X_i(k-1)$$

where,

X_i^L is the local best position of the i^{th} particle found after the last k-1 iteration.

X^G is the global best position among all particles in the swarm visited so far.

w is the inertia weight used to reduce the impact of previous velocities on the current velocity so that it does not go out of control.

c_1 and c_2 are two positive parameters representing the cognition and social learning factors, respectively. If c_1 is large, then the particles tend to move towards their own local best, but if c_2 is large, then the particles tend to move towards the known global best so far.

r_1 and r_2 are random numbers between 0 and 1.

The velocity of any particle is restricted in the interval $[V_{min}, V_{max}]$. If the new velocity, V_{ij} is smaller than V_{min}, then V_{ij} is set to V_{min}. Similarly, if the new velocity, V_{ij} is larger than V_{max}, then V_{ij} is set to V_{max}.

4.2.2 Discrete PSO

Kennedy and Eberhart [12] developed a discrete binary PSO. The particles are represented as binary variables holding values of 0 or 1. Moreover, the velocities of the particle no longer represent the speed but rather represent either the probability of a position changing its value to one or the probability of a position being 0 [37, 41]. Thus, the values of the velocities are restricted to the interval [0, 1].

In discrete PSO, the particle's velocity is updated the same way as in the continuous PSO. However, a normalization function is used to transform the real numbers to binary numbers. This is done using the sigmoid function that is stated as follows:

$$v'_{ij}(k) = sig\left(v_{ij}(k)\right) = \frac{1}{1 + e^{-v_{ij}(k)}}$$

Once the velocity is normalized (i.e. its value is between 0 and 1), it is then used to update the position of the particle using the following equation:

$$x_{ij}(k+1) = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \begin{vmatrix} if\ r_{ij} < sig(v_{ij}(k+1)) \\ otherwise \end{vmatrix}$$

where, r_{ij} is a random number between [0,1].

4.2.3 Quantum Discrete PSO

Yang et al. [40] proposed the quantum discrete PSO, which is based on quantum theory. In the quantum theory, the quantum particle position, $X_i(k)$, consists of qubits, where each qubit (or bit) holds the values of 0 or 1. A quantum particle vector, $V_i(k)$, denotes the particle's velocity, v_{ij}, which represents the probability that the j^{th} bit of the i^{th} particle being 0.

The quantum discrete PSO uses the following equations to update the velocity of the particle and to obtain a new binary position:

$$V_i^L(k) = \alpha X_i^L(k) + \beta(1 - X_i^L(k))$$

$$V_i^G(k) = \alpha X^G(k) + \beta(1 - X^G(k))$$

$$V_i(k) = wV_i(k-1) + c_1 V_i^L(k) + c_2 V_i^G(\mathrm{k})$$

$$X_i(k) = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}\begin{vmatrix} if\ (rand < V_i(k)) \\ otherwise \end{vmatrix}$$

where,

α and β are control parameters which indicate the control degree of V, with $\alpha + \beta = 1$ and $0 \leq \alpha, \beta \leq 1$.

w, $c1$, and $c2$ represent the inertia weight, and the cognitive and social learning factors, respectively; where $w + c_1 + c_2 = 1$ and $0 \leq w, c_1, c_2 \leq 1$.

rand is a random number generated between [0, 1].

4.3 Implementation of PSO to the Resource Leveling Problem with Activity Splitting

In this research, a particle within the PSO population denotes a feasible schedule for a project having n activities and p resource types. Therefore, a particle represents the set of non-critical activities since the critical activities are not changed when leveling the resources.

Hereafter, each PSO particle consists of a number of bits, each representing the y_{tj} value of a given non-critical activity. Recall that the y_{tj} indicates if activity j is active at time period t. To illustrate this representation, consider a particle composed of the y_{tj} values of the example introduced in Chapter 3 (refer to the project's schedule in Table 2).

$$X_i(k) = [x_{i1}(k), x_{i2}(k), \ldots, x_{iN}(k)]$$

where,

$$x_{ij}(k) = \{y_{tj}\ values\ of\ the\ j^{th}\ non - critical\ activity\ of\ the\ i^{th} particle\ at\ the\ k^{th}\ iteration\}$$

For example, referring to Table 2, the i^{th} particle, which is composed of 7 non-critical activities, is represented as

$$X_i(k) = \{x_{i1}(k), x_{i2}(k), \ldots, x_{i7}(k)\}$$

$$X_i(k) = \begin{Bmatrix} (1,0), (0,1,1,1,1,1,0,0,0,0), (0,0,0,0,0,0,1,1,1,0), (1,1,1,1,1,1,0,1,1,1), (0,0,0,0,1), \\ (0,0,1,0,0,0), (0,1,0,0,0,0) \end{Bmatrix};$$

Note that the dimension, or the total number of bits, of $X_i(k)$ is calculated as

$$\sum_{j=1}^{N}(LF_j - ES_j + 1)$$

Figure 6, below, depicts the pseudo-code for the PSO algorithm which consists of mainly two phases: initialization and searching for best particle. In the initialization phase, P particles with feasible schedules are first generated. In this research the first particle is composed of the project schedule resulting from the CPM method; i.e. a feasible schedule without splitting. As for the remaining $P - 1$ particles, they are randomly generated using the algorithm described in section 4.3.1, which ensures that the particles are feasible. Next, K and $maxK_{imp}$ are initialized to the maximum number of iterations and the maximum number of iterations without any cost improvement, respectively. Finally, k, the iteration counter is set to 1 and k_{imp}, the iteration improvement counter, is initialized to 0.

In the second phase, the particles' positions are assessed. First, the fitness, $F(X_i(k))$, of each particle, i, at the k^{th} iteration is computed. Next, $F(X_i(k))$ is compared with the fitness of the best local position for particle i, $F(X_i^L)$. If $F(X_i(k))$ is a better fit than $F(X_i^L)$, then X_i^L is set to $X_i(k)$ and $F(X_i^L)$ is set to $F(X_i(k))$. Similarly, $F(X_i(k))$ is compared with the fitness of the global best position, $F(X^G)$. If $F(X_i(k))$ has a better cost performance than the global best position, then the global best position, $F(X^G)$, is set to $F(X_i(k))$ and k_{imp} is initialized to zero since a new global best is found; otherwise k_{imp} is incremented. Afterwards, the velocity, $V_i(k)$, and the position, $X_i(k)$, of the i^{th} particle are updated. Finally, iteration counter, k, is incremented. The search for the best particle continues until either the iteration counter, k, reaches the maximum number of iterations, K, or k_{imp} reaches the maximum number of steady state iterations.

Initialization:

Generate P feasible particles
Set K = maximum iteration counter
Set $maxK_{imp}$ = maximum number of iterations with no cost improvement
Set iteration counter, k, to 1
Set iteration improvement counter, $k_{imp} = 0$

Searching for best particle:

Do while ($k < K$ and $k_{imp} < maxK_{imp}$)

For each particle, i

Compute the fitness, F, for $X_i(k) \rightarrow F(X_i(k))$

If $F(X_i(k)) < F(X_i^L)$
$X_i^L = X_i(k); \quad F_i^L = F(X_i^L)$

If $F(X_i(k)) < F(X^G)$
$X^G = X_i(k); \quad F^G = F(X_i(k)); \quad k_{imp} = 0$
Else
$k_{imp} = k_{imp} + 1$

Update the velocity, $V_i(k)$

Update the position, $X_i(k)$

End while

$k = k + 1$

End while

Figure 6: Pseudo-code for PSO Algorithm

4.3.1 Generation of the Initial Particle Positions

The initial particles of a population are generated such that each particle represents a feasible project schedule. A project schedule is feasible if it meets the duration constraint, where the sum of the y_{tj} values of each activity is equal to its duration, and the network logic constraint, where each activity starts once all of its predecessors are finished. (Refer to section3.2 for more details) The first two steps of the algorithm ensure that the network logic constraints of the schedule are satisfied, and step 3 makes sure that the duration constraint is satisfied.

For each non-critical activity, *j*, within particle, *i*, the following steps are taken:

Step 1: If activity, *j*, does not have any non-critical activity as its predecessor:

- Generate a discrete random number between ES_j and $ES_j + TF_j - 1$.
- Set the starting time of activity *j*, S_j, to the generated random number.
- Set $y_{tj} = 0$, for all $ES_j, ES_j + 1, \ldots, S_{j-1}$.
- Go to step 3

Step 2: If activity, *j*, does have some non-critical activities as predecessors:

- Find the finishing times of the non-critical predecessors, which have been already determined, and set F as the largest finishing time.
- Set $S_j = F + 1$
- Set $y_{tj} = 0$, for all $t < S_j$.
- Go to step 3.

Step 3:

- Generate T_j discrete random numbers between S_j and LF_j.
- Set the cells (y_{tj}) corresponding to these random numbers to 1 and the remaining y_{tj} values to 0.

Figure 7: Pseudo-code for the Generation of Feasible Particle Algorithm

4.3.2 Generation of the Initial Particle Velocities

For each of the particle, a position and a velocity is determined. The initial velocities, $V_{ij}(0)$, are created, by randomly generating numbers between the lower and upped bounds defined for the velocity. In this research, the lower and upper bounds are set as $V_{min} = 0$ and $V_{max} = 1$, respectively. This ensures that the velocity, which denotes a probability, is restricted to values between 0 and 1.

4.3.3 Transformation of a Particle's Position into a Feasible Particle's Position

Once a particle's position is updated, it is possible that the new position does not represent a feasible solution. Therefore, the algorithm below is applied to transform the updated particle's position into a particle representing a feasible project schedule.

For each non-critical activity, j, in the i^{th} particle,

1. If activity, j, does not have any non-critical activities as predecessors, go to step 6; otherwise continue to step 2.
2. Find the starting time for activity j, S_j, using equation 5 in chapter 3.
3. Calculate the finishing times of activity j's predecessors and set F as the latest finishing time.
4. If $F > S_j$, then set $S_j = F + 1$.
5. For all $t < S_j$, set $y_{tj} = 0$.
6. Count the number of y_{tj} values that are equal to 1 between S_j and LF_j. If the sum of the ones is less than the activity's duration, T_j, continue on to step 7, otherwise go to step 12.
7. For $t \geq S_j$, count the number of y_{tj} values that are equal to zero, say z, and assign an equal probability for each, $\frac{1}{z}$.
8. Randomly choose a number between [0,1], r.
9. If $r \; \varepsilon \left[\frac{u-1}{z}, \frac{u}{z}\right]$, where $u = 1, 2, \ldots, z$, then set the u^{th} bit of the particle that has a value of 0 to 1.
10. Increment the count by 1.
11. Repeat steps 8-10 until the sum of the y_{tj} values that is equal to the activity's duration, T_j, and skip to step 17.
12. For $t \geq S_j$, count the number of y_{tj} values that are equal to one, say z, and assign an equal probability for each, $\frac{1}{z}$.
13. Randomly choose a number between [0,1], r.
14. If $r \; \varepsilon \left[\frac{u-1}{z}, \frac{u}{z}\right]$ where $u = 1, 2, \ldots, z$, then set the u^{th} bit of the particle that has a value of 1 to 0.
15. Decrement the count by 1.
16. Repeat steps 13-15 until the sum of the y_{tj} values is equal to the activity's duration, T_j.
17. Stop.

Figure 8: Pseudo-code for the Transformation of a Particle into a Feasible Particle Algorithm

Upon completing the afore-mentioned steps for all the non-critical activities, a feasible solution is attained. Note that if activity, j, has no predecessors, steps 2 – 5 are not required.

For example, assume that there are two activities A and B, where A is the predecessor of B. Activity A has duration of 2 and Activity B has duration of 3. Note that in the figure below, activity B does not meet the duration constraint. Therefore, the algorithm detailed above is utilized to attain a feasible solution as illustrated in Figures 9 and 10.

A	1	0	0	1				
B				1	1	1	1	1

Figure 9: Example – Infeasible Solution

A	1	0	0	1	**ES**			**LF**
B				~~1~~ 0	1	~~1~~ 0	1	1
	Must be set to 0 to satisfy the precedence constraint				0-25% u = 1	25-50% u = 2	50-75% u = 3	75-100% u = 4

Figure 10: Example - Feasible Solution

Equal percentage ranges of ¼ are set for each y_{tj} having a value of 1 as shown in the figure above. A random number, *r*, is generated, and the range it falls within is determined. The value of y_{tj} having that range is set to 0. This procedure is repeated until the sum of the y_{tj} of Activity B becomes 3.

In order to calculate the cost for a particle, the bits of the particle's position which consists of the y_{tj} values of the non-critical activities are input into the project's schedule, and all of the cost components are computed. (Refer to section 3.2 for more information regarding the cost function.)

4.4 PSO Heuristic Procedures

In this research, six different PSO procedures are proposed, whereby in each procedure a different mechanism to update a particle's velocity and/or position is employed. All of the six procedures utilize the PSO algorithm, summarized in section 4.3, to solve the 180 instance problems developed for this research.

Several experiments for each of the heuristic procedures are conducted; each with a different parameter setting (i.e. number of particles, maximum number of iterations, and maximum number of steady iterations). Based on the experiments, it is observed that the

PSO has more chances to hit the optimal solution for large numbers of particles; however, the optimal solution is generated at the expense of its time efficiency. Therefore, the size of the particle's population should balance the cost and time efficiencies of the PSO algorithm. In this study, the number of particles is problem dependent and is set equal to twice the number of binary variables; contrary to the PSO literature, where the number of particles is assumed to be fixed regardless of the size of the problem to be solved. Recall that for all problems the number of binary variables is equal to $\sum_{j=1}^{nn}(LF_j - ES_j + 1)$ which is small compared to the total number of possible solutions which is computed as

$$\#\ of\ possible\ solutions = \prod_{j=1}^{nn}\binom{tj}{EF_j - ES_j + 1}$$

Note that for each heuristic procedure, the maximum number of iterations is set to 400 with a steady position of no more than 100 iterations. (I.e. the global best position does not change in the last 100 iterations).

In the following subsections, a description of each heuristic procedure, which includes the update velocity and position mechanism, is presented. For illustrative purposes, each procedure is conducted for the project schedule example introduced in chapter 3. For each of the proposed heuristic procedures, the PSO algorithm is performed 10 times with the constant parameters representing the cognitive and social learning, c_1 and c_2, are both set to 0.25 and the inertia weight, w, is set to 0.50. For each run, the global best particle position and its cost are recorded. After completing the 10 runs, the particle with the least cost among the global best positions of the 10 runs becomes the best solution of the resource leveling problem. Moreover, the computational time to complete the 10 runs is recorded. The results of the heuristics are summarized at the end of the section.

4.4.1 PSO Procedure 1

This heuristic is based on the continuous PSO model presented by Kenedy and Eberhart [10], where the new velocity of the i^{th} particle, $V_i(k)$, which is restricted to [0, 1], is achieved using the equation:

$$V_i(k) = wV_i(k-1) + c_1r_1[X_i^L - X_i(k-1)] + c_2r_2[X^G - X_i(k-1)]$$

The particle's position is updated using

$$x_{ij}(k) = \begin{Bmatrix} 0 & \left| \begin{matrix} if\ v_{ij}(k) < 0.5 \\ otherwise \end{matrix} \right. \\ 1 & \end{Bmatrix}$$

Note that in this procedure the velocity is defined as the probability that the particle holds the value of 0 or 1. It is assumed that there is a 50-50 % chance for the particle to hold the values 0 or 1.

4.4.2 PSO Procedure 2

This heuristic is also based on the continuous PSO model presented by Kenedy and Eberhart [10]. However, the velocity is defined as the probability that the particle changes its value. The following mechanism is used to update the particle's position

$$x_{ij}(k) = \begin{cases} x_{ij}(k-1) & if\ v_{ij}(k) < 0.5 \\ 1 - x_{ij}(k-1) & otherwise \end{cases}$$

4.4.3 PSO Procedure 3

This procedure is based on the quantum discrete PSO algorithm, where the position of the particle contains only binary values. However, in this procedure the velocity is defined as the probability that the particle holds the value of 0 or 1. It is assumed that there is a 50-50 % chance for the particle to hold the values 0 or 1.

The new velocity of the i^{th} particle, $V_i(t)$, is achieved using the following set of equations:

$$V_i^L(k) = \alpha X_i^L(k) + \beta(1 - X_i^L(k))$$

$$V_i^G(k) = \alpha X^G(k) + \beta(1 - X^G(k))$$

$$V_i(k) = wV_i(k-1) + c_1 V_i^L(k) + c_2 V_i^G(k)$$

where,

α and β are random variables which indicate the control degree of V, with $\alpha + \beta = 1$ and $0 \leq \alpha,\beta \leq 1$.

w, c_1, and c_2 represent the inertia weight, and the **cognitive and** social learning factors, respectively; where $w + c_1 + c_2 = 1$ and $0 \leq w,c_1,c_2 \leq 1$.

Next, the new position of the particle is determined as follows:

$$x_{ij}(k) = \begin{Bmatrix} 0 & \text{if } v_{ij}(k) < 0.5 \\ 1 & \text{otherwise} \end{Bmatrix}$$

4.4.4 PSO Procedure 4

This procedure also relies on the quantum discrete PSO algorithm, and therefore, uses the same set of equations to update the velocity. However, the velocity is defined as the probability that the particle changes its value. The new position of the particle is determined as follows:

$$x_{ij}(k) = \begin{Bmatrix} x_{ij}(k-1) & \text{if } v_{ij}(k) < 0.5 \\ 1 - x_{ij}(k-1) & \text{otherwise} \end{Bmatrix}$$

4.4.5 PSO Procedure 5

In this procedure, a new PSO discrete algorithm is introduced, which extends the discrete PSO of Kennedy and Eberhart. This procedure presents a newly developed mechanism which calculates the new velocity of a particle and uses the same algorithm as the one presented in PSO Procedure 4 to update the particle's position.

The following equations are used to update the velocity of a particle:

$$V_i^L(k) = \alpha * X_i^L(k) + (1 - \alpha) * (1 - X_i^L(k))$$

$$V_i^G(k) = \beta * X^G(k) + (1 - \beta) * (1 - X^G(k))$$

$$V_{New} = w * V_i(k-1) + c_1 * V_i^L(k) + c_2 * V_i^G(k)$$

where,

α and β are random variables which indicate the control degree of $V_i^L(k) and\, V_i^L(k)$, with $0 \leq \alpha,\beta \leq 1$.

w, c_1, and c_2 represent the inertia weight, and the cognitive and social learning factors, respectively; where $w + c_1 + c_2 = 1$ and $0 \leq w, c_1, c_2 \leq 1$.

The particle's position is updated using

$$x_{ij}(k) = \begin{Bmatrix} 0 \left| \begin{matrix} if\ v_{ij}(k) < 0.5 \\ otherwise \end{matrix} \right. \\ 1 \end{Bmatrix}$$

4.4.6 PSO Procedure 6

In this procedure, the same PSO algorithm presented in PSO Procedure 5 is conducted but with the following slight change to the update mechanism of the particle's position.

$$x_{ij}(k) = \begin{Bmatrix} x_{ij}(k-1) \\ 1 - x_{ij}(k-1) \end{Bmatrix} \left| \begin{matrix} if\ v_{ij}(k) < 0.5 \\ otherwise \end{matrix} \right\}$$

Table 4 shows the summary of the results in terms of cost and computational time after performing each of the above-mentioned six heuristic procedures on the illustrative example of Chapter 3.

Table 4: Results of PSO Heuristic Procedures

PSO Procedure	Cost	Computational Time (seconds)	Percentage Difference
Optimization Model	328	84	0%
PSO Procedure 1	382	27	16%
PSO Procedure 2	440	99	34%
PSO Procedure 3	368	5	12%
PSO Procedure 4	379	181	16%
PSO Procedure 5	390	54	19%
PSO Procedure 6	401	37	22%

From Table 4, it can be noted that none of the PSO procedures were able to attain the optimal solution generated by the optimization model presented in Chapter 3 for this particular example. However, PSO Procedure 3 generated the nearest solution with a cost of 368 in 5 seconds, a 12% cost difference between the generated solution and the optimal.

4.5 Performance Analysis of the Six Heuristic Procedures

Each of the proposed six heuristic procedures is assessed using the 180 test problems generated in Chapter 3. The heuristic procedures are programmed in Java and are run on an HP Pavilion Notebook PC 2.13 GHz with 3.0 GB RAM. The same set of initial PSO particles, generated by setting the seed of the random number to "123456789", is used for all of the procedures.

For each procedure, c_1 and c_2 are initially varied between [0.05, 0.95] with an increment of 0.05 (ie. c_1 = 0.05 and c_2 = 0.05, c_1 = 0.05 and c_2 = 0.10 … c_1 = 0.95 and c_2 = 0.05). Upon running the six procedures with the different values of c_1 and c_2, it is noted that when the value of c_1 is equal to the value of c_2 and are in the range of [0.20, 0.50] better results are obtained. In other words, when the particles move with equal probability to the best global solution, each heuristic procedure generates better results. Therefore, it is decided to analyze the results of the procedures when the values of c_1 and c_2 are equal to each other and vary in the interval [0.25, 0.45], with an increment of 0.05.

The performance of each heuristic procedure is based on the percentage deviation of its cost from the optimal solution and the CPU time. Appendix C contains the complete results (cost and CPU time) of the heuristic procedures assessed using all 180 test problems.

The cost quality of each heuristic is also assessed using the number of problems resulting in 0% cost deviation and the number of problems having a cost deviation of less than or equal to 2%, 5%, and 10%. Moreover, for each variation in c_1 and c_2, the minimum, maximum, and average cost percentage difference for all the procedures are calculated.

Tables 5 to 10 report the results of the six PSO procedures when conducted for c_1, c_2 equal to {0.25, 0.30, 0.35, 0.40, and 0.45}.The highlighted row in each table shows the best cost performance of the corresponding heuristic.

Table 5: PSO Procedure 1 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	15.47	47.72	59	61	74	78
0.30	0	16.48	47.72	53	57	68	72
0.35	0	15.61	58.33	52	57	66	70
0.40	0	13.47	37.70	52	56	66	73
0.45	0	13.52	53.33	51	54	67	79

Table 6: PSO Procedure 2 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	38.20	162.50	27	27	29	34
0.30	0	49.65	136.36	1	1	4	6
0.35	0	49.65	136.36	1	1	4	6
0.40	0	49.65	136.36	1	1	4	6
0.45	0	49.65	136.36	1	1	4	6

Table 7: PSO Procedure 3 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	7.74	34.51	57	66	91	114
0.30	0	8.89	38.64	51	61	78	105
0.35	0	7.50	35.29	63	69	90	116
0.40	0	7.76	31.11	66	72	89	110
0.45	0	7.85	33.33	63	69	89	112

Table 8: PSO Procedure 4 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	16.84	58.33	40	42	57	63
0.30	0	16.31	58.33	41	44	56	67
0.35	0	18.57	58.33	40	42	52	62
0.40	0	15.99	47.06	43	47	58	65
0.45	0	16.53	58.33	39	43	59	69

Table 9: PSO Procedure 5 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	19.48	60.00	39	40	45	54
0.30	0	20.96	100.00	36	36	42	53
0.35	0	22.53	67.24	31	31	33	43
0.40	0	23.97	109.09	30	31	34	41
0.45	0	24.21	81.81	31	32	36	41

Table 10: PSO Procedure 6 - Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	16.86	58.33	41	47	59	63
0.30	0	16.68	50.00	40	45	57	68
0.35	0	17.37	58.83	41	44	54	64
0.40	0	16.96	50.00	40	42	56	61
0.45	0	16.26	58.33	37	39	55	66

By comparing the results of the heuristic procedures conducted for all the test problems, it can be concluded that PSO Procedure 3 has generated the best results, in terms of percentage difference between the optimal and the yielded results. The generated project schedules have resulted in near optimum solutions, with an average percentage difference of 7.5%. Moreover, 116 problems have a percentage difference of less than or equal to 10%. PSO Procedure 2 has generated the worst results where in some instances only one problem out of the 180 had a percentage difference of 0.

For large size problems, the computational time is significantly less when performing resource leveling using the proposed heuristic procedures. Moreover, it can be noted that most runs of the PSO algorithms ended upon reaching 100 iterations, which means that the particles were trapped in local optimal and were unable to reach the global optimal solutions. Therefore, the need to combine Particle Swarm Optimization with Simulated Annealing is required to avoid having particles trapped in local optimum, and thus generate improved results (i.e. near optimum project schedules).

4.6 Chapter Summary

An overview of the original PSO, discrete PSO and quantum discrete PSO are presented. Furthermore, the particle swarm optimization model for resource leveling with activity splitting is discussed along with the algorithms to initialize particle positions and velocities as well as transforming particles into feasible project schedules. The constraints related to resource leveling have been discussed whereby a particle must satisfy the duration constraints and the network logic constraints. Moreover, six different heuristic procedures were presented and analyzed; each with a newly developed PSO approach. PSO Procedure 3 has shown good results with an average cost deviation of 7.5%. In the next chapter, each of the proposed PSO procedures is combined with Simulated Annealing to avoid particles from becoming trapped in local optimum and hence generate better results.

Chapter 5: Hybrid Particle Swarm Optimization and Simulated Annealing Solution to the Multi-Resource Leveling Problem with Activity Splitting

5.1 Introduction

The Particle Swarm Optimization heuristic has many advantages; some of these include its simplicity in coding, ease of implementation with fewer parameters to adjust and its consistency in performance, along with its local and global search abilities. However, one drawback of PSO is the possibility of being trapped in local optima. Therefore, PSO is combined with Simulated Annealing (SA) to overcome this deficiency.

This chapter begins with an overview of Simulated Annealing. Next, the Simulated Annealing search procedure is presented. Then, each of the six heuristic procedures presented in Chapter 4 is run for the illustrative example using PSO combined with SA. After that, the six heuristic procedures are assessed based on their cost and time performance. Finally, the chapter concludes with a summary of the findings.

5.2 Overview of Simulated Annealing

Simulated Annealing (SA) is a probabilistic based search meta-heuristic to locate a good solution to a global optimization problem with multiple local optimal. Simulated annealing was introduced by various researchers in the mid 1980s [42]. The concept of simulated annealing is based on the analogy between the simulation of the annealing of solids and the problem of solving large combinatorial optimization problems [8]. Annealing refers to the process in which the particles of a solid are randomly arranged once the solid turns to liquid at high temperatures. Technically speaking, as the temperature rises, the particles of a solid tend to move around each other faster to make new forms. One of the main advantages of simulated annealing is its ability to find good solutions without being trapped in a local optimum.

The Simulated Annealing search procedure consists of two phases: initialization and searching for the best neighboring solution. The initialization phase begins by setting an initial solution as the best solution found so far, *sbest*. The initial solution is usually the best solution generated by another search procedure. Also, the initial temperature,

$temp_{0,}$ is set to a fixed or calculated temperature. *temp* is first set to the initial temperature which decreases at a rate of λ as the procedure iterates to search for a good solution in the next phase.

In the second phase, the procedure iterates until it finds a candidate solution. During each iteration, a neighboring solution, *s'*, is generated from the neighborhood of *s*. Next, the fitness of the neighboring solution is computed, *F(s')* and is compared to the fitness of the current solution *s*, *F(s)*, where their difference is stored as Δ. The neighboring solution is considered as a candidate for *sbest* if it is either a better solution than the one found so far or the probability of accepting a worse solution is high. This probability relies on the variable, *temp*, which represents the temperature in the annealing process. The higher the value of *temp*, the more chances a solution is accepted (i.e. more randomness). The following is a pseudo-code for the classical simulated annealing search procedure.

```
Initialization:
        Get initial solution, s
        Set sbest = s
        Set Temp_o = initial temperature
        Set Temp_f = final temperature
       Set Temp = Temp_0
Searching for best neighboring solution:
        Do while Temp < Temp_f
           Do while r < R (Perform the following steps R times)
                Generate randomly a neighboring solution s' from neighborhood of s
                Compute fitness of s'
                Compute Δ = fitness (s') – fitness (s)
                If (Δ ≤ 0,)
                   s' is accepted and sbest = s'
                Else
                   If (rand < exp(-Δ/Temp))
                        s' is accepted and sbest = s'
                r ← r + 1
```

End while

Update the temperature: *Temp* ← *λ Temp*

End while

where,

$Temp_0$ is the initial temperature of the annealing process

$Temp_f$ is the final temperature to be reached

λ the rate at which the temperature will decrease

R the total number of iterations at each temperature *temp*

s the best solution found so far

s' the neighboring solution

Figure 11: Pseudo-code for SA

5.3 Simulated Annealing for the Multi-Resource Leveling Problem with Activity Splitting

In this research, the Simulated Annealing algorithm is slightly modified in order to increase the chances of achieving a near optimum solution in less time. At each change in temperature, 10 neighboring solutions are generated as opposed to the general algorithm in which only one neighbor is generated. Next, the fitness is calculated for each of the neighboring solutions, and the neighboring solution with the best fit (*s'*) is compared to the best solution found so far (*s*). Note that each neighboring solution represents a feasible project schedule which is generated using the neighborhood selection algorithm and the control parameter settings described in the subsequent subsections.

5.3.1 Neighborhood Selection

In this research, each neighboring solution represents a feasible project schedule, which is composed of the y_{tj} values of the non-critical activities. A neighboring solution is generated by swapping a random pair of y_{tj}*'s* having different values, for each of the non-critical activities within the particle. The pairs are determined using a discrete probability algorithm.

Prior to determining the pair to be swapped, the y_{tj} values of the non-critical activity must first satisfy the duration and the network logic constraints. In other words, the sum of the y_{tj} values of the activity should be equal to its duration and that the y_{tj}'s of the activity are only active between its calculated start and finish times. Refer to section 4.3 for more details on how to transform an activity to a feasible activity.

For each feasible non-critical activity, two discrete random numbers between its starting and finishing times are generated, say *u* and *v*. If the values of y_{uj} and y_{vj} are different, then they are swapped. However, in case the two y_{tj} values are equal, two new random numbers are generated until they correspond to different y_{tj} values. This process is repeated for all of the non-critical activities within a particle, and thus, a new neighboring solution is generated.

5.3.2 Control Parameter Settings

The SA algorithm has several parameters. The main three parameters are $Temp_0$, $Temp_f$, and λ. $Temp_0$ denotes the initial temperature and $Temp_f$ is the final temperature. At each iteration of the algorithm, the temperature is decreased at a constant rate, λ, which is between 0 and 1. The smaller the value of λ, the slower the algorithm reaches the final temperature, and thus, increases the chances of finding a better solution. However, a slow search increases the computational time. Therefore, it is very important to choose wisely the settings of the parameters.

In this research, it is decided to vary the initial temperature depending on the problem's characteristic. In general, a neighboring solution at the *n*th iteration is accepted if it is a better fit (less cost) than the best generated solution so far or if it is near to the best solution by a certain probability, $\mathrm{e}^{-\Delta/\mathrm{temp_n}}$.

Let Δ denote the change in the fitness between the best solution, *F(s)*, and the neighboring solution, *F(s')*; $temp_n$ represent the current temperature, which is equivalent to $\lambda^n temp_0$; and P_{max} denote the maximum probability to accept a neighboring solution. Therefore, the initial temperature is determined using the acceptance probability as follows:

$$P_{max} = e^{-\Delta/temp_n}$$

$$P_{max} = e^{-\frac{\Delta}{\lambda^n temp_0}}$$

$$temp_0 = -\frac{\Delta}{\lambda^n \ln(P_{max})} > -\frac{\Delta}{\lambda \ln(\mathrm{P_{max}})}$$

In this research, the initial temperature, $temp_{0,}$ is calculated given that a neighboring solution is accepted with a maximum probability, P_{max}, of 80% and if there is an increase in the cost performance of no more than 20%. Thus,

$$\Delta = 0.20 * F(s)$$

Furthermore, since there is a tradeoff between the computational time and a good solution, the initial temperature is set as

$$temp_0 = -\frac{0.2\, F(s)}{0.9 \ln(0.8)}$$

to reduce the computation time. Also, λ is set as 0.90, so that the temperature descends at a slower rate where there are more chances of finding a good solution.

5.4 Implementation of PSO and SA Search Procedure

The PSO/SA search procedure is composed of three stages. In the first stage, only the PSO search procedure is performed, with the number of particles equal to twice the number of y_{tj} values. During the run the local positions of each of the particles is updated along with the global best position of all the particles. The PSO search procedure stops by either reaching the maximum number of iterations or becoming stuck in local optimum.

In the next stage, the local best particle positions obtained from the previous stage are input in the SA search procedure. For each local best particle, 10 feasible neighboring solutions are generated. The neighboring solution with the least cost is selected to be compared to the best local solution of that particle. The selected neighboring solution is accepted as a good solution if it reduces the cost of the particle or if its cost is not greater than 20% of the best SA position. Once Simulated Annealing procedure ends by reaching the final temperature at 0.01, the global best PSO position is updated only if the SA has produced a better cost performance.

Finally, in the last stage, the PSO search procedure is performed one more time with SA output as initial particles. This procedure (PSO-SA-PSO) is repeated 10 times and the best global position is returned. (Refer to Figure 12)

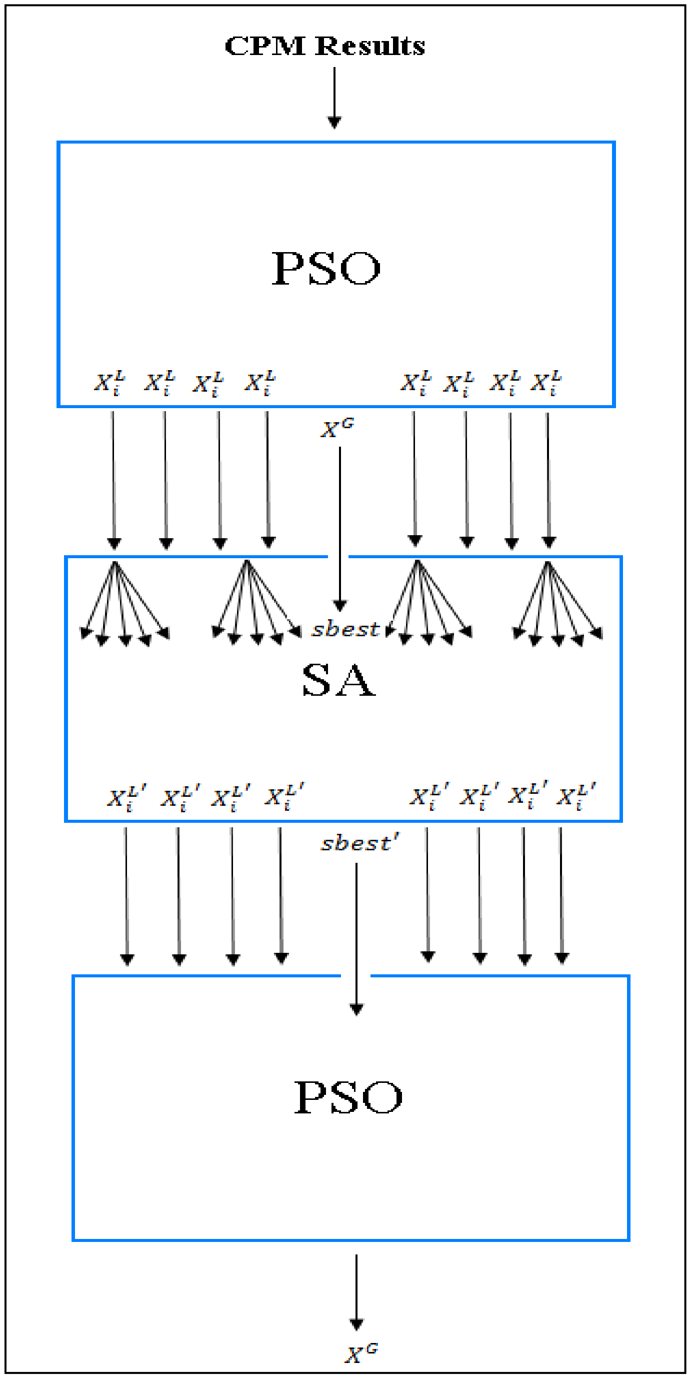


Figure 12: PSO-SA-PSO Search Procedure

5.5 PSO/SA Illustrative Examples

In this section, the six heuristic procedures, presented in Chapter 4, are extended to demonstrate the improvement resulting by incorporating the SA algorithm. Table 11 shows the results of the PSO heuristic procedures with SA that are attained for the illustrative example used throughout this research where $c_1 = c_2 = 0.25$.

Table 11: Results of PSO Heuristic Procedures with SA

PSO/SA Procedure	PSO Results			PSO/SA Results		
	Cost	Computational Time (seconds)	Percentage Difference	Cost	Computational Time (seconds)	Percentage Difference
PSO/SA Procedure 1	382	27	16%	335	222	2%
PSO/SA Procedure 2	440	99	34%	348	239	6%
PSO/SA Procedure 3	368	5	12%	328	264	0%
PSO/SA Procedure 4	379	181	16%	345	432	5%
PSO/SA Procedure 5	390	54	19%	343	174	5%
PSO/SA Procedure 6	401	37	22%	343	269	5%

By combining the PSO with SA, a better project schedule is generated, in terms of cost, for each of the six heuristic procedures. PSO/SA Procedure 3 actually generated the optimal solution having a cost of 328 within 264 seconds.

5.6 Performance Analysis for the Combined PSO/SA Search Procedure

Each of the six heuristic procedures is assessed using the 180 generated test problems. The PSO/SA procedure consists of three stages. In the first stage, only the PSO search procedure is performed. In the next stage, the local best particle positions obtained from the previous stage are input in the SA search procedure. For each local best particle, 10 feasible neighboring solutions are generated. The neighboring solution with the least cost is selected to be compared to the best local solution of that particle. The selected neighboring solution is accepted as a good solution if it reduces the cost of the particle or if its cost is not greater than 20% of the best SA position. Note that the search is reduced to neighboring solutions that will only result in an increase in cost of no more than 20%.

This will help to determine a good starting value of the temperature as discussed in section 5.3.2. Once the final temperature, which is set to 0.01, is reached, the global best PSO position is updated only if the Simulated Annealing has produced a better cost performance. Finally, in the last stage, the PSO search procedure is performed one more time with SA output as initial particles. This procedure (PSO-SA-PSO) is repeated 10 times and the best global position is returned.

5.6.1 Cost Performance of PSO/SA

For each procedure, the cost of the generated solution is recorded, in which the minimum, maximum, and average percentage difference on cost for all the procedures is calculated. The analysis performed is based upon the percentage difference of cost between the optimal solution and the generated solutions. Tables 12 to 17 report the results of the six PSO procedures combined with SA when conducted for c_1, c_2 equal to {0.25, 0.30, 0.35, 0.40, and 0.45}. The highlighted row in each table shows the best cost performance of the corresponding heuristic. Note that when c_1, c_2 = 0.25, PSO/SA procedure 3 resulted with an average cost difference of 4.23% and the costs of 147 out of the 180 problems were within 10% of the optimal ones.

Table 12: PSO/SA Procedure 1 – Results

	% Difference			Frequency			
c1, c2	**Minimum**	**Average**	**Maximum**	**= 0%**	**≤ 2%**	**≤ 5%**	**≤ 10%**
0.25	0	6.85	23.53	67	77	89	115
0.30	0	6.94	23.53	58	69	85	116
0.35	0	7.39	23.53	56	66	77	112
0.40	0	7.02	22.41	57	70	79	117
0.45	0	7.31	22.22	57	65	77	114

Table 13: PSO/SA Procedure 2 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	9.67	47.06	46	55	71	95
0.30	0	10.11	58.33	46	53	74	99
0.35	0	10.11	58.33	46	53	74	99
0.40	0	10.11	58.33	46	53	74	99
0.45	0	10.11	58.33	46	53	74	99

Table 14: PSO/SA Procedure 3 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	4.23	18.18	87	95	113	147
0.30	0	4.79	18.18	74	88	107	143
0.35	0	4.47	20.00	84	95	112	147
0.40	0	4.29	22.41	82	95	111	149
0.45	0	4.85	19.82	75	88	108	144

Table 15: PSO/SA Procedure 4 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	7.94	45.83	53	61	75	102
0.30	0	7.72	24.24	56	60	72	107
0.35	0	8.67	39.39	56	65	79	104
0.40	0	7.73	22.15	53	58	70	104
0.45	0	7.78	33.33	55	64	73	109

Table 16: PSO/SA Procedure 5 – Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	8.37	34.09	48	57	70	102
0.30	0	8.23	58.33	50	59	73	105
0.35	0	8.65	33.33	45	54	75	103
0.40	0	8.61	47.73	43	52	70	106
0.45	0	9.19	58.33	44	57	72	103

Table 17: PSO/SA Procedure 6 - Results

	% Difference			Frequency			
c1, c2	Minimum	Average	Maximum	= 0%	≤ 2%	≤ 5%	≤ 10%
0.25	0	8.21	35.29	56	62	71	103
0.30	0	8.54	41.67	55	62	73	103
0.35	0	8.53	35.29	56	59	71	100
0.40	0	8.16	33.33	53	62	71	105
0.45	0	8.28	32.32	53	62	75	103

The combination of Particle Swarm Optimization with Simulated Annealing has allowed the particles to search for solutions in different spaces rather than becoming trapped in local optimum. By analyzing the results of the heuristic procedures assessed using the 180 test problems, it can be concluded that PSO/SA Procedure 3 has generated the best results, in terms of percentage difference between the optimal and generated results. The generated project schedules have resulted in near optimum solutions, with an average percentage difference of only 4.23%. Furthermore, 81.67% of the test problems have a percentage difference of less or to 10%.

5.6.2 Computation Time Performance of PSO/SA

For each heuristic, the time required to generate a solution, also known as the computational time, is recorded. The table below displays the computation times recorded for test problems 151 – 180, with 178 binary variables, using the exact optimization procedure and the PSO/SA Procedure 3, which generated the best results. (Refer to Appendix D for full results).

Table 18: Exact Procedure vs Best Heuristic Computation Times

Problem #	Exact Time (seconds)	Best Heuristic Time (seconds)
151	1189	487
152	1479	560
153	861	501
154	1299	505
155	1355	498
156	1937	541
157	2549	520
158	1789	563
159	728	517
160	621	528
161	5359	569
162	8861	560
163	6899	589
164	2544	588
165	6551	537
166	6121	617
167	2384	596
168	1680	626
169	4673	593
170	9064	617
171	5955	560
172	10785	638
173	4236	592
174	5935	603
175	4480	580
176	6492	609
177	2609	653
178	7153	630
179	5445	603
180	4743	626

It should be mentioned here that an odd time performance is noticed for the *What's Best* application utilized to solve the 180 procedures with the exact procedure. In some problem instances, the computation time of a given large size problem is smaller than another problem with fewer number of binary variables. For example, it takes

What's Best 15211 seconds to solve a problem with 130 binary variables. However, an optimal solution for a problem with 178 binary variables and the same number of resources is obtained after 2384 seconds. Moreover, the average computation times for problems with 130 and 178 binary variables are 6563 and 4192 seconds, respectively. This abnormal observation is explained by the fact that the time performance of What's Best depends on the initial solution entered in the Excel sheet. Obviously, if such initial solution is close to the optimal one, it will take shorter time to terminate. Consequently, it is decided to assess the time performance of PSO/SA only for large size problems with 178 binary variables. Note that from the above table that the computational time recorded using PSO/SA Procedure 3 is far less than the computational time of the optimization model. The average reduction in computation time for the large size problems is 7 times, where in some problems a computation time reduction of 15 times is attained.

To further illustrate the significance of the proposed heuristic procedure and to show the extent of time savings for larger problems, two relatively large problems with 25 activities, of which 15 are non-critical, are created. These two problems, having 320 binary variables, are solved using the exact and the proposed heuristic procedures. When solved for the exact procedure, the *What's Best* solver was interrupted after having computation times of more than 24 hours, and hence, no optimal solution is generated. However, when the two problems are run using the best PSO/SA heuristic approach (PSO Procedure 3), solutions are generated in 1380 and 1512 seconds, which is equivalent to 23 and 25 minutes, respectively. This proves that heuristic procedures are more computationally time efficient for large sized project schedules.

In conclusion, even with the implementation of Simulated Annealing along with the PSO model, the computational time is indeed significantly less the optimization model presented in Chapter 3 for large size problems. Moreover, the computational time of PSO/SA can be further reduced by carefully designing a time efficient mechanism for the update of the particles' position which does not affect the feasibility of the particles. In fact, through the experimentations of the PSO/SA search procedures, it can be noted that much of the computation time of the search procedures is taken in making the particle feasible. The feasibility algorithm, introduced in section 4.3, is run for each particle after updating its position and at each iteration. This observation and its suggestion could be the subject of future research.

5.7 Chapter Summary

In this chapter each of the Particle Swarm Optimization heuristic procedures is combined with Simulated Annealing to overcome PSO's drawback of having particles being trapped in local optimum. First, an overview of Simulated Annealing is presented, which is followed by a description of the Simulated Annealing search procedure. Afterwards, a summary of the results of the heuristic PSO procedures with SA for the illustrative example are presented. The heuristic procedures were assessed using all the 180 problems, where it was evident that PSO/SA Procedure 3 has generated the best results with the lowest average in percentage change and the most number of problems with a percentage difference of less than or equal to 10%.

Chapter 6: Conclusion and Recommendations

Resource leveling is an important technique that is applied by project managers in order to improve the resource profile by minimizing the fluctuations in resource requirements. Most resource leveling techniques assumes that activities are continuous. A recent research proposed a new method to level resources while allowing activity splitting using optimization techniques. Optimization techniques allow reaching the optimum solution but it is time-consuming especially for large projects. Based on the review of the literature, it is clear that there is a need for a search procedure for the multi-resource leveling problem with activity splitting that is computationally efficient. This thesis presents Particle Swarm Optimization search procedures complemented with Simulated Annealing for the resource leveling of project schedules with activity splitting.

The first step in this research was to develop a set of 180 test problems to serve as benchmark problems in order to assess the performance of the proposed meta-heuristics. Each of the 180 problems was solved using the optimization model, where the cost and computation time were recorded. It was noted that as the size of the problem increased, the computation time increased dramatically. For example, a problem with 178 binary variables took an average of 4193 seconds to generate the optimal solution.

One of the main advantages of meta-heuristics is their ability to find near – optimum solutions in a short time period. The Particle Swarm Optimization procedure consists of a population of particles; each having a position and a velocity. The particles' positions and velocities are updated using their own previous best positions and the best position of all the other particles. In resource leveling terms, the particle's positions are represented by the y_{tj} values of the non-critical activities of a feasible project schedule.

In this research, six PSO heuristic approaches were developed; each having different approaches to update the particle's velocity and position. Each of the search procedures with different parameter settings were assessed using the 180 benchmark problems. The results were analyzed by calculating the minimum, average, and maximum percentage difference in costs for all the procedures. In addition, the cost quality of each heuristic was also assessed using the number of problems resulting in 0% cost deviation and the number of problems having a cost deviation of less than or equal to 2%, 5%, and 10%.

By analyzing the results of the heuristic procedures conducted for all the test problems with variations in the parameters, it can be concluded that PSO Procedure, which is based on the quantum discrete PSO, has generated the best results, with an average percentage cost difference of 7.5%. Moreover, 116 out of the 180 problems have a percentage cost difference of less than or equal to 10%. As for the computation time, all the heuristic procedures were able to generate solutions in less time than the optimization procedure, especially for large problems. For example, a problem with 130 binary variables generated an optimal solution in 4382 seconds while the heuristic procedure generated a solution with a 2% cost difference in 130 seconds.

However, it was noticed that for large size problems the heuristics were trapped in local optimum and the search discontinued. Therefore, it was decided to take the PSO search procedure a step further to combine it with Simulated Annealing. The main purpose of Simulated Annealing is to move particles to different search spaces without being trapped in local optimum. In SA, a neighboring solution is determined by swapping one pair of distinct y_{tj} values for each non-critical activity within a particle.

The six heuristic procedures, along with their different parameter variations, were assessed using the 180 benchmark problems. For each heuristic procedure, the minimum, maximum, and average cost is calculated along with the percentage cost difference. PSO/SA Procedure 3, having $c_1 = c_2 = 0.25$, has attained the best results with an average cost difference of 4.23% and the costs of 147 out of the 180 problems were within 10% of the optimal ones. Furthermore, 81.67% of the test problems have a percentage difference of less than or equal to 10%. As for the computation time, the heuristic procedures generated solutions in less time as compared to the optimization model. The average reduction in computation time for the large size problems is 7 times, where in some problems a computation time reduction of 15 times is attained.

Furthermore, to illustrate the significance of the proposed heuristic procedure, two project schedules having 25 activities, of which 15 are non-critical, were created. These two problems, having 320 binary variables, were solved using the exact and the proposed heuristic procedures. The exact procedure was unable to generate solutions for both of these schedules within a 24 hour time period. However, the proposed heuristic procedure generated results within 25 minutes; a large saving in the time. Therefore, this proves that heuristic procedures are more computationally time efficient.

This research is an important additional step in the ongoing research on resource leveling. The proposed heuristic procedure offers several improvements over the current resource leveling techniques. The proposed procedure allows for activity splitting, which is more realistic and results in better resource profile. The new procedure takes advantage of combining Particle Swarm Optimization with Simulated Annealing to reach the optimum or near optimum solution in a short period. The proposed procedure allows planners to consider the tradeoff between the cost of activity splitting and the cost of resource fluctuations resulting in a minimum overall project cost.

It is recommended that a software program is developed that enables the use of this procedure by leading scheduling software such as Primavera and Microsoft project. This will make it easier for practitioners to use this technique. Another recommendation for future research is to use this technique to solve the combined problem of resource leveling and time-cost tradeoff with and without allowed activity splitting. The improvement of the computation time of the hybrid PSO/SA is another line of future research. Indeed, based on our numerical experimentation with the proposed heuristics, the mechanism for particle position's update can be redesigned so that it does not affect the feasibility of the algorithm. Finally, one more topic for future research is to compare the proposed PSO/SA search procedure with other meta-heuristics such as genetic or tabu search procedures.

References

[1] PMBOK, *A Guide to the Project Management Body of Knowledge*, 4th Edition, Project Management Institute, 2008.

[2] H. Kerzner, Project *Management: A System Approach to Planning, Scheduling, and Controlling,* 9th Edition, John Wiley & Sons Inc., 2009.

[3] W. S. Herroelen, "Resource-Constrained Project Scheduling – The State of the Art," *Operational Research Quarterly*, vol. 23, no. 3, pp. 261-275, September 1972.

[4] M. Hariga and S. M. El-Sayegh, "Cost Optimization for the Multi-Resource Leveling Problem with Allowed Activity Splitting," *Journal of Construction Engineering and Management*, ASCE, vol. 137, no. 1, pp. 56-64, January 2011.

[5] F. S. Hillier and G. J. Leiberman, *Introduction to Operations Research*, 9th Edition, New York: McGraw – Hill International Edition, 2010.

[6] S. M. Easa, "Resource Leveling in Construction by Optimization," *Journal of Construction Engineering and Management*, ASCE, vol. 115, no. 2, June 1989.

[7] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research*, vol. 5, pp. 533-549, 1986.

[8] P. Laarhoven and E. Aarts. *Simulated Annealing: Theory and Applications. Dordretcht*, The Netherlands: Kluwer Academic Publisher, 1987.

[9] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons Ltd, Chichester, England, 1995.

[10] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE International Conference in Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[11] H. Zhang, H. Li, and C. M. Tam, "Particle Swarm Optimization for Resource-Constrained Project Scheduling," *International Journal of Project Management*, vol. 24, pp. 83-92, 2006.

[12] J. Kennedy and R. Eberhart, “A Discrete Binary Version of the Particle Swarm Algorithm”, *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp.4104-4108, 1997.

[13] C. F. Gray and E. W. Larson, *Project Management: The Managerial Process*, 4th Edition, New York: McGraw-Hill Irwin, 2008.

[14] J. W. Hinze, *Construction Planning and Scheduling*, 3rd Edition. New Jersey: Pearson Prentice Hall, 2008.

[15] T. Hegazy, *Computer-based Construction Project Management*, New Jersey: Prentice Hall, 2002.

[16] R. B. Harris, *Precedence and Arrow networking Techniques for Construction*, New York: Wiley, 1978.

[17] M. A. Hiyassat, “Modification of Minimum Moment Approach in Resource Leveling,” *Journal of Construction Engineering and Management*, vol. 126, no. 4, July/August 2000.

[18] M. A. Hiyassat, “Applying Modified Minimum Moment Method to Multiple Resource Leveling,” *Journal of Construction Engineering and Management*, vol. 127, no. 3, May/June 2001.

[19] R. B. Harris, “Packing Method for Resource Leveling (PACK),” *Journal of Construction Engineering and Management*, vol. 116, no. 2, June 1990.

[20] H. Zhang, H. Li, and C. M. Tam, “Heuristic Scheduling of Resource-Constrained, Multiple Mode and Repetitive Projects,” *Construction Management and Economics,* vol. 24, pp. 159-169, February 2006.

[21] A. B. Pritsker, L. J. Watters, and P. M. Wolfe, “Multi-Project Scheduling with Limited Resources: A Zero-One Programming Approach,” *Management Science*, vol. 16, no. 1, September 1969.

[22] R. N. Ramlogan and I. C. Goulter, "Mixed Integer Model for Resource Allocation in Project Management," *Engineering Optimization*, vol. 15, pp. 97-111, 1989.

[23] M. Bandelloni, M. Tucci, and R. Rinaldi, "Optimal Resource Leveling using Non-Serial Dynamic Programming," *European Journal of Operational Research*, vol. 78, pp. 162-177, 1994.

[24] N. Nudtasomboon and S. U. Randhawa, "Resource-Constrained Project Scheduling with Renewable and Non-Renewable Resources and Time-Resource Tradeoffs," *Computers and Industrial Engineering*, vol. 32, no. 1, pp. 227-242, 1997.

[25] K. G. Mattila and D. M. Abraham, "Resource Leveling of Linear Schedules Using Integer Linear Programming," *Journal of Construction Engineering and Management*, ASCE, vol. 124, no. 3, May/June 1998.

[26] A. B. Senouci and H. Adeli, "Resource Scheduling Using Neural Dynamics Model of Adeli and Park", *Journal of Construction Engineering and Management*, ASCE, vol. 127, no. 1, January/February 2001.

[27] J. Son and K. G. Mattila, "Binary Resource Leveling Model: Activity Splitting Allowed," *Journal of Construction Engineering and Management*, ASCE, vol. 130, no. 6, December 2004.

[28] S. S. Leu and C. H. Yang, "A Genetic-Algorithm-Based Resource-Constrained Construction Scheduling System," *Construction Management and Economics*, vol. 17, pp. 767-776, 1999.

[29] S. S. Leu and T. H. Hung, "A Genetic Algorithm-Based Optimal Resource-Constrained Scheduling Simulation Model," *Construction Management and Economics*, vol. 20, pp. 131-141, 2002.

[30] N. Dawood and E. Sriprasert, "Construction Scheduling Using Multi-Constraint and Genetic Algorithms Approach," *Construction Management and Economics*, vol. 24, pp. 19-30, January 2006.

[31] J. R. Montoya-Torres, E. Gutierrez-Franco, and C. Pirachicán-Mayorga, "Project Scheduling with Limited Resources Using a Genetic Algorithm," *International Journal of Project Management*, vol. 28, no. 6, pp. 619-628, August 2010.

[32] A. B. Senouci and N. N. Eldin, "Use of Genetic Algorithms in Resource Scheduling of Construction Projects," *Journal of Construction Engineering and Management*, ASCE, vol. 130, no. 6, December 2004.

[33] S. Christodoulou, "Scheduling Resource-Constrained Projects with Ant Colony Optimization Artificial Agents," *Journal of Computing in Civil Engineering*, vol. 24, no. 1, January 2010.

[34] H. Zhang, H. Li, and C. M. Tam, "Particle Swarm Optimization for Preemptive Scheduling under Break and Resource-Constraints," *Journal of Construction Engineering and Management*, ASCE, vol. 132, no. 3, March 2006.

[35] J. Son and M. J. Skibniewski, "Multiheuristic Approach for Resource Leveling Problem in Construction Engineering: Hybrid Approach," *Journal of Construction Engineering and Management*, ASCE, vol. 125, no. 1, January/February 1999.

[36] H. Zhang, H. Li, and C. M. Tam, "Permutation-Based Particle Swarm Optimization for Resource-Constrained Project Scheduling," *Journal of Computing in Civil Engineering*, vol. 20, no. 2, March 2006.

[37] M. A. Khanesar, "A Novel Binary Particle Swarm Optimization," *Proceedings of the 15th Mediterranean Conference on Control & Automation*, Athens, Greece, T33-001, July 2007.

[38] X. Jun and H. Chang, "The Discrete Binary Version of the Improved Particle Swarm Optimization Algorithm," *International Conference on Management and Service Science*, pp. 1-6, September 2009.

[39] C. J. Liao, C. T. Tseng, and P. Luarn, "A Discrete Version of Particle Swarm Optimization for Flowshop Scheduling Problems," *Computers & Operations Research*, vol. 34, no. 10, pp. 3099-3111, 2007.

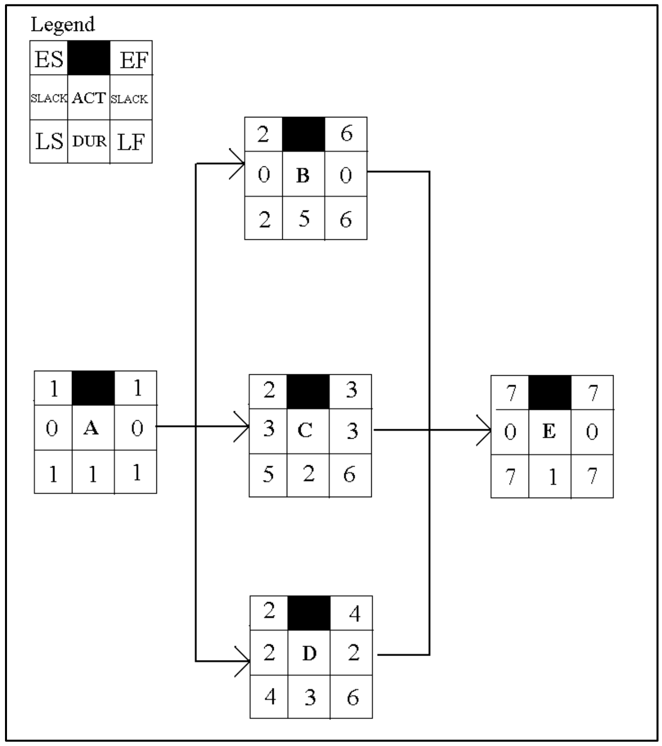
[40] S. Yang, M. Wang, and L. Jiao, “A Quantum Particle Swarm Optimization,” *Congress on Evolutionary Computations*, pp 320-324, 2004.

[41] A. Chen, G. Yang, and Z. Wu, “Hybrid Discrete Particle Swarm Optimization algorithm for Capacitated Vehicle Routing Problem,” *Journal of Zhejiang University Science A*, vol. 7, no. 4, pp. 607-614, 2006.

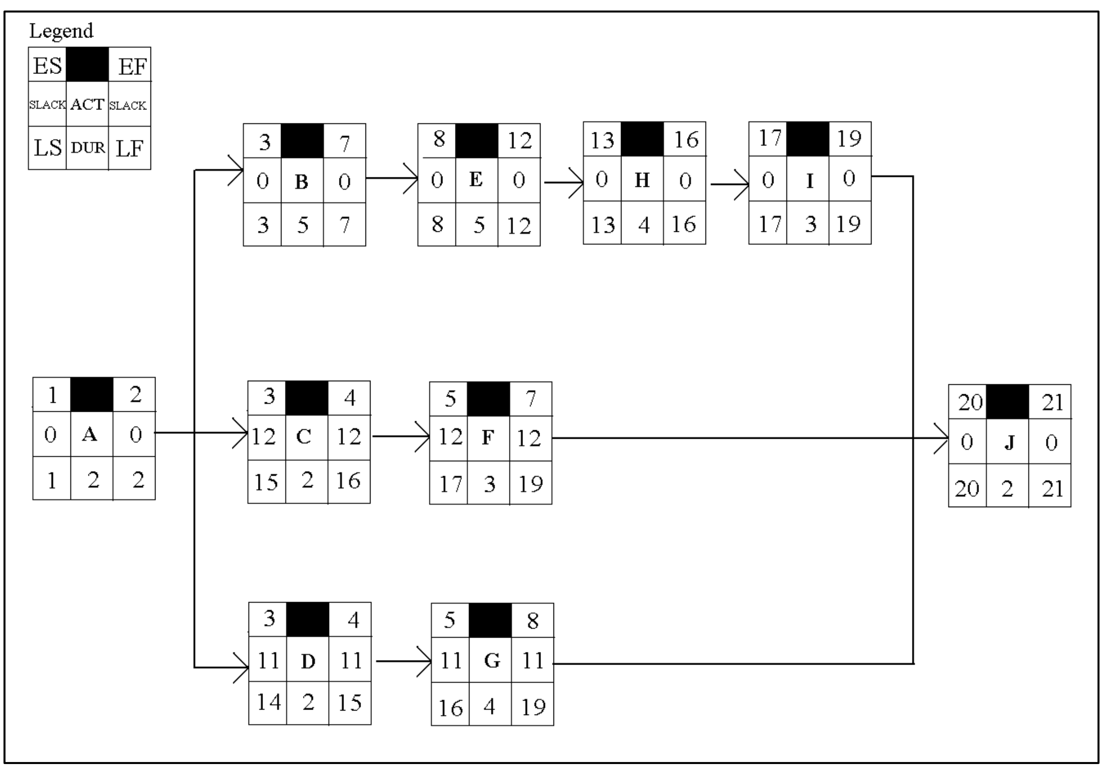
[42] S. Luke, *Essentials of Metaheuristics*, 2009. [E-book]
Available: http://cs.gmu.edu/!sean/book/metaheuristics.

Appendix A Network Diagrams for the 180 Test Problems

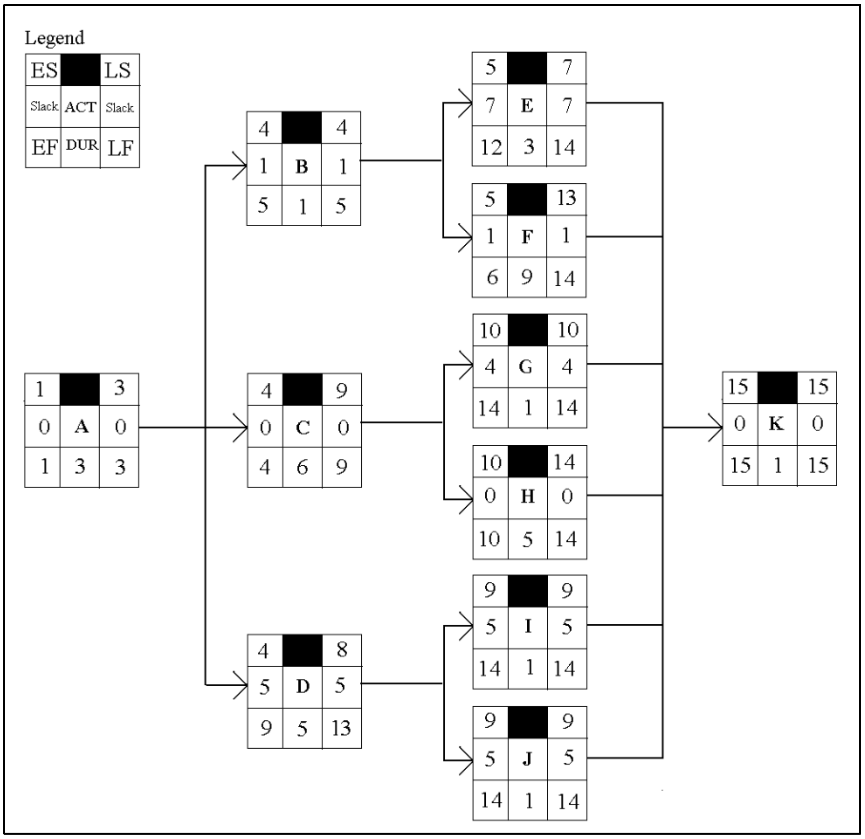
Number of Non-Critical Activities: 2



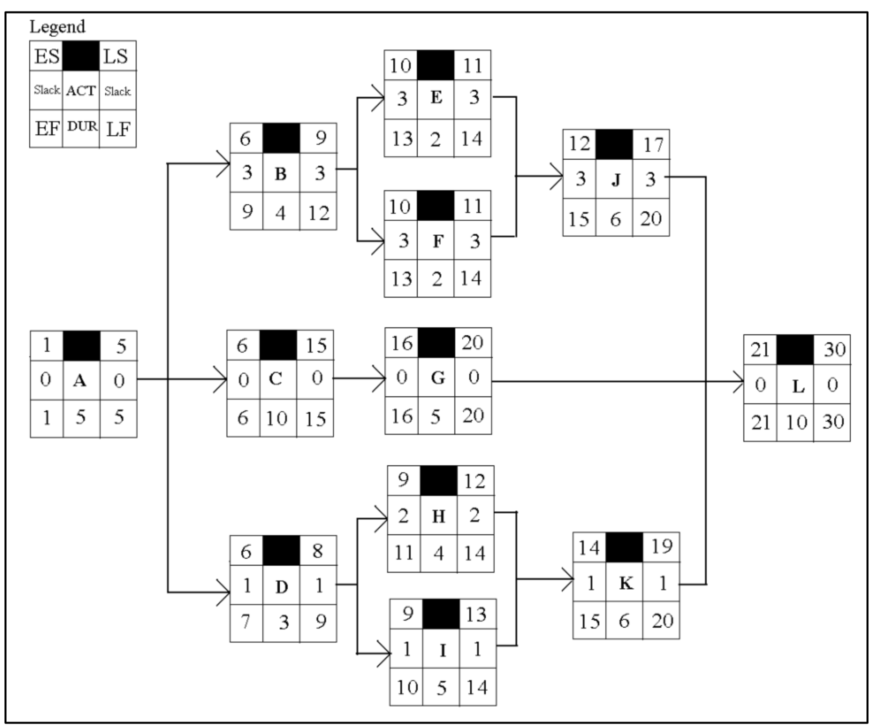
Number of Non-Critical Activities: 4



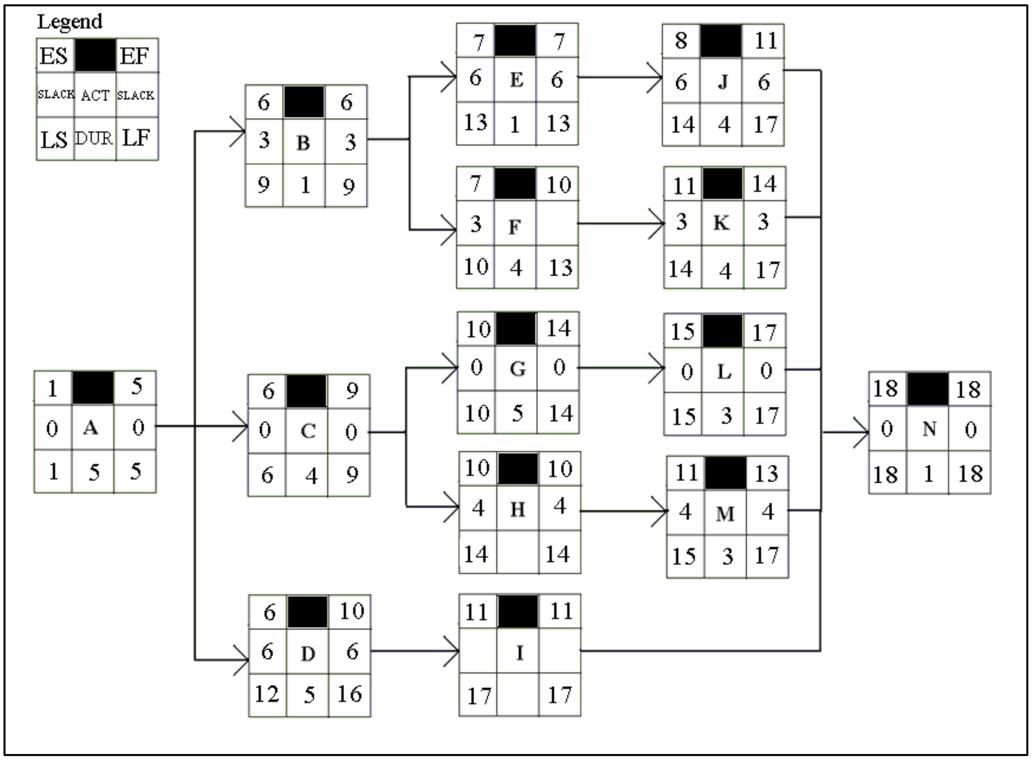
Number of Non-Critical Activities: 7



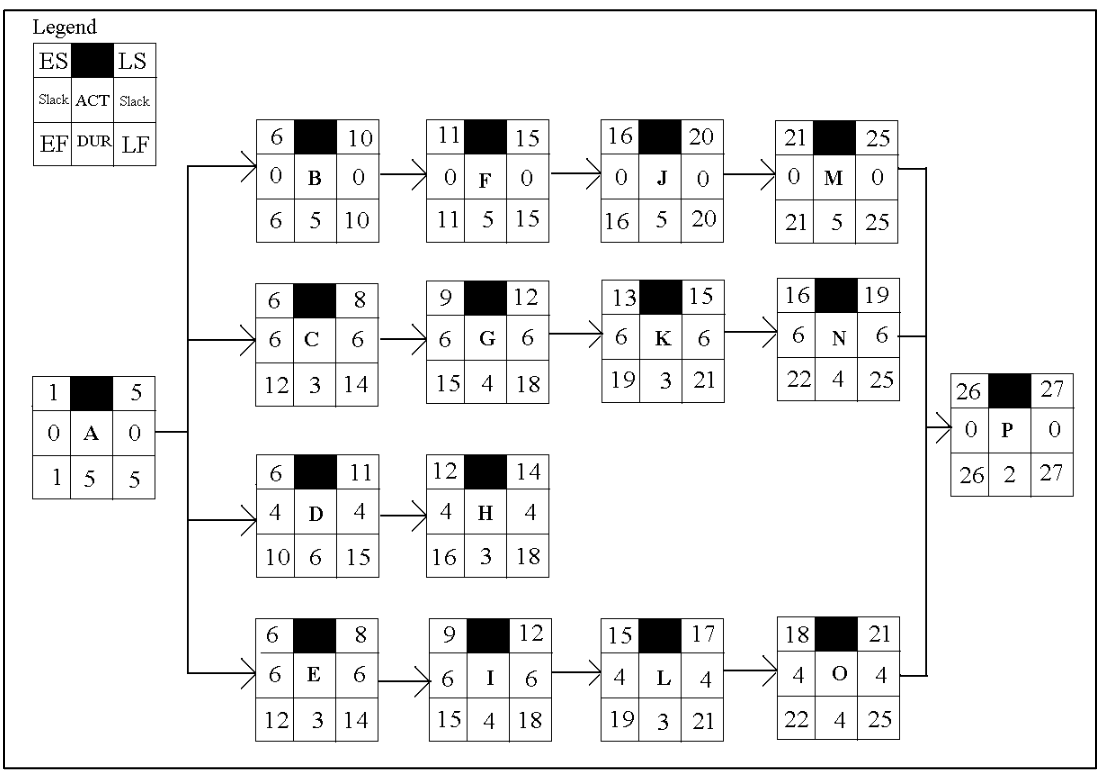
Number of Non-Critical Activities: 8



Number of Non-Critical Activities: 9



Number of Non-Critical Activities: 10



Appendix B Optimization Results for 180 Test Problems

Problem #	# of Non-Critical Activities	# of Resources	Instance #	# of Binary Variables	Optimal Cost	Time in Seconds
1	2	2	1	20	7.00	0
2	2	2	2	20	14.00	0
3	2	2	3	20	9.00	0
4	2	2	4	20	7.00	0
5	2	2	5	20	16.00	0
6	2	2	6	20	14.00	0
7	2	2	7	20	9.00	0
8	2	2	8	20	16.00	0
9	2	2	9	20	11.00	0
10	2	2	10	20	17.00	0
11	2	4	1	20	18.00	0
12	2	4	2	20	61.00	0
13	2	4	3	20	49.00	0
14	2	4	4	20	48.00	0
15	2	4	5	20	61.00	0
16	2	4	6	20	49.00	0
17	2	4	7	20	92.00	0
18	2	4	8	20	92.00	0
19	2	4	9	20	44.00	0
20	2	4	10	20	54.00	0
21	2	6	1	20	35.00	0
22	2	6	2	20	95.00	0
23	2	6	3	20	82.00	0
24	2	6	4	20	35.00	0
25	2	6	5	20	95.00	0
26	2	6	6	20	82.00	0
27	2	6	7	20	142.00	0
28	2	6	8	20	142.00	0
29	2	6	9	20	98.00	0
30	2	6	10	20	113.00	0
31	4	2	1	114	17.00	6
32	4	2	2	114	33.00	3
33	4	2	3	114	24.00	4
34	4	2	4	114	17.00	5
35	4	2	5	114	33.00	5
36	4	2	6	114	24.00	4
37	4	2	7	114	40.00	2
38	4	2	8	114	54.00	5
39	4	2	9	114	44.00	4
40	4	2	10	114	99.00	8
41	4	4	1	114	67.00	16
42	4	4	2	114	239.00	85
43	4	4	3	114	220.00	93
44	4	4	4	114	67.00	23
45	4	4	5	114	334.00	84
46	4	4	6	114	239.00	67
47	4	4	7	114	220.00	94
48	4	4	8	114	334.00	18
49	4	4	9	114	220.00	30
50	4	4	10	114	191.00	10
51	4	6	1	114	108.00	181
52	4	6	2	114	326.00	176
53	4	6	3	114	298.00	129
54	4	6	4	114	108.00	49
55	4	6	5	114	520.00	48

56	4	6	6	114	326.00	186
57	4	6	7	114	298.00	134
58	4	6	8	114	426.00	167
59	4	6	9	114	332.00	58
60	4	6	10	114	380.00	326
61	7	2	1	98	33.00	1
62	7	2	2	98	65.00	1
63	7	2	3	98	58.00	1
64	7	2	4	98	33.00	1
65	7	2	5	98	65.00	1
66	7	2	6	98	58.00	0
67	7	2	7	98	90.00	0
68	7	2	8	98	79.00	1
69	7	2	9	98	97.00	1
70	7	2	10	98	122.00	1
71	7	4	1	98	83.00	6
72	7	4	2	98	263.00	24
73	7	4	3	98	241.00	29
74	7	4	4	98	84.00	2
75	7	4	5	98	263.00	7
76	7	4	6	98	241.00	6
77	7	4	7	98	420.00	17
78	7	4	8	98	208.00	3
79	7	4	9	98	392.00	13
80	7	4	10	98	250.00	4
81	7	6	1	98	141.00	19
82	7	6	2	98	380.00	58
83	7	6	3	98	346.00	53
84	7	6	4	98	142.00	8
85	7	6	5	98	382.00	15
86	7	6	6	98	348.00	10
87	7	6	7	98	588.00	45
88	7	6	8	98	539.00	25
89	7	6	9	98	328.00	84
90	7	6	10	98	503.00	99
91	8	2	1	98	41.00	6
92	8	2	2	98	78.00	5
93	8	2	3	98	74.00	6
94	8	2	4	98	42.00	4
95	8	2	5	98	80.00	4
96	8	2	6	98	76.00	4
97	8	2	7	98	106.00	7
98	8	2	8	98	114.00	7
99	8	2	9	98	100.00	5
100	8	2	10	98	162.00	4
101	8	4	1	98	86.00	31
102	8	4	2	98	260.00	34
103	8	4	3	98	236.00	33
104	8	4	4	98	86.00	6
105	8	4	5	98	260.00	4
106	8	4	6	98	236.00	5
107	8	4	7	98	410.00	8
108	8	4	8	98	212.00	9
109	8	4	9	98	218.00	6
110	8	4	10	98	430.00	8
111	8	6	1	98	139.00	27
112	8	6	2	98	373.00	14
113	8	6	3	98	356.00	15
114	8	6	4	98	139.00	6
115	8	6	5	98	603.00	15
116	8	6	6	98	373.00	7
117	8	6	7	98	356.00	9
118	8	6	8	98	598.00	12

119	8	6	9	98	528.00	15
120	8	6	10	98	751.00	17
121	9	2	1	130	45.00	167
122	9	2	2	130	94.00	481
123	9	2	3	130	90.00	448
124	9	2	4	130	50.00	194
125	9	2	5	130	236.00	539
126	9	2	6	130	151.00	349
127	9	2	7	130	143.00	412
128	9	2	8	130	226.00	605
129	9	2	9	130	149.00	107
130	9	2	10	130	195.00	254
131	9	4	1	130	99.00	4496
132	9	4	2	130	279.00	2267
133	9	4	3	130	289.00	5146
134	9	4	4	130	104.00	1604
135	9	4	5	130	468.00	4382
136	9	4	6	130	289.00	7514
137	9	4	7	130	299.00	15211
138	9	4	8	130	480.00	9382
139	9	4	9	130	490.00	5610
140	9	4	10	130	529.00	9041
141	9	6	1	130	180.00	30725
142	9	6	2	130	433.00	7457
143	9	6	3	130	443.00	9550
144	9	6	4	130	188.00	10688
145	9	6	5	130	695.00	16826
146	9	6	6	130	440.00	3955
147	9	6	7	130	450.00	5925
148	9	6	8	130	702.00	13175
149	9	6	9	130	450.00	18051
150	9	6	10	130	1469.00	12334
151	10	2	1	178	59.00	1189
152	10	2	2	178	214.00	1479
153	10	2	3	178	122.00	861
154	10	2	4	178	118.00	1299
155	10	2	5	178	60.00	1355
156	10	2	6	178	128.00	1937
157	10	2	7	178	124.00	2549
158	10	2	8	178	299.00	1789
159	10	2	9	178	174.00	728
160	10	2	10	178	178.00	621
161	10	4	1	178	104.00	5359
162	10	4	2	178	316.00	8861
163	10	4	3	178	285.00	6899
164	10	4	4	178	302.00	2544
165	10	4	5	178	107.00	6551
166	10	4	6	178	288.00	6121
167	10	4	7	178	305.00	2384
168	10	4	8	178	483.00	1680
169	10	4	9	178	489.00	4673
170	10	4	10	178	525.00	9064
171	10	6	1	178	153.00	5955
172	10	6	2	178	338.00	10785
173	10	6	3	178	385.00	4236
174	10	6	4	178	408.00	5935
175	10	6	5	178	153.00	4480
176	10	6	6	178	640.00	6492
177	10	6	7	178	697.00	2609
178	10	6	8	178	460.00	7153
179	10	6	9	178	344.00	5445
180	10	6	10	178	700.00	4743

Appendix C PSO Heuristic Procedures – Results

PSO Heuristic Procedure 1

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	0	7	0	7	0	7	0	7	0
2	14	0	14	0	14	0	14	1	14	0
3	9	0	9	0	9	0	9	0	9	0
4	7	0	7	0	7	0	7	0	7	0
5	16	0	16	0	16	0	16	0	16	0
6	14	1	14	1	14	0	14	0	14	0
7	9	0	9	0	9	0	9	0	9	0
8	16	0	16	0	16	0	16	0	16	0
9	11	0	11	0	11	0	11	0	11	0
10	17	1	17	0	17	0	17	0	17	0
11	18	1	18	0	18	0	18	0	18	1
12	61	0	61	0	61	0	61	0	61	0
13	49	0	49	0	49	0	49	0	49	0
14	48	0	48	0	48	0	48	0	48	0
15	61	0	61	0	61	0	61	0	61	0
16	49	0	49	0	49	0	49	0	49	1
17	92	0	92	0	92	0	92	0	92	0
18	92	0	92	1	92	0	92	0	92	1
19	44	0	44	0	44	0	44	0	44	0
20	54	0	54	0	54	0	54	0	54	0
21	35	0	35	0	35	0	35	0	35	0
22	95	0	95	0	95	0	95	0	95	0
23	82	0	82	1	82	0	82	1	82	1
24	35	0	35	0	35	0	35	0	35	0
25	95	1	95	0	95	0	95	0	95	0
26	82	0	82	0	82	1	82	1	82	0
27	142	0	142	0	142	0	142	0	142	0
28	142	0	142	0	142	0	142	0	142	0
29	98	0	98	0	98	0	98	0	98	1
30	113	0	113	0	113	0	113	1	113	0
31	20	40	19	52	19	31	19	32	18	65
32	37	61	40	31	36	92	39	29	33	44
33	35	31	35	31	38	26	24	28	28	3
34	21	60	25	25	25	79	23	3	25	78
35	36	21	47	78	44	2	37	4	36	71
36	29	40	29	40	34	103	28	4	35	15
37	44	21	44	21	44	21	44	21	44	22
38	76	30	76	29	70	30	68	4	60	21

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
91	42	5	42	5	42	5	42	5	42	6
92	80	208	80	93	79	156	80	86	79	156
93	76	133	75	157	76	112	76	27	76	5
94	42	5	42	5	42	9	42	6	42	6
95	80	5	80	6	80	5	80	5	82	5
96	76	5	76	5	76	5	76	5	78	24
97	111	169	111	141	111	5	111	133	111	79
98	114	5	114	5	114	5	114	6	114	5
99	101	6	101	6	101	5	101	5	101	9
100	164	157	177	143	169	5	177	147	169	223
101	86	6	86	5	86	5	86	7	86	7
102	260	6	260	6	260	6	260	5	260	6
103	236	5	236	6	236	5	236	5	236	6
104	86	5	86	5	86	5	86	5	86	6
105	260	6	260	5	260	6	260	6	260	5
106	236	5	236	6	236	6	236	5	236	5
107	410	5	410	6	410	6	410	5	410	6
108	212	6	212	6	212	5	212	7	212	6
109	218	5	218	5	218	6	218	6	218	6
110	430	5	430	6	430	5	430	6	430	7
111	144	10	145	88	145	91	142	134	144	8
112	386	202	384	69	379	220	379	66	386	143
113	369	202	362	222	367	116	356	91	362	128
114	145	6	145	5	145	179	147	56	145	193
115	632	37	612	133	612	127	612	38	620	180
116	383	247	383	147	386	121	388	161	383	120
117	366	247	369	93	356	41	366	248	356	35
118	630	205	620	118	638	119	612	139	598	155
119	548	37	546	69	537	225	548	146	528	10
120	773	6	774	31	773	6	773	5	773	60
121	57	8	61	245	57	242	57	194	57	70
122	120	9	126	201	115	281	119	113	114	9
123	107	143	122	200	120	235	107	75	121	40
124	64	38	62	66	60	8	60	260	60	8
125	291	112	262	141	291	114	285	48	300	179
126	184	114	163	142	186	50	188	8	173	70
127	176	113	184	116	179	17	184	115	155	149
128	304	251	284	288	302	212	294	308	284	105

39	65	28	65	28	65	28	49	69	54	62	129	190	114	176	142	190	115	200	36	171	221
40	111	21	135	28	143	43	111	30	111	33	130	248	324	260	337	244	102	241	73	212	41
41	67	22	85	3	79	3	77	80	85	3	131	132	105	119	305	111	38	119	138	117	198
42	300	3	300	3	271	4	290	3	258	129	132	324	204	365	166	342	295	348	141	311	9
43	281	3	246	138	259	79	271	3	259	85	133	365	46	345	111	327	87	321	229	321	9
44	67	22	79	126	79	3	81	3	79	3	134	126	169	116	134	122	44	120	37	116	116
45	334	21	334	21	334	21	424	3	424	3	135	577	82	582	245	570	12	584	41	543	41
46	300	3	239	22	279	3	279	4	279	2	136	326	86	365	84	360	48	314	151	340	165
47	281	3	281	3	260	4	260	19	260	3	137	363	197	349	88	377	235	324	338	329	161
48	334	21	394	127	394	3	424	3	334	21	138	642	165	590	89	594	45	546	47	570	42
49	220	22	220	22	220	22	280	3	261	2	139	604	47	583	126	596	124	583	125	623	223
50	253	3	191	22	215	3	215	3	219	3	140	639	84	618	126	619	276	615	337	577	192
51	108	22	118	141	133	2	133	2	133	132	141	220	39	217	74	211	341	201	336	201	48
52	326	22	385	3	359	4	326	22	326	22	142	533	128	482	84	512	5	510	288	485	38
53	298	23	354	130	298	22	298	22	357	3	143	511	213	543	47	552	40	496	292	505	6
54	108	22	134	4	134	3	128	120	128	131	144	204	140	226	286	204	150	202	148	204	186
55	520	23	520	22	609	3	520	22	520	23	145	862	128	799	9	811	7	799	11	834	155
56	326	22	366	142	326	23	326	22	355	80	146	533	48	501	281	488	311	498	162	498	12
57	298	22	298	22	298	23	354	5	354	2	147	515	211	525	154	508	127	498	86	525	155
58	426	22	426	22	530	131	426	23	506	134	148	890	169	898	331	802	7	786	91	782	220
59	332	22	332	22	332	22	410	3	332	23	149	509	117	499	281	515	153	530	44	485	310
60	380	22	380	22	466	130	380	23	380	22	150	1820	302	1864	127	1686	260	1759	89	1775	49
61	43	4	43	175	44	154	38	44	44	61	151	73	183	76	382	68	355	73	73	76	16
62	85	125	86	43	83	88	86	42	76	109	152	279	203	279	187	263	70	253	314	256	387
63	77	5	76	65	78	26	62	85	78	162	153	159	126	173	10	159	442	141	24	147	11
64	41	95	45	98	47	154	43	5	43	23	154	159	70	169	11	141	20	141	449	143	11
65	84	5	83	25	82	48	76	46	84	155	155	86	65	68	371	70	463	76	12	92	10
66	77	4	77	5	71	65	71	24	77	83	156	185	12	188	66	169	61	156	410	158	197
67	126	43	118	5	118	62	110	5	122	22	157	181	11	181	11	167	117	167	12	155	454
68	107	129	105	124	99	46	95	44	103	89	158	386	118	386	119	386	117	365	146	365	88
69	120	25	132	166	120	25	127	4	126	196	159	216	231	204	126	222	279	194	179	208	57
70	159	5	159	120	164	48	168	64	159	165	160	206	417	222	73	220	171	206	300	226	12
71	104	64	101	26	98	94	101	194	98	27	161	130	340	121	372	130	76	121	377	126	56
72	312	156	311	176	311	67	327	203	313	140	162	386	244	412	290	388	122	362	146	390	79
73	290	26	286	110	296	5	290	161	306	87	163	383	60	393	10	343	156	370	78	341	369
74	106	172	104	27	100	4	100	94	100	27	164	382	303	410	10	386	413	363	241	372	306
75	336	89	337	90	337	116	341	70	315	164	165	145	10	137	20	127	182	143	398	125	56
76	301	4	293	162	282	45	309	46	293	26	166	382	360	380	298	393	9	350	356	346	364
77	530	88	530	88	526	131	520	4	436	98	167	397	303	387	239	399	356	397	17	355	172
78	252	111	258	147	253	25	247	67	251	118	168	618	122	613	392	658	10	658	10	658	10
79	476	209	494	213	501	49	476	208	489	170	169	625	306	623	242	586	62	585	150	591	425
80	303	4	318	112	250	96	307	132	301	90	170	661	69	623	70	664	62	623	152	659	19
81	159	134	167	47	165	26	159	85	165	32	171	198	233	202	17	192	434	194	133	193	135
82	482	54	476	186	445	47	445	207	457	26	172	430	213	430	211	445	157	463	86	456	552
83	416	26	435	214	448	54	419	73	368	153	173	509	412	493	356	543	10	525	128	543	9
84	174	5	168	27	166	156	172	46	170	176	174	524	70	566	10	566	10	511	187	528	302
85	450	47	482	191	470	5	454	218	470	117	175	193	454	221	11	203	74	201	181	201	72
86	438	96	431	77	440	172	431	47	423	24	176	827	199	863	130	849	152	861	16	773	71
87	680	133	762	54	700	219	748	224	644	129	177	914	93	914	91	925	126	856	454	890	94
88	678	93	676	5	675	96	544	154	667	194	178	600	187	614	409	612	345	614	399	566	452
89	382	27	416	134	404	117	367	133	392	142	179	462	10	452	366	421	30	425	66	421	220
90	649	97	624	74	628	69	561	162	654	173	180	951	23	895	436	947	363	859	20	841	101

PSO Heuristic Procedure 2

	c1 = c2 = 0.25		c1 = c2 = 0.30		c1 = c2 = 0.35		c1 = c2 = 0.40		c1 = c2 = 0.45	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	11	0	11	0	11	0	11	0	11	0
2	14	0	19	0	19	0	19	0	19	1
3	9	0	14	0	14	0	14	0	14	0
4	11	1	11	0	11	0	11	0	11	0
5	16	0	22	0	22	0	22	0	22	1
6	22	0	22	0	22	0	22	0	22	0
7	17	0	17	0	17	0	17	0	17	0
8	16	0	26	0	26	0	26	0	26	0
9	19	0	19	0	19	0	19	0	19	0
10	17	0	28	0	28	0	28	0	28	0
11	18	0	24	0	24	0	24	0	24	0
12	61	0	73	0	73	0	73	0	73	0
13	49	0	61	0	61	1	61	0	61	0
14	48	0	62	0	62	0	62	0	62	0
15	61	0	77	0	77	0	77	0	77	0
16	49	1	65	0	65	0	65	0	65	0
17	92	0	110	0	110	0	110	0	110	0
18	92	0	114	0	114	0	114	0	114	0
19	44	0	58	0	58	0	58	1	58	0
20	54	0	76	0	76	0	76	0	76	0
21	35	0	43	0	43	0	43	0	43	1
22	95	0	110	1	110	0	110	0	110	0
23	82	0	97	0	97	0	97	0	97	0
24	35	0	47	0	47	0	47	0	47	0
25	95	0	114	0	114	0	114	0	114	0
26	82	0	101	0	101	0	101	0	101	0
27	142	0	164	0	164	0	164	0	164	0
28	142	0	168	0	168	0	168	0	168	0
29	98	0	122	0	122	0	122	0	122	0
30	113	0	143	0	143	0	143	0	143	0
31	31	2	32	1	32	2	32	1	32	1
32	48	68	60	2	60	1	60	2	60	2
33	47	27	51	51	51	52	51	52	51	52
34	31	2	35	2	35	1	35	1	35	1
35	57	54	62	14	62	14	62	14	62	14
36	63	107	55	1	55	2	55	1	55	1
37	78	2	86	66	86	67	86	66	86	67
38	84	1	100	1	100	1	100	1	100	2
39	95	96	104	106	104	108	104	106	104	108
40	187	31	184	15	184	15	184	15	184	15
41	106	2	113	1	113	1	113	2	113	1
42	385	2	457	47	457	49	457	48	457	49

	c1 = c2 = 0.25		c1 = c2 = 0.30		c1 = c2 = 0.35		c1 = c2 = 0.40		c1 = c2 = 0.45	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
91	48	72	43	112	43	114	43	113	43	116
92	90	192	94	19	94	19	94	19	94	19
93	82	51	83	18	83	19	83	19	83	20
94	42	182	52	1	52	2	52	1	52	1
95	94	5	88	18	88	19	88	19	88	19
96	78	75	82	18	82	18	82	19	82	19
97	132	103	121	54	121	55	121	55	121	56
98	114	76	114	19	114	19	114	19	114	20
99	108	76	103	156	103	159	103	159	103	161
100	195	79	193	20	193	21	193	21	193	20
101	92	5	105	2	105	2	105	1	105	1
102	282	32	320	2	320	2	320	2	320	2
103	265	5	296	1	296	2	296	2	296	2
104	100	54	108	20	108	20	108	20	108	20
105	292	5	323	2	323	2	323	2	323	2
106	268	5	299	2	299	2	299	2	299	2
107	459	6	509	2	509	1	509	2	509	2
108	226	5	259	2	259	2	259	2	259	1
109	254	32	266	1	266	2	266	2	266	2
110	508	33	522	2	522	2	522	2	522	2
111	154	61	142	22	142	22	142	22	142	22
112	436	6	447	2	447	2	447	2	447	2
113	397	259	430	2	430	2	430	2	430	2
114	157	32	173	1	173	2	173	1	173	2
115	707	272	710	190	710	188	710	187	710	188
116	435	149	450	2	450	2	450	2	450	2
117	368	178	400	24	400	24	400	24	400	24
118	630	93	728	2	728	2	728	2	728	2
119	650	64	631	96	631	94	631	93	631	94
120	849	35	913	30	913	167	913	166	913	168
121	73	123	81	3	81	2	81	2	81	3
122	158	102	160	270	160	269	160	268	160	269
123	137	199	153	120	153	121	153	120	153	121
124	62	65	88	2	88	2	88	3	88	3
125	394	39	445	3	445	3	445	3	445	3
126	232	138	284	2	284	3	284	3	284	2
127	200	239	263	283	263	283	263	281	263	283
128	342	178	412	35	412	35	412	35	412	35
129	243	71	269	96	269	96	269	95	269	96
130	314	4	287	194	287	194	287	193	287	194
131	139	4	152	181	152	183	152	181	152	181
132	370	111	439	135	439	136	439	135	439	136

43	366	2	367	1	367	1	367	1	367	2	**133**	380	39	439	69	439	69	439	69	439	69
44	107	2	113	1	113	1	113	1	113	2	**134**	138	167	156	182	156	183	156	181	156	183
45	525	2	564	1	564	1	564	1	564	2	**135**	622	341	751	142	751	142	751	142	751	142
46	367	2	386	1	386	2	386	1	386	2	**136**	429	146	393	3	393	3	393	3	393	2
47	348	1	367	1	367	1	367	2	367	2	**137**	403	4	479	136	479	136	479	135	479	136
48	529	2	564	2	564	1	564	2	564	2	**138**	674	228	654	3	654	2	654	2	654	3
49	361	2	364	1	364	2	364	2	364	2	**139**	665	193	723	2	723	2	723	2	723	2
50	304	2	331	1	331	1	331	1	331	1	**140**	711	230	809	108	809	109	809	107	809	108
51	169	63	164	1	164	2	164	2	164	1	**141**	249	146	280	293	280	298	280	94	280	294
52	492	2	492	1	492	1	492	2	492	1	**142**	587	5	633	215	633	218	633	214	633	215
53	464	2	470	51	470	52	470	52	470	52	**143**	596	118	597	285	597	289	597	285	597	285
54	164	2	164	1	164	2	164	2	164	1	**144**	268	108	258	3	258	3	258	2	258	3
55	792	1	792	1	792	1	792	2	792	2	**145**	938	43	990	2	990	3	990	3	990	2
56	492	1	492	1	492	1	492	2	492	2	**146**	616	156	615	179	615	180	615	179	615	179
57	464	1	464	1	464	2	464	2	464	2	**147**	602	80	600	215	600	217	600	214	600	216
58	632	1	696	86	696	88	696	87	696	88	**148**	958	242	990	2	990	3	990	2	990	2
59	500	2	500	1	500	1	500	2	500	1	**149**	673	270	642	179	642	180	642	178	642	180
60	576	1	576	1	576	2	576	2	576	1	**150**	2056	46	2426	239	2426	239	2426	237	2426	238
61	54	58	55	31	55	32	55	32	55	32	**151**	89	146	92	4	92	4	92	4	92	5
62	95	24	110	34	110	35	110	35	110	35	**152**	323	10	333	177	333	178	333	177	333	178
63	84	43	79	1	79	1	79	1	79	1	**153**	179	158	230	127	230	128	230	126	230	127
64	57	22	65	17	65	17	65	17	65	17	**154**	178	10	208	86	208	87	208	86	208	86
65	109	103	105	68	105	71	105	69	105	70	**155**	104	426	92	5	92	4	92	4	92	5
66	93	62	85	1	85	2	85	1	85	1	**156**	203	57	199	4	199	4	199	4	199	4
67	134	24	146	70	146	71	146	71	146	72	**157**	178	9	195	5	195	5	195	4	195	4
68	118	165	107	2	107	1	107	1	107	2	**158**	440	9	443	49	443	49	443	48	443	48
69	148	24	146	19	146	18	146	19	146	19	**159**	264	355	296	172	296	171	296	171	296	173
70	186	109	194	18	194	19	194	19	194	19	**160**	280	208	268	340	268	338	268	344	268	341
71	118	170	118	1	118	1	118	1	118	1	**161**	162	107	182	46	182	46	182	47	182	46
72	348	141	382	159	382	162	382	161	382	163	**162**	432	61	490	416	490	414	490	423	490	416
73	334	95	357	40	357	41	357	41	357	42	**163**	382	9	434	233	434	231	434	237	434	233
74	106	3	120	1	120	1	120	1	120	1	**164**	424	337	410	5	410	4	410	5	410	4
75	361	49	404	81	404	82	404	82	404	83	**165**	161	110	145	5	145	5	145	4	145	5
76	353	140	369	60	369	62	369	62	369	62	**166**	482	435	393	371	393	367	393	377	393	371
77	556	3	660	43	660	43	660	43	660	43	**167**	500	161	410	4	410	4	410	5	410	5
78	278	161	341	176	341	179	341	179	341	179	**168**	738	459	832	388	832	387	832	392	832	388
79	567	75	594	169	594	172	594	171	594	172	**169**	723	397	830	196	830	196	830	197	830	196
80	339	209	360	21	360	22	360	21	360	21	**170**	802	63	895	435	895	437	895	438	895	436
81	190	3	224	153	224	156	224	156	224	157	**171**	243	168	265	274	265	275	265	276	265	274
82	566	27	570	85	570	87	570	87	570	87	**172**	543	121	523	149	523	151	523	151	523	151
83	477	3	476	43	476	45	476	44	476	44	**173**	620	60	596	347	596	348	596	349	596	347
84	202	205	218	21	218	21	218	21	218	21	**174**	550	295	647	152	647	152	647	152	647	151
85	509	100	550	65	550	66	550	65	550	66	**175**	255	220	263	94	263	95	263	95	263	94
86	483	3	546	191	546	193	546	194	546	195	**176**	979	123	921	463	921	465	921	465	921	463
87	860	179	882	178	882	181	882	181	882	182	**177**	976	547	1176	159	1176	159	1176	160	1176	158
88	687	128	761	23	761	24	761	23	761	24	**178**	694	416	734	153	734	154	734	154	734	153
89	440	99	473	104	473	106	473	106	473	108	**179**	513	177	533	439	533	441	533	441	533	439
90	680	151	760	88	760	89	760	89	760	90	**180**	1039	488	1234	415	1234	416	1234	418	1234	415

PSO Heuristic Procedure 3

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	1	7	0	7	0	7	0	7	0
2	14	0	14	0	14	0	14	0	14	0
3	9	0	9	0	9	1	9	0	9	0
4	7	0	7	0	7	1	7	0	7	0
5	16	0	16	0	16	1	16	0	16	0
6	14	0	14	0	14	0	14	0	14	0
7	9	0	9	0	9	0	9	0	9	0
8	16	0	16	0	16	0	16	0	16	1
9	11	0	11	1	11	1	11	0	11	0
10	17	0	17	0	17	0	17	0	17	0
11	18	0	18	0	18	1	18	0	18	0
12	61	0	61	0	61	1	61	0	61	1
13	49	0	49	1	49	0	49	0	49	0
14	48	0	48	0	48	1	48	0	48	0
15	61	0	61	0	61	0	61	0	61	0
16	49	0	49	0	49	0	49	0	49	0
17	92	0	92	0	92	0	92	0	92	0
18	92	0	92	0	92	0	92	0	92	0
19	44	0	44	0	44	0	44	0	44	0
20	54	0	54	0	54	0	54	0	54	0
21	35	0	35	0	35	0	35	0	35	0
22	95	0	95	0	95	0	95	0	95	0
23	82	0	82	0	82	1	82	0	82	0
24	35	0	35	0	35	0	35	0	35	0
25	95	0	95	1	95	0	95	0	95	1
26	82	0	82	0	82	1	82	0	82	0
27	142	0	142	0	142	1	142	0	142	1
28	142	1	142	0	142	0	142	0	142	0
29	98	0	98	0	98	0	98	0	98	0
30	113	0	113	0	113	1	113	0	113	0
31	20	8	20	21	17	8	17	41	17	17
32	40	41	33	13	33	68	33	36	33	4
33	28	95	24	12	24	68	28	38	24	4
34	21	37	17	24	23	3	17	83	17	8
35	33	30	36	20	33	42	33	158	33	22
36	24	24	24	67	32	8	24	45	32	5
37	40	94	44	24	40	87	40	6	45	49
38	54	28	62	19	54	39	62	146	54	102
39	44	4	61	57	44	22	44	21	54	8
40	99	9	99	8	99	25	116	54	99	5
41	80	21	77	163	67	37	77	20	80	3

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
91	42	6	42	9	42	6	42	60	42	5
92	80	6	82	25	80	149	81	4	81	45
93	76	118	78	4	76	170	76	23	76	88
94	42	5	42	3	42	4	42	5	42	78
95	85	4	82	8	80	4	80	4	80	5
96	81	5	76	30	76	4	76	4	76	5
97	111	117	111	27	112	26	111	25	111	65
98	114	26	114	5	114	3	114	4	114	6
99	101	8	101	5	101	5	101	6	101	5
100	169	29	169	27	169	4	169	71	169	111
101	87	5	87	6	87	27	86	6	98	4
102	265	8	260	9	265	5	260	5	265	6
103	241	8	236	9	241	5	236	5	241	5
104	90	7	86	9	90	26	86	27	86	46
105	268	35	265	4	268	6	260	4	260	6
106	244	7	241	4	244	6	236	4	236	6
107	419	32	410	12	496	4	419	6	410	4
108	214	4	214	5	212	27	212	6	242	4
109	221	5	221	6	238	7	218	7	248	4
110	436	4	436	5	496	28	430	7	490	4
111	139	10	142	13	139	31	139	31	142	6
112	373	37	402	106	373	217	373	105	405	7
113	363	40	376	119	356	55	368	29	356	33
114	139	10	151	6	149	5	139	77	151	5
115	632	93	623	7	623	6	623	4	652	161
116	398	63	386	122	391	31	373	188	373	6
117	366	88	356	122	356	117	392	191	356	7
118	620	93	634	38	598	196	620	5	620	80
119	541	62	548	5	544	189	541	7	543	4
120	765	35	762	60	770	254	762	36	762	6
121	53	107	56	9	54	229	51	6	53	6
122	101	7	126	79	105	186	106	10	100	70
123	97	7	100	84	102	8	98	141	97	134
124	56	44	58	38	58	5	52	159	54	98
125	246	237	290	118	243	8	245	261	243	346
126	154	353	154	13	154	12	154	7	196	41
127	146	124	146	12	146	12	146	7	146	210
128	304	115	244	84	262	359	254	7	264	148
129	171	9	168	230	168	233	171	8	168	190
130	243	48	225	5	225	6	220	255	225	8
131	109	8	111	115	104	185	106	320	107	177

42	267	64	239	99	278	4	239	97	278	3	**132**	311	7	319	350	311	168	306	76	311	5
43	220	25	252	151	220	23	220	59	220	91	**133**	327	223	321	7	321	6	321	4	321	5
44	67	27	81	3	81	3	81	100	83	37	**134**	112	8	110	190	112	7	112	5	112	5
45	374	24	394	43	394	4	394	5	376	76	**135**	479	130	523	6	543	293	512	6	523	195
46	239	86	279	4	279	3	279	4	239	40	**136**	323	14	306	45	314	4	314	5	314	4
47	260	4	260	4	246	79	260	4	260	3	**137**	333	14	324	113	324	5	316	75	324	4
48	334	86	384	82	399	21	374	169	334	23	**138**	530	131	526	8	526	5	568	78	524	121
49	261	6	258	77	220	58	245	46	259	4	**139**	618	354	576	264	541	98	550	5	550	5
50	215	4	215	3	191	61	215	3	215	3	**140**	577	8	577	7	577	5	573	232	577	5
51	108	46	118	112	108	7	108	6	127	21	**141**	195	219	187	203	193	50	199	122	187	210
52	355	50	326	23	326	147	326	45	326	44	**142**	494	303	482	129	470	139	462	257	481	47
53	354	104	357	22	339	181	354	138	298	46	**143**	481	6	471	254	492	84	514	6	495	166
54	108	25	118	114	128	55	134	3	108	124	**144**	198	49	190	88	200	6	190	6	198	150
55	520	136	520	169	596	63	520	128	596	44	**145**	760	12	740	180	791	177	763	95	779	47
56	326	112	326	46	365	145	326	79	362	32	**146**	539	90	473	16	519	130	473	138	473	10
57	298	9	338	117	336	86	358	122	320	84	**147**	486	358	478	62	482	290	510	288	483	11
58	426	140	495	64	426	64	448	84	426	62	**148**	746	91	786	270	802	346	766	59	782	49
59	368	70	344	125	332	85	332	129	344	142	**149**	503	174	514	291	485	168	467	218	472	238
60	445	25	422	63	445	126	464	184	380	173	**150**	1634	7	1634	225	1720	8	1634	6	1634	6
61	35	172	42	4	42	3	36	61	41	41	**151**	67	344	67	102	61	146	66	399	62	222
62	68	194	68	185	68	89	76	6	68	69	**152**	246	512	226	162	226	206	258	263	253	11
63	61	76	61	94	61	124	61	102	61	144	**153**	131	269	144	109	132	441	146	370	153	10
64	37	89	39	40	33	4	33	3	33	146	**154**	139	15	147	362	141	51	127	363	137	296
65	68	190	81	172	75	126	65	124	68	11	**155**	68	11	68	10	68	52	74	182	68	49
66	68	10	71	146	61	166	71	4	61	11	**156**	149	128	146	9	133	204	137	217	149	200
67	106	73	110	109	114	4	118	167	106	43	**157**	133	319	142	9	136	197	139	148	129	154
68	93	34	91	79	91	118	83	50	91	150	**158**	313	290	313	177	358	8	347	9	342	154
69	115	60	127	194	101	50	116	4	115	46	**159**	198	230	194	17	198	146	212	53	194	10
70	129	81	152	91	129	138	145	144	129	46	**160**	202	56	206	409	202	105	206	285	212	441
71	83	10	95	48	83	109	95	4	95	4	**161**	111	420	120	467	119	9	134	245	112	466
72	269	60	269	25	309	143	305	188	269	71	**162**	348	511	360	65	362	471	336	329	358	17
73	285	30	318	3	241	80	269	73	288	71	**163**	346	86	343	341	343	235	351	429	315	20
74	84	102	104	130	90	53	86	115	100	5	**164**	380	131	335	506	360	322	332	72	332	20
75	263	165	263	216	272	121	302	96	263	27	**165**	123	349	125	316	125	14	113	384	117	56
76	260	176	250	72	264	169	277	187	248	69	**166**	318	85	357	502	353	415	363	342	359	263
77	452	137	468	29	420	84	484	32	420	26	**167**	342	497	363	71	354	332	364	10	357	176
78	243	99	208	169	208	189	209	4	247	200	**168**	587	278	530	21	576	184	573	77	536	399
79	481	114	482	54	392	57	458	77	447	169	**169**	623	546	581	9	568	298	585	166	581	10
80	250	109	250	122	293	142	294	53	250	7	**170**	603	244	605	224	558	359	605	343	614	166
81	160	189	146	29	151	93	142	165	142	26	**171**	183	393	184	122	181	14	181	72	184	162
82	400	89	400	215	382	31	427	31	380	152	**172**	373	518	399	76	371	71	394	444	405	70
83	419	29	374	7	410	150	361	8	433	118	**173**	427	194	463	189	459	15	439	400	473	402
84	142	114	142	191	142	174	142	5	144	118	**174**	489	491	448	277	482	15	462	172	506	66
85	489	59	433	157	390	221	382	104	470	225	**175**	157	264	191	12	179	445	179	14	181	402
86	362	85	434	82	366	225	348	107	348	175	**176**	768	247	751	324	750	9	760	12	766	80
87	588	185	676	184	658	106	680	85	588	105	**177**	823	491	800	535	800	190	784	73	823	120
88	560	36	586	211	608	38	597	113	554	158	**178**	582	394	552	14	544	123	552	11	544	15
89	368	5	333	82	333	182	392	201	333	223	**179**	443	439	406	7	387	302	417	234	401	181
90	517	7	580	29	514	174	514	10	591	186	**180**	855	376	841	505	841	13	804	123	889	132

PSO Heuristic Procedure 4

Problem #	$c_1 = c_2 = 0.25$ Min. Cost	$c_1 = c_2 = 0.25$ Total Time in Seconds	$c_1 = c_2 = 0.30$ Min. Cost	$c_1 = c_2 = 0.30$ Total Time in Seconds	$c_1 = c_2 = 0.35$ Min. Cost	$c_1 = c_2 = 0.35$ Total Time in Seconds	$c_1 = c_2 = 0.40$ Min. Cost	$c_1 = c_2 = 0.40$ Total Time in Seconds	$c_1 = c_2 = 0.45$ Min. Cost	$c_1 = c_2 = 0.45$ Total Time in Seconds
1	7	1	7	0	7	0	7	1	7	0
2	14	0	14	0	14	0	14	0	14	0
3	9	0	9	0	9	0	9	0	9	0
4	7	0	7	0	7	0	7	0	7	0
5	16	0	16	0	16	0	16	0	16	0
6	14	1	14	1	14	0	14	0	14	1
7	9	0	9	0	9	0	9	0	9	0
8	16	0	16	0	16	0	16	0	16	0
9	11	0	11	0	11	1	11	0	11	0
10	17	1	17	0	17	1	17	1	17	0
11	18	0	18	0	18	0	18	0	18	0
12	61	0	61	0	61	0	61	0	61	1
13	49	0	49	0	49	1	49	0	49	0
14	48	0	48	1	48	0	48	1	48	1
15	61	0	61	1	61	0	61	1	61	1
16	49	0	49	0	49	0	49	0	49	0
17	92	0	92	0	92	0	92	1	92	0
18	92	0	92	0	92	0	92	0	92	1
19	44	1	44	0	44	0	44	1	44	0
20	54	1	54	0	54	1	54	0	54	0
21	35	1	35	1	35	1	35	0	35	0
22	95	0	95	0	95	0	95	1	95	0
23	82	0	82	1	82	0	82	0	82	1
24	35	1	35	1	35	0	35	1	35	1
25	95	0	95	0	95	0	95	0	95	1
26	82	0	82	0	82	1	82	1	82	0
27	142	0	142	0	142	0	142	0	142	1
28	142	0	142	1	142	0	142	1	142	0
29	98	1	98	0	98	1	98	0	98	0
30	113	0	113	1	113	0	113	0	113	0
31	20	79	21	159	24	65	25	116	17	94
32	36	171	40	137	47	315	47	56	40	17
33	38	287	27	285	31	12	27	152	38	214
34	21	224	25	98	23	85	19	9	21	34
35	38	239	38	139	47	80	38	246	39	270
36	38	31	38	165	38	6	29	142	32	71
37	45	71	44	233	60	246	44	15	44	82
38	62	160	58	274	74	219	60	10	78	183
39	49	326	66	78	66	233	49	158	49	93
40	143	70	111	83	143	166	111	44	116	10
41	75	40	77	7	79	7	80	6	78	138

Problem #	$c_1 = c_2 = 0.25$ Min. Cost	$c_1 = c_2 = 0.25$ Total Time in Seconds	$c_1 = c_2 = 0.30$ Min. Cost	$c_1 = c_2 = 0.30$ Total Time in Seconds	$c_1 = c_2 = 0.35$ Min. Cost	$c_1 = c_2 = 0.35$ Total Time in Seconds	$c_1 = c_2 = 0.40$ Min. Cost	$c_1 = c_2 = 0.40$ Total Time in Seconds	$c_1 = c_2 = 0.45$ Min. Cost	$c_1 = c_2 = 0.45$ Total Time in Seconds
91	42	12	42	11	42	7	42	9	43	6
92	80	15	80	18	80	118	80	11	80	11
93	75	142	76	55	75	198	76	353	76	91
94	42	14	44	7	44	45	44	6	42	11
95	82	6	80	7	80	53	80	9	80	7
96	78	6	76	7	78	7	76	8	76	10
97	111	227	111	13	108	284	111	91	111	356
98	114	11	114	11	114	83	114	13	114	7
99	101	8	101	10	103	7	101	10	101	7
100	169	391	169	293	177	349	169	258	169	89
101	86	13	86	13	92	6	92	7	86	10
102	266	10	266	14	260	23	260	9	265	76
103	242	10	242	14	236	25	236	9	238	18
104	86	17	92	7	86	11	92	7	92	7
105	275	10	260	13	267	13	265	9	260	7
106	251	9	236	13	251	55	241	9	236	8
107	410	9	420	8	410	7	412	10	420	10
108	217	12	224	13	212	8	214	7	222	20
109	218	12	218	10	233	7	218	10	221	11
110	440	10	460	6	460	7	430	17	430	18
111	142	228	147	421	145	6	139	416	143	414
112	373	61	379	12	393	444	373	17	400	347
113	369	268	362	11	384	6	356	16	368	56
114	139	102	145	95	143	183	153	310	145	52
115	603	434	603	188	603	62	603	12	633	157
116	386	301	373	316	373	396	373	158	388	60
117	381	71	356	171	378	7	368	57	371	258
118	598	111	612	111	626	6	598	309	638	112
119	546	185	528	171	541	11	541	369	544	101
120	785	12	773	312	751	263	785	160	774	169
121	53	15	58	88	58	334	58	141	60	13
122	119	222	111	161	115	13	119	85	122	321
123	108	166	120	160	111	13	122	279	128	319
124	60	376	56	345	64	288	62	494	58	83
125	282	571	307	459	295	75	309	421	300	654
126	184	18	198	80	197	606	189	616	200	588
127	189	558	187	351	189	72	180	364	189	335
128	294	386	282	424	310	210	304	545	294	94
129	193	388	200	343	203	90	190	327	196	614
130	225	427	244	555	238	73	264	9	251	12
131	122	70	133	9	124	75	114	72	119	26

42	275	203	278	7	278	157	278	9	278	204	**132**	362	465	359	270	378	595	354	503	350	288
43	259	8	259	6	259	6	246	245	259	11	**133**	367	159	370	87	386	282	374	159	385	231
44	79	13	79	72	81	267	79	9	81	9	**134**	132	11	120	139	128	11	130	9	116	125
45	394	18	384	345	395	15	384	78	394	7	**135**	602	292	591	167	582	10	567	12	571	99
46	279	48	271	79	275	12	279	6	279	7	**136**	392	379	353	510	336	169	336	156	321	620
47	260	12	260	10	260	104	260	7	260	7	**137**	365	696	356	79	382	370	356	366	352	99
48	394	12	384	11	394	108	389	42	384	199	**138**	570	14	606	10	630	615	626	686	576	165
49	247	87	259	191	254	171	259	10	258	73	**139**	595	243	589	388	603	598	583	538	583	595
50	215	8	215	9	215	7	215	77	215	6	**140**	693	400	700	702	615	532	577	407	603	543
51	130	178	128	148	120	7	118	335	118	249	**141**	211	86	209	149	208	512	209	363	211	354
52	385	115	382	311	382	136	382	117	401	240	**142**	524	263	495	640	554	694	489	719	521	173
53	339	318	331	240	331	128	354	7	357	222	**143**	544	19	523	335	560	224	552	298	522	247
54	118	288	118	228	118	319	134	9	118	70	**144**	202	97	214	437	204	14	220	221	194	489
55	578	11	604	12	609	7	604	10	556	237	**145**	831	406	862	19	799	629	873	404	804	252
56	382	259	364	159	389	7	367	13	382	11	**146**	493	472	488	13	528	616	492	547	507	321
57	332	155	336	317	354	8	332	233	339	122	**147**	587	87	487	157	571	224	561	469	567	309
58	506	10	466	130	501	231	437	117	466	48	**148**	868	660	866	334	812	477	870	238	862	245
59	405	109	392	323	368	237	350	82	362	18	**149**	550	389	482	329	502	249	527	433	522	613
60	422	246	422	351	422	208	450	221	450	154	**150**	1750	20	1760	576	1776	655	1892	326	1760	16
61	44	305	42	43	38	113	44	182	44	41	**151**	80	32	78	676	69	402	77	323	70	104
62	86	172	82	187	77	44	79	80	84	264	**152**	267	124	273	170	274	314	298	19	278	32
63	73	244	79	343	80	189	79	4	75	109	**153**	153	35	165	230	172	578	147	532	157	277
64	49	4	45	11	47	37	45	155	47	276	**154**	142	305	165	920	141	29	168	604	152	55
65	83	152	85	77	96	35	84	16	88	122	**155**	78	16	70	19	92	100	86	19	76	32
66	82	214	85	5	81	97	66	203	78	219	**156**	155	15	148	31	172	632	170	247	164	842
67	122	8	126	286	126	5	114	286	114	118	**157**	175	388	144	33	160	27	161	347	173	21
68	101	185	105	135	109	40	95	245	103	45	**158**	389	469	401	142	408	722	395	239	387	760
69	119	189	129	287	134	267	129	46	129	314	**159**	240	325	248	146	220	872	220	507	222	389
70	164	193	159	88	159	209	152	7	164	97	**160**	206	37	222	244	232	21	216	45	226	32
71	107	220	97	50	105	155	101	169	102	122	**161**	138	16	131	650	139	537	124	241	136	664
72	318	51	324	135	316	50	302	262	331	145	**162**	384	832	360	257	410	137	416	899	392	665
73	287	186	299	47	311	166	273	93	277	94	**163**	374	251	370	124	378	15	378	854	371	635
74	102	160	116	308	102	48	102	7	104	41	**164**	383	124	377	161	395	18	368	19	380	485
75	329	145	323	53	306	144	332	272	304	354	**165**	139	108	131	547	139	100	135	572	145	17
76	310	7	263	393	292	58	307	228	293	52	**166**	351	853	346	880	374	18	360	938	380	670
77	526	99	504	230	542	187	488	241	516	265	**167**	383	781	407	118	394	319	413	335	403	33
78	263	94	252	92	268	392	248	356	242	172	**168**	631	926	617	961	624	16	615	149	647	597
79	518	190	496	359	489	240	490	249	514	48	**169**	623	24	591	925	652	604	591	1016	630	519
80	304	370	312	306	294	304	263	82	297	356	**170**	632	490	624	43	673	293	606	21	694	400
81	175	8	176	136	173	367	178	6	171	382	**171**	210	119	202	757	219	472	199	715	205	789
82	455	319	444	93	448	145	451	277	400	10	**172**	456	182	484	28	466	821	479	18	432	451
83	457	240	416	317	411	139	461	279	433	366	**173**	523	740	517	426	518	234	510	815	490	1073
84	174	95	166	14	168	218	178	187	178	88	**174**	478	351	516	25	553	1087	513	25	524	337
85	459	331	456	143	461	14	464	55	465	7	**175**	215	124	211	877	231	853	203	323	219	211
86	463	224	423	325	424	110	493	5	438	141	**176**	820	132	867	244	833	837	797	657	847	1059
87	724	97	728	193	680	103	742	361	766	285	**177**	894	761	905	324	932	442	954	434	921	509
88	672	238	695	192	666	284	665	236	658	420	**178**	624	678	570	751	598	817	588	494	606	25
89	379	181	390	8	400	282	401	227	411	414	**179**	460	459	458	887	444	131	433	585	421	319
90	629	232	630	428	634	281	636	383	632	12	**180**	934	357	965	255	978	138	925	781	971	617

PSO Heuristic Procedure 5

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$			$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	0	7	1	7	0	7	1	7	0	91	44	3	43	17	45	75	42	16	46	18
2	14	0	14	0	14	1	14	0	14	0	92	80	64	83	18	82	141	82	65	82	18
3	9	0	9	0	9	0	9	0	9	0	93	79	109	77	92	81	34	86	3	76	125
4	7	0	7	0	7	0	7	0	7	0	94	48	3	44	118	46	2	52	3	48	3
5	16	0	16	0	16	0	16	0	16	0	95	80	80	85	81	86	2	94	3	80	141
6	14	0	14	0	14	0	14	0	14	0	96	76	50	76	63	82	3	80	18	78	18
7	9	0	9	0	9	0	9	0	9	0	97	118	3	113	34	115	48	118	51	122	3
8	16	0	16	0	16	0	16	0	16	0	98	114	49	114	49	114	131	120	19	120	3
9	11	0	11	0	11	0	11	0	11	0	99	101	31	105	110	105	3	101	127	103	66
10	17	0	17	0	17	0	17	0	17	0	100	178	134	169	102	178	134	177	53	196	153
11	18	0	18	0	18	0	18	0	18	0	101	86	3	86	3	98	2	100	2	100	19
12	61	0	61	0	61	0	61	0	61	0	102	309	3	283	3	320	2	320	2	309	21
13	49	1	49	0	49	0	49	0	49	0	103	285	3	259	3	296	2	296	2	285	3
14	48	0	48	0	48	0	48	0	48	0	104	102	2	98	3	104	3	102	3	104	3
15	61	0	61	0	61	0	61	0	61	0	105	310	2	305	3	323	3	323	2	310	21
16	49	0	49	0	49	0	49	0	49	0	106	286	2	281	3	299	3	259	21	286	3
17	92	0	92	1	92	0	92	0	92	0	107	495	3	472	2	509	3	509	3	495	3
18	92	1	92	0	92	0	92	0	92	0	108	212	3	212	3	241	3	229	20	241	2
19	44	0	44	0	44	0	44	0	44	0	109	218	2	218	3	247	2	253	3	248	2
20	54	0	54	0	54	0	54	0	54	1	110	430	3	430	3	486	3	498	3	486	2
21	35	0	35	0	35	0	35	0	35	0	111	149	91	157	3	153	38	165	75	162	145
22	95	0	95	0	95	0	95	0	95	0	112	406	22	413	100	447	3	447	3	406	3
23	82	0	82	0	82	0	82	0	82	1	113	389	2	381	100	430	3	407	41	443	62
24	35	0	35	0	41	0	35	0	35	0	114	143	38	151	92	163	3	173	3	149	3
25	95	0	95	0	95	0	95	0	95	0	115	682	107	620	63	709	24	667	187	612	84
26	82	0	82	0	82	0	82	0	82	0	116	395	60	422	43	391	121	418	82	424	23
27	142	0	142	1	142	0	142	0	142	1	117	366	139	429	23	409	178	424	22	390	22
28	142	0	142	0	142	0	142	0	142	0	118	646	146	658	104	696	23	702	103	662	144
29	98	0	98	0	98	0	98	0	98	0	119	548	24	618	64	528	143	543	124	585	64
30	113	0	113	0	113	0	113	0	113	0	120	785	25	851	126	849	44	773	45	773	65
31	22	55	24	57	22	99	24	80	19	56	121	59	162	75	70	61	4	73	49	69	90
32	39	24	40	105	40	82	45	96	44	105	122	126	27	129	222	126	50	123	198	120	146
33	27	69	48	3	31	82	31	25	37	47	123	116	77	97	170	124	26	126	30	116	99
34	17	3	23	23	19	14	25	35	27	3	124	80	116	58	167	66	118	66	25	68	26
35	48	49	43	26	54	3	49	27	60	83	125	306	217	317	136	308	162	331	212	359	82
36	34	2	38	3	32	47	39	97	38	2	126	196	159	193	132	223	199	187	5	204	55
37	57	3	49	61	49	2	60	27	60	26	127	200	233	195	27	176	201	250	233	194	177
38	68	27	74	49	60	50	68	63	80	3	128	284	163	316	33	312	29	294	240	310	210
39	61	3	66	15	58	89	92	75	59	14	129	206	104	172	28	203	132	190	207	184	80
40	127	90	124	26	124	41	128	40	164	105	130	254	165	250	163	259	132	263	31	298	234
41	80	3	80	3	93	3	77	3	80	2	131	127	29	129	55	132	177	121	207	132	51

42	295	4	280	31	278	3	360	3	280	136	**132**	329	120	387	3	317	83	355	4	355	245
43	276	4	259	3	259	3	261	74	259	3	**133**	373	141	316	143	366	251	354	59	366	194
44	77	55	99	4	93	3	77	2	77	40	**134**	112	2	138	4	152	30	126	29	134	3
45	374	87	385	90	385	123	384	3	435	139	**135**	628	63	548	232	593	87	573	257	571	181
46	270	120	279	3	279	3	361	3	279	2	**136**	369	61	353	63	351	166	314	194	374	55
47	301	3	260	3	290	129	342	3	260	3	**137**	335	198	352	87	384	3	340	137	387	110
48	399	3	399	3	464	2	419	137	399	3	**138**	584	262	574	260	616	31	558	232	616	3
49	220	45	277	35	261	3	251	73	298	103	**139**	582	186	595	118	598	60	645	117	718	211
50	259	124	215	3	215	3	248	3	215	3	**140**	681	218	693	181	580	58	637	87	658	31
51	133	32	127	3	124	132	127	47	133	2	**141**	226	85	211	58	221	159	217	4	240	191
52	362	65	382	3	372	18	377	126	385	108	**142**	493	120	494	62	487	35	604	89	519	207
53	357	3	341	17	357	3	331	52	429	92	**143**	491	162	522	63	518	240	535	36	534	237
54	144	3	120	98	152	17	120	2	134	3	**144**	212	4	228	87	230	3	228	111	214	98
55	701	3	673	148	672	87	596	140	609	65	**145**	819	3	851	98	812	33	862	95	836	33
56	365	32	392	92	449	125	389	5	372	50	**146**	581	150	519	184	519	125	517	33	567	60
57	369	142	408	3	354	3	361	6	408	127	**147**	583	274	507	69	504	266	506	234	571	61
58	503	19	466	4	463	55	503	39	536	82	**148**	878	33	880	279	802	123	834	4	900	3
59	387	35	387	61	398	115	387	65	386	20	**149**	567	93	572	61	523	277	522	215	556	3
60	445	64	536	139	536	18	445	21	466	3	**150**	1874	5	1840	130	1820	164	1960	3	1774	290
61	42	54	43	93	45	119	47	97	40	93	**151**	80	267	81	198	82	75	67	74	81	134
62	89	58	82	114	88	16	83	60	85	88	**152**	250	125	226	295	271	247	280	169	274	15
63	85	111	79	72	78	86	79	2	88	30	**153**	156	185	160	244	166	40	150	44	155	15
64	45	107	43	2	51	54	59	16	43	55	**154**	163	8	162	72	153	77	165	112	151	115
65	92	72	88	45	88	16	107	134	85	72	**155**	70	6	82	75	90	6	72	5	88	99
66	76	29	85	2	97	30	93	32	96	30	**156**	164	183	182	111	158	82	202	146	166	15
67	118	103	118	104	138	31	126	2	124	60	**157**	152	14	170	144	159	175	164	10	162	49
68	111	116	122	16	118	120	127	91	107	31	**158**	353	316	402	42	399	5	365	10	394	201
69	127	60	135	60	131	75	144	32	149	17	**159**	208	10	236	224	224	45	254	5	254	5
70	175	77	164	134	158	48	171	108	182	48	**160**	218	42	206	111	246	144	262	223	272	75
71	99	119	118	2	111	47	125	133	120	136	**161**	134	327	147	41	140	217	135	247	137	42
72	341	2	333	85	387	1	375	36	351	86	**162**	408	48	384	7	416	208	372	49	444	217
73	299	68	311	85	336	69	333	155	346	102	**163**	339	127	380	50	375	7	394	47	365	281
74	116	32	126	120	118	63	112	19	118	137	**164**	392	353	359	198	399	46	386	362	380	6
75	365	2	390	35	354	102	314	123	372	86	**165**	133	148	129	6	139	6	129	6	129	6
76	339	19	321	136	314	2	291	119	363	68	**166**	368	160	349	202	356	165	393	210	347	353
77	532	72	526	72	618	72	508	57	604	19	**167**	400	45	420	327	410	130	381	6	381	6
78	273	148	247	149	258	151	268	103	262	100	**168**	630	176	623	207	668	136	591	338	602	131
79	503	37	532	161	489	91	526	55	489	72	**169**	653	126	729	88	639	7	716	216	597	84
80	313	136	327	69	307	36	360	53	341	19	**170**	627	295	622	174	674	6	705	305	664	330
81	164	3	185	131	175	65	193	130	181	2	**171**	194	44	185	156	191	215	211	6	185	375
82	446	37	453	73	545	160	538	20	506	159	**172**	474	168	415	6	486	6	464	131	466	332
83	444	160	477	160	512	143	489	125	482	125	**173**	508	348	555	329	535	217	565	172	526	47
84	202	67	184	82	202	50	190	146	204	65	**174**	509	50	517	263	537	48	563	130	549	7
85	466	20	552	2	452	73	529	1	508	38	**175**	217	82	185	7	201	88	197	196	207	358
86	485	161	508	72	502	2	499	20	501	161	**176**	750	344	825	259	751	242	799	217	831	367
87	822	78	786	93	818	20	824	39	822	131	**177**	845	6	830	276	959	327	838	137	899	101
88	649	2	640	111	655	2	680	166	850	149	**178**	596	7	628	90	638	7	604	95	630	270
89	390	54	439	2	406	2	431	89	473	124	**179**	457	129	462	91	460	354	419	261	418	330
90	717	112	693	94	681	38	614	2	677	93	**180**	983	226	980	91	855	93	922	50	992	263

PSO Heuristic Procedure 6

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$			$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	0	7	0	7	0	7	0	7	1	91	42	4	42	8	42	7	42	6	42	4
2	14	0	14	0	14	0	14	1	14	0	92	81	4	81	5	80	6	80	30	80	7
3	9	0	9	0	9	0	9	0	9	0	93	76	53	76	26	76	6	76	113	76	51
4	7	1	7	0	7	0	7	0	7	0	94	42	4	42	6	42	7	44	4	42	8
5	16	0	16	0	16	0	16	0	16	0	95	80	6	80	9	82	5	80	4	80	5
6	14	1	14	0	14	0	14	0	14	0	96	76	6	76	9	76	25	76	4	76	5
7	9	0	9	0	9	0	9	0	9	0	97	111	28	111	27	111	49	111	117	111	30
8	16	1	16	0	16	0	16	0	16	0	98	114	5	114	5	114	7	114	6	114	6
9	11	0	11	0	11	0	11	0	11	0	99	101	27	101	8	101	5	101	4	101	8
10	17	1	17	0	17	0	17	0	17	0	100	169	5	169	4	162	6	169	31	169	85
11	18	0	18	0	18	0	18	0	18	0	101	89	9	89	6	86	9	86	9	88	31
12	61	1	61	0	61	0	61	0	61	0	102	262	12	265	8	265	12	272	5	266	6
13	49	0	49	0	49	1	49	0	49	1	103	238	12	241	8	241	12	248	5	242	6
14	48	0	48	0	48	0	48	0	48	0	104	94	7	88	8	86	11	86	4	86	9
15	61	0	61	0	61	0	61	0	61	0	105	265	37	265	9	260	7	270	6	267	7
16	49	0	49	1	49	1	49	0	49	0	106	243	9	241	9	236	6	246	6	243	7
17	92	1	92	0	92	0	92	0	92	0	107	418	20	410	5	425	5	431	4	417	5
18	92	0	92	0	92	0	92	0	92	0	108	212	42	216	7	229	5	216	5	223	4
19	44	0	44	0	44	0	44	0	44	1	109	223	6	223	8	236	5	218	4	223	7
20	54	0	54	1	54	1	54	0	54	0	110	440	35	430	5	436	4	430	5	430	7
21	35	0	35	0	35	0	35	0	35	0	111	139	94	139	94	139	60	144	103	143	12
22	95	0	95	0	95	0	95	1	95	0	112	379	119	394	4	373	189	383	7	401	191
23	82	0	82	0	82	0	82	0	82	0	113	368	257	374	97	384	63	356	129	369	13
24	35	0	35	0	35	1	35	0	35	0	114	139	147	145	109	147	55	145	111	145	35
25	95	0	95	0	95	0	95	0	95	0	115	603	66	612	132	603	6	623	100	633	103
26	82	1	82	0	82	0	82	0	82	1	116	373	35	386	6	410	150	397	257	404	35
27	142	0	142	0	142	0	142	1	142	0	117	366	36	356	68	369	92	397	94	369	165
28	142	0	142	0	142	0	142	0	142	0	118	598	7	638	4	638	156	598	101	598	121
29	98	1	98	0	98	0	98	0	98	0	119	544	4	528	154	551	8	543	43	560	11
30	113	0	113	0	113	0	113	0	113	0	120	751	6	751	44	770	97	751	6	802	74
31	20	151	24	150	23	6	23	84	23	59	121	50	215	53	256	55	14	57	77	58	235
32	37	39	47	130	41	121	39	20	39	41	122	120	241	125	13	117	6	124	235	135	265
33	38	4	27	171	27	27	31	164	35	170	123	122	117	120	280	115	150	122	197	107	47
34	23	109	25	112	25	36	23	20	23	37	124	66	185	66	158	56	40	64	6	62	7
35	47	129	39	148	44	156	38	141	41	21	125	316	134	308	288	285	247	300	260	267	248
36	27	5	34	7	38	107	32	103	38	4	126	200	314	190	218	197	6	188	117	200	214
37	50	108	53	94	50	4	44	24	60	58	127	185	286	164	231	174	46	195	321	183	293
38	68	6	72	110	78	60	68	4	60	76	128	304	169	294	215	294	81	284	9	304	344
39	59	24	66	54	59	41	66	75	52	24	129	203	118	188	239	222	6	200	10	198	6
40	124	118	111	4	131	58	123	76	131	139	130	260	333	250	8	261	241	264	380	258	12
41	79	6	79	4	79	8	79	8	79	6	131	122	49	113	207	124	199	120	196	113	273

42	278	7	258	7	278	4	278	4	278	6	**132**	323	94	323	392	323	87	375	163	347	6
43	259	128	239	6	259	123	259	4	259	6	**133**	389	174	382	138	326	8	327	165	357	56
44	77	5	79	6	79	4	79	4	79	4	**134**	126	7	142	44	112	119	130	180	116	42
45	365	134	394	8	384	115	394	150	385	73	**135**	594	137	591	315	598	141	584	93	531	232
46	279	4	279	4	279	4	279	5	279	4	**136**	320	295	369	273	314	134	371	226	377	50
47	260	4	260	4	260	4	260	5	260	4	**137**	348	96	377	302	363	43	324	358	371	256
48	394	6	394	5	399	150	399	4	394	7	**138**	536	405	572	57	594	102	618	274	526	97
49	247	155	259	5	259	6	261	6	261	6	**139**	638	8	534	194	598	141	606	351	550	221
50	215	5	215	5	215	4	215	4	215	5	**140**	675	261	618	6	684	143	635	11	699	416
51	124	166	128	7	133	145	120	87	127	136	**141**	203	93	215	58	201	137	201	258	195	300
52	359	197	385	4	369	178	385	5	382	7	**142**	559	175	581	47	554	305	557	317	485	273
53	354	6	357	4	354	4	354	179	354	6	**143**	495	289	503	281	563	5	569	426	522	59
54	134	5	118	114	128	4	120	143	118	91	**144**	202	136	200	54	208	44	212	10	226	214
55	609	6	596	5	556	79	603	209	556	31	**145**	867	193	862	202	882	369	855	14	871	237
56	366	138	389	26	382	116	382	7	382	198	**146**	546	262	541	250	488	139	535	178	533	416
57	339	138	354	89	339	200	358	154	338	6	**147**	575	273	503	23	555	271	547	379	547	317
58	501	4	501	81	530	4	498	96	506	163	**148**	872	280	826	283	890	266	802	424	786	188
59	387	6	381	81	392	74	387	25	383	157	**149**	482	294	482	105	554	278	542	135	502	102
60	450	27	450	5	415	126	450	7	443	122	**150**	1788	242	1722	450	1776	163	1840	59	1810	107
61	42	68	44	48	44	190	41	24	42	22	**151**	69	25	77	20	82	268	72	312	71	175
62	85	165	85	47	85	50	76	132	77	164	**152**	285	193	285	17	256	22	282	17	321	10
63	75	54	78	206	75	69	69	31	78	50	**153**	169	18	149	233	167	65	162	341	155	73
64	47	126	45	24	43	4	49	5	47	71	**154**	146	231	141	168	141	149	162	11	139	283
65	89	183	81	217	91	181	94	88	87	24	**155**	82	12	90	399	88	314	82	387	86	365
66	89	90	80	6	81	6	80	116	72	74	**156**	163	9	160	10	160	79	165	120	156	297
67	122	5	128	47	114	5	126	115	118	28	**157**	161	299	159	294	151	142	160	289	174	250
68	105	165	104	26	105	4	105	162	105	65	**158**	382	352	354	427	402	116	374	359	374	504
69	125	98	121	164	125	166	133	25	120	29	**159**	228	15	236	406	232	452	224	128	222	15
70	167	27	159	145	156	97	162	221	152	215	**160**	206	13	246	306	210	182	222	12	206	246
71	103	106	92	75	105	99	101	168	104	198	**161**	138	194	132	14	133	13	126	472	130	14
72	309	59	328	210	338	127	331	185	307	193	**162**	398	330	360	135	360	121	372	72	370	423
73	289	88	306	245	313	112	288	168	302	5	**163**	351	76	365	208	371	71	369	136	368	335
74	110	48	104	197	108	74	108	72	110	98	**164**	387	575	382	440	389	369	364	76	382	154
75	344	243	306	183	323	186	321	239	304	216	**165**	125	30	137	343	137	24	151	60	143	372
76	310	8	297	219	287	34	292	127	282	167	**166**	375	336	349	566	389	261	374	497	354	246
77	516	33	540	148	508	11	484	67	530	107	**167**	384	440	411	129	407	278	385	16	366	87
78	243	34	262	130	261	203	252	161	260	80	**168**	599	327	609	493	615	11	590	85	561	162
79	507	213	517	33	498	245	483	191	479	261	**169**	675	72	622	441	560	407	621	443	615	383
80	316	30	316	10	330	137	304	80	304	255	**170**	649	394	614	232	650	290	659	197	606	12
81	161	81	163	10	163	58	173	32	169	220	**171**	192	316	204	20	199	541	199	397	198	81
82	464	36	459	87	455	186	462	64	427	135	**172**	453	265	440	338	471	220	462	83	472	260
83	428	233	365	175	443	223	413	212	433	268	**173**	492	548	519	207	497	605	498	232	489	404
84	160	240	180	4	170	27	182	216	156	233	**174**	519	469	515	212	538	81	507	335	532	13
85	481	151	488	160	497	250	456	255	476	61	**175**	201	207	205	139	209	297	201	242	221	9
86	425	70	460	66	454	57	444	175	471	194	**176**	827	216	766	17	827	203	828	133	776	25
87	688	206	712	177	778	92	672	45	620	118	**177**	921	19	908	494	972	12	902	658	870	87
88	702	4	657	60	692	138	668	177	625	71	**178**	590	80	632	68	594	594	546	399	630	336
89	401	37	348	116	387	135	382	116	366	109	**179**	415	13	440	195	428	585	435	325	404	11
90	615	9	587	116	576	115	570	251	634	221	**180**	967	460	920	293	881	223	989	341	947	88

Appendix D PSO Heuristic Procedures with SA – Results

PSO/SA Heuristic Procedure 1

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	6	7	6	7	6	7	6	7	6
2	14	7	14	7	14	7	14	7	14	7
3	9	6	9	6	9	6	9	6	9	6
4	7	7	7	7	7	6	7	6	7	6
5	16	6	16	6	16	7	16	7	16	7
6	14	7	14	7	14	7	14	7	14	6
7	9	7	9	7	9	6	9	6	9	7
8	16	7	16	7	16	7	16	7	16	6
9	11	6	11	6	11	6	11	6	11	7
10	17	7	17	7	17	7	17	7	17	6
11	18	8	18	7	18	7	18	7	18	7
12	61	8	61	8	61	8	61	8	61	8
13	49	8	49	8	49	8	49	8	49	8
14	48	8	48	8	48	8	48	7	48	8
15	61	8	61	8	61	8	61	8	61	7
16	49	8	49	8	49	8	49	8	49	8
17	92	8	92	8	92	8	92	8	92	8
18	92	9	92	9	92	8	92	9	92	9
19	44	8	44	8	44	8	44	7	44	7
20	54	8	54	8	54	8	54	8	54	8
21	35	8	35	8	35	8	35	8	35	8
22	95	9	95	8	95	8	95	8	95	8
23	82	8	82	9	82	9	82	9	82	9
24	35	8	35	8	35	8	35	8	35	8
25	95	9	95	9	95	8	95	8	95	8
26	82	9	82	9	82	9	82	9	82	9
27	142	9	142	9	142	9	142	9	142	9
28	142	9	142	9	142	9	142	9	142	8
29	98	9	98	9	98	9	98	8	98	9
30	113	9	113	9	113	8	113	9	113	9
31	20	131	19	124	19	119	19	128	18	119
32	37	134	37	134	36	130	36	134	33	129
33	28	131	28	130	28	136	24	124	27	127
34	21	117	21	122	21	115	17	113	19	111
35	36	131	38	130	33	130	37	121	36	127
36	29	129	29	124	27	124	28	122	27	122
37	44	134	44	137	44	138	44	136	44	139
38	62	131	62	131	62	134	54	133	60	136

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
91	42	202	42	202	42	200	42	218	42	216
92	79	225	79	217	79	216	79	205	79	216
93	75	216	75	215	75	213	75	213	76	215
94	42	208	42	199	42	211	42	206	42	207
95	80	222	80	217	80	215	80	208	80	206
96	76	211	76	217	76	220	76	214	76	213
97	109	230	107	228	107	225	109	216	107	235
98	114	225	114	225	114	227	114	230	114	227
99	101	236	101	219	101	217	101	222	101	230
100	164	247	164	229	164	244	164	232	165	241
101	86	236	86	240	86	233	86	234	86	228
102	260	295	260	273	260	265	260	252	260	263
103	236	286	236	273	236	252	236	252	236	256
104	86	230	86	229	86	269	86	224	86	232
105	260	273	260	260	260	247	260	259	260	245
106	236	261	236	260	236	260	236	252	236	257
107	410	277	410	276	410	269	410	271	410	262
108	212	261	212	260	212	257	212	254	212	263
109	218	261	218	253	218	250	218	251	218	252
110	430	267	430	267	430	256	430	268	430	286
111	142	253	139	269	142	268	139	250	142	258
112	379	282	373	287	379	264	379	273	373	271
113	362	284	362	266	362	273	356	276	362	291
114	145	240	145	240	143	251	143	258	139	233
115	612	297	603	296	612	292	612	296	603	304
116	383	266	383	284	384	279	384	257	383	274
117	366	266	356	294	356	281	366	267	356	282
118	612	282	612	281	612	279	612	267	598	295
119	528	273	537	268	537	298	536	269	528	290
120	762	300	762	271	762	308	762	281	762	289
121	53	297	53	303	53	323	53	303	53	316
122	111	321	109	326	107	335	110	330	108	358
123	107	320	105	329	104	324	107	323	105	344
124	58	291	58	306	60	307	56	283	60	307
125	262	309	262	339	262	363	264	352	276	350
126	163	349	163	332	174	354	163	338	173	339
127	155	348	155	353	167	347	155	349	155	340
128	270	344	266	348	266	350	254	335	270	347

39	54	130	54	134	49	123	49	130	49	123	**129**	176	349	176	325	176	351	174	327	171	348
40	111	147	116	138	111	147	111	139	111	146	**130**	231	349	223	363	231	335	225	327	212	335
41	67	148	79	138	79	135	77	130	79	128	**131**	109	331	110	330	111	324	110	324	111	319
42	278	154	278	154	271	151	278	143	258	140	**132**	317	368	311	380	311	362	311	347	311	369
43	259	162	246	150	239	149	259	142	252	136	**133**	321	360	318	362	327	375	321	365	321	351
44	67	146	75	138	79	132	79	129	79	130	**134**	114	331	116	330	108	340	114	318	112	347
45	334	166	334	164	334	156	394	149	374	146	**135**	523	370	515	393	523	389	523	357	523	355
46	279	160	239	157	279	142	279	141	279	140	**136**	321	371	314	372	319	393	314	347	319	376
47	260	156	260	154	260	144	252	140	260	137	**137**	330	374	324	357	328	370	324	363	329	366
48	334	156	374	156	394	149	394	145	334	154	**138**	534	393	526	375	526	397	534	356	526	369
49	220	157	220	154	220	154	259	146	259	142	**139**	550	382	536	379	550	386	550	374	536	369
50	215	156	191	155	215	143	215	141	215	135	**140**	599	377	568	386	568	380	587	364	577	382
51	108	153	118	152	128	145	128	138	118	144	**141**	197	358	193	359	194	368	193	361	195	365
52	326	162	378	161	359	161	326	162	326	162	**142**	485	402	482	379	485	377	475	407	485	369
53	298	173	338	161	298	160	298	157	354	149	**143**	495	393	492	384	492	379	495	398	492	368
54	108	153	128	146	128	142	120	146	118	144	**144**	200	353	196	343	196	346	198	342	192	337
55	520	180	520	169	604	162	520	161	520	162	**145**	783	391	788	385	779	385	783	368	779	379
56	326	173	366	171	326	161	326	162	355	156	**146**	492	373	488	386	488	376	488	387	478	377
57	298	162	298	163	298	160	354	153	354	152	**147**	495	400	498	369	483	376	495	380	490	380
58	426	174	426	168	493	162	426	166	493	163	**148**	788	391	782	400	792	391	786	385	782	399
59	332	163	332	162	332	158	392	154	332	156	**149**	474	403	482	389	489	367	490	392	485	375
60	380	163	380	162	436	161	380	156	380	155	**150**	1611	416	1629	416	1629	415	1611	392	1587	421
61	36	182	35	188	36	195	38	194	38	180	**151**	66	543	62	506	62	478	62	527	66	503
62	76	198	69	197	73	197	69	202	76	195	**152**	229	567	229	543	246	534	229	589	241	540
63	69	199	69	198	69	195	62	201	63	195	**153**	131	533	131	553	131	538	131	558	141	518
64	39	182	35	185	39	185	39	184	39	187	**154**	127	507	137	535	127	510	131	527	127	524
65	78	200	78	199	79	199	76	193	68	203	**155**	68	528	68	486	68	494	70	473	68	491
66	71	196	71	199	71	196	71	186	69	194	**156**	145	554	137	543	137	511	137	492	137	541
67	110	201	110	192	110	198	110	204	110	195	**157**	141	547	141	536	133	529	133	489	133	538
68	83	199	95	200	93	195	93	203	93	201	**158**	318	546	318	577	318	543	318	556	318	597
69	117	201	101	204	103	199	116	209	112	211	**159**	184	523	184	541	184	522	184	532	184	528
70	143	203	138	215	145	221	145	199	138	201	**160**	188	497	188	541	188	537	188	538	198	556
71	84	210	92	206	89	212	84	208	90	208	**161**	111	530	111	561	118	544	111	512	118	514
72	275	217	269	215	263	214	282	220	282	220	**162**	336	547	336	556	336	555	336	582	356	559
73	258	216	249	215	248	226	247	217	260	215	**163**	315	586	333	552	315	608	315	575	315	585
74	92	209	84	204	90	198	90	207	86	214	**164**	350	586	332	563	350	580	332	556	332	580
75	270	235	272	213	285	224	263	220	286	226	**165**	123	504	123	497	117	519	117	538	117	514
76	273	223	249	222	256	220	263	222	273	227	**166**	321	582	321	575	321	549	339	583	328	557
77	420	215	432	222	452	213	420	230	436	232	**167**	338	584	357	554	343	555	338	574	355	544
78	223	216	222	231	208	207	209	208	223	228	**168**	536	592	536	613	536	586	536	567	559	569
79	392	227	416	230	418	230	418	225	404	231	**169**	544	593	539	585	544	596	544	589	574	574
80	272	222	274	222	250	221	270	213	270	216	**170**	558	554	558	561	558	574	589	625	558	580
81	141	218	144	213	141	212	150	230	150	218	**171**	177	537	173	531	182	532	173	574	179	555
82	382	238	393	227	400	236	400	224	400	222	**172**	406	578	405	613	414	572	413	574	410	591
83	346	238	349	232	366	238	361	236	368	237	**173**	434	563	434	557	434	559	449	582	434	608
84	142	214	144	210	154	217	150	208	142	214	**174**	457	565	457	601	457	571	457	603	457	575
85	397	238	400	233	396	236	387	235	387	230	**175**	181	549	187	567	179	537	179	560	179	560
86	363	238	368	238	358	231	366	234	366	231	**176**	718	571	742	609	718	636	718	596	719	624
87	588	256	620	243	620	237	636	241	596	241	**177**	811	617	784	610	784	579	784	612	784	610
88	567	232	556	245	572	235	544	253	560	234	**178**	540	567	522	595	540	617	538	635	540	629
89	335	222	336	221	346	227	336	250	343	222	**179**	392	577	377	571	377	600	377	600	392	549
90	503	234	506	240	517	242	507	249	519	243	**180**	804	617	827	597	827	589	804	645	841	653

PSO/SA Heuristic Procedure 2

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	6	7	6	7	7	7	6	7	6
2	14	7	14	8	14	7	14	7	14	8
3	9	7	9	7	9	7	9	7	9	7
4	7	7	7	6	7	7	7	7	7	6
5	16	8	16	8	16	7	16	8	16	8
6	14	7	14	7	14	8	14	7	14	7
7	9	7	9	7	9	7	9	7	9	8
8	16	8	16	8	16	7	16	8	16	7
9	11	7	11	7	11	7	11	7	11	7
10	17	8	17	8	17	8	17	8	17	8
11	18	7	18	7	18	8	18	7	18	8
12	61	9	61	9	61	8	61	9	61	9
13	49	8	49	9	49	9	49	9	49	8
14	48	9	48	9	48	9	48	9	48	9
15	61	8	61	9	61	9	61	9	61	9
16	49	9	49	8	49	9	49	9	49	9
17	92	9	92	10	92	9	92	9	92	9
18	92	10	92	9	92	9	92	9	92	10
19	44	8	44	9	44	9	44	9	44	9
20	54	9	54	9	54	9	54	9	54	8
21	35	9	35	8	35	9	35	9	35	9
22	95	9	95	10	95	9	95	9	95	10
23	82	10	82	10	82	10	82	10	82	10
24	35	9	35	9	35	9	35	9	35	9
25	95	9	95	10	95	10	95	10	95	10
26	82	10	82	9	82	10	82	9	82	9
27	142	10	142	11	142	10	142	11	142	11
28	142	10	142	10	142	10	142	10	142	10
29	98	10	98	10	98	10	98	10	98	10
30	113	10	113	10	113	10	113	10	113	10
31	25	123	25	117	25	119	25	117	25	119
32	43	132	47	126	47	126	47	126	47	128
33	32	129	32	124	32	126	32	125	32	127
34	25	124	25	121	25	121	25	120	25	122
35	46	132	38	126	38	127	38	127	38	129
36	32	131	38	125	38	127	38	126	38	128
37	48	135	56	130	56	131	56	129	56	131
38	78	137	78	132	78	132	78	132	78	134
39	59	135	64	131	64	133	64	131	64	134
40	135	145	136	139	136	140	136	139	136	141
41	79	144	79	137	79	140	79	140	79	141

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
91	42	222	42	157	42	160	42	159	42	163
92	79	233	79	169	79	172	79	171	79	173
93	75	230	77	168	77	171	77	171	77	174
94	42	222	44	156	44	159	44	158	44	161
95	82	235	82	168	82	172	82	171	82	173
96	78	233	76	167	76	170	76	170	76	173
97	107	244	109	176	109	179	109	178	109	181
98	114	238	114	174	114	177	114	176	114	180
99	104	237	103	172	103	175	103	174	103	177
100	164	248	171	183	171	187	171	187	171	190
101	86	244	86	180	86	183	86	182	86	185
102	260	269	260	205	260	207	260	206	260	209
103	236	269	236	202	236	206	236	205	236	209
104	86	247	86	178	86	181	86	180	86	183
105	260	269	260	208	260	206	260	205	260	209
106	236	267	236	208	236	206	236	204	236	207
107	410	282	410	221	410	217	410	216	410	219
108	212	265	212	205	212	202	212	201	212	202
109	218	267	218	204	218	202	218	201	218	202
110	430	279	430	219	430	217	430	215	430	217
111	139	266	142	203	142	200	142	199	142	201
112	379	289	380	225	380	223	380	222	380	223
113	362	283	363	225	363	222	363	221	363	222
114	145	263	145	202	145	200	145	198	145	200
115	612	297	612	235	612	232	612	231	612	233
116	385	286	385	225	385	222	385	221	385	222
117	368	286	367	223	367	221	367	221	367	221
118	612	293	620	234	620	232	620	231	620	232
119	541	294	537	235	537	231	537	229	537	231
120	762	298	762	240	762	236	762	235	762	237
121	53	299	55	276	55	273	55	271	55	273
122	111	322	109	298	109	296	109	295	109	296
123	102	324	102	296	102	296	102	294	102	296
124	58	300	60	275	60	275	60	274	60	275
125	270	354	260	328	260	327	260	326	260	329
126	173	337	182	311	182	312	182	311	182	312
127	161	335	159	312	159	312	159	310	159	312
128	270	347	266	323	266	323	266	322	266	324
129	182	335	187	311	187	311	187	310	187	312
130	229	344	231	319	231	318	231	317	231	319
131	111	322	111	298	111	301	111	297	111	298

42	278	160	275	154	275	158	275	156	275	159	**132**	316	357	316	333	316	334	316	332	316	334
43	259	160	259	154	259	157	259	156	259	158	**133**	326	356	332	333	332	333	332	332	332	334
44	79	145	79	138	79	141	79	140	79	142	**134**	114	326	112	300	112	300	112	298	112	300
45	394	165	394	158	394	161	394	160	394	162	**135**	539	374	523	348	523	349	523	348	523	350
46	279	161	279	154	279	158	279	156	279	159	**136**	325	357	319	333	319	333	319	333	319	334
47	260	160	260	154	260	157	260	157	260	158	**137**	329	360	324	334	324	335	324	333	324	334
48	394	165	394	159	394	161	394	160	394	163	**138**	534	374	534	351	534	350	534	348	534	351
49	259	160	259	154	259	158	259	157	259	159	**139**	539	376	574	350	574	352	574	349	574	350
50	215	161	215	152	215	154	215	153	215	156	**140**	602	376	577	352	577	357	577	350	577	353
51	125	154	128	149	128	153	128	151	128	153	**141**	196	350	194	323	194	328	194	356	194	324
52	382	171	382	165	382	168	382	167	382	169	**142**	484	377	489	354	489	358	489	352	489	355
53	354	170	337	165	337	167	337	167	337	169	**143**	495	377	491	353	491	358	491	353	491	353
54	128	153	128	149	128	153	128	151	128	154	**144**	200	348	196	325	196	329	196	323	196	326
55	604	175	604	172	604	175	604	174	604	176	**145**	760	394	783	369	783	374	783	368	783	370
56	382	170	382	166	382	168	382	168	382	170	**146**	487	379	477	353	477	355	477	353	477	354
57	354	168	354	164	354	168	354	167	354	169	**147**	502	379	495	355	495	358	495	353	495	356
58	506	172	505	169	505	172	505	170	505	174	**148**	790	394	786	369	786	372	786	368	786	369
59	392	168	392	164	392	166	392	166	392	168	**149**	490	380	493	354	493	355	493	352	493	355
60	450	169	450	166	450	170	450	169	450	171	**150**	1629	416	1629	394	1629	396	1629	392	1629	394
61	38	184	39	152	39	155	39	154	39	155	**151**	62	454	66	377	66	378	66	374	66	379
62	76	198	78	164	78	167	78	167	78	169	**152**	246	517	229	432	229	434	229	430	229	433
63	69	196	62	163	62	166	62	165	62	167	**153**	131	492	131	409	131	411	131	407	131	410
64	39	189	41	157	41	159	41	159	41	161	**154**	127	489	127	408	127	411	127	408	127	409
65	76	199	76	168	76	171	76	170	76	171	**155**	68	462	70	380	70	378	70	377	70	381
66	58	198	71	166	71	169	71	168	71	169	**156**	137	490	137	410	137	406	137	408	137	411
67	102	207	102	172	102	176	102	175	102	176	**157**	140	488	141	410	141	407	141	407	141	410
68	93	202	93	169	93	172	93	170	93	172	**158**	338	523	318	445	318	441	318	441	318	443
69	116	205	116	173	116	175	116	176	116	176	**159**	184	494	184	420	184	416	184	416	184	420
70	143	210	143	177	143	180	143	179	143	181	**160**	188	499	188	419	188	417	188	425	188	420
71	89	209	90	175	90	178	90	178	90	178	**161**	111	494	118	415	118	411	118	422	118	415
72	281	230	273	198	273	201	273	200	273	201	**162**	336	535	336	457	336	455	336	465	336	458
73	242	231	260	197	260	201	260	200	260	202	**163**	315	535	337	458	337	454	337	465	337	458
74	90	208	84	176	84	179	84	179	84	179	**164**	332	546	341	459	341	454	341	467	341	458
75	290	231	274	200	274	202	274	201	274	203	**165**	123	492	117	416	117	413	117	424	117	417
76	241	230	268	197	268	201	268	201	268	202	**166**	321	536	321	459	321	454	321	465	321	458
77	432	241	420	209	420	211	420	210	420	212	**167**	357	530	350	459	350	454	350	466	350	459
78	223	224	214	194	214	197	214	197	214	197	**168**	564	561	536	479	536	480	536	484	536	480
79	392	240	404	209	404	212	404	212	404	213	**169**	544	554	564	479	564	479	564	481	564	479
80	266	229	271	197	271	200	271	199	271	200	**170**	558	555	558	479	558	480	558	481	558	479
81	144	223	149	190	149	194	149	193	149	194	**171**	173	529	173	449	173	450	173	451	173	448
82	396	242	380	210	380	213	380	213	380	214	**172**	406	569	404	483	404	485	404	486	404	484
83	381	242	356	210	356	214	356	213	356	214	**173**	434	567	448	488	448	490	448	491	448	489
84	146	224	146	191	146	194	146	194	146	194	**174**	457	571	457	490	457	491	457	492	457	490
85	397	244	400	211	400	215	400	213	400	215	**175**	185	529	185	450	185	451	185	452	185	450
86	366	242	368	210	368	213	368	213	368	215	**176**	718	586	730	509	730	511	730	511	730	510
87	620	251	596	221	596	224	596	224	596	226	**177**	784	596	811	512	811	514	811	517	811	512
88	558	249	539	217	539	221	539	220	539	223	**178**	522	588	544	495	544	497	544	497	544	495
89	348	239	343	206	343	210	343	209	343	213	**179**	392	563	377	482	377	485	377	484	377	483
90	521	248	521	217	521	219	521	219	521	222	**180**	804	597	824	513	824	515	824	516	824	513

PSO/SA Heuristic Procedure 3

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$			$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	8	7	7	7	7	7	7	7	7	**91**	42	201	42	195	42	186	42	185	42	192
2	14	8	14	8	14	7	14	7	14	7	**92**	79	221	79	204	79	203	79	200	79	197
3	9	8	9	7	9	7	9	7	9	7	**93**	75	220	75	199	75	208	76	191	75	201
4	7	7	7	7	7	7	7	7	7	7	**94**	42	200	42	184	42	184	42	188	42	185
5	16	8	16	8	16	8	16	8	16	8	**95**	80	208	80	199	80	197	80	198	80	196
6	14	8	14	8	14	8	14	8	14	7	**96**	76	208	76	200	76	198	76	200	76	193
7	9	8	9	7	9	7	9	7	9	7	**97**	107	222	107	209	107	209	107	206	107	212
8	16	8	16	8	16	7	16	8	16	8	**98**	114	221	114	206	114	201	114	201	114	209
9	11	8	11	8	11	8	11	7	11	8	**99**	101	217	101	213	101	205	101	206	101	203
10	17	8	17	8	17	7	17	8	17	7	**100**	169	231	169	213	165	214	169	218	169	212
11	18	9	18	8	18	8	18	8	18	8	**101**	86	229	86	214	86	219	86	214	86	212
12	61	9	61	9	61	10	61	9	61	10	**102**	260	259	260	249	260	239	260	236	260	239
13	49	10	49	10	49	8	49	9	49	9	**103**	236	253	236	244	236	243	236	237	236	234
14	48	9	48	9	48	10	48	9	48	9	**104**	86	237	86	223	86	210	86	217	86	206
15	61	10	61	9	61	9	61	10	61	9	**105**	260	259	260	247	260	243	260	231	260	232
16	49	10	49	9	49	9	49	9	49	9	**106**	236	251	236	245	236	241	236	232	236	233
17	92	10	92	10	92	10	92	9	92	10	**107**	410	265	410	270	410	241	410	247	410	248
18	92	10	92	12	92	9	92	10	92	9	**108**	212	255	212	226	212	230	212	230	212	223
19	44	10	44	9	44	9	44	9	44	9	**109**	218	246	218	237	218	249	218	240	218	226
20	54	10	54	9	54	9	54	9	54	9	**110**	430	265	430	255	430	246	430	252	430	248
21	35	9	35	9	35	9	35	9	35	9	**111**	139	258	142	240	139	234	139	240	142	238
22	95	11	95	10	95	10	95	10	95	10	**112**	373	278	379	262	373	263	373	252	380	273
23	82	10	82	10	82	10	82	10	82	10	**113**	363	276	356	277	356	267	362	253	356	259
24	35	10	35	9	35	9	35	9	35	9	**114**	139	247	145	237	145	228	139	240	145	239
25	95	11	95	11	95	10	95	10	95	11	**115**	603	292	613	277	603	269	613	275	612	263
26	82	10	82	10	82	10	82	10	82	9	**116**	385	274	373	280	385	253	373	273	373	258
27	142	11	142	11	142	11	142	11	142	11	**117**	366	268	356	270	356	259	356	261	356	256
28	142	12	142	10	142	10	142	10	142	10	**118**	598	289	620	268	598	269	620	273	618	263
29	98	11	98	11	98	10	98	10	98	11	**119**	528	284	541	262	537	267	541	268	541	268
30	113	12	113	10	113	11	113	11	113	10	**120**	751	288	762	266	751	276	762	268	762	272
31	17	182	17	172	17	155	17	160	17	152	**121**	52	341	53	318	54	313	51	304	53	308
32	33	187	33	173	33	159	33	165	33	150	**122**	101	355	106	351	105	345	106	358	100	334
33	24	180	24	167	24	158	24	165	24	156	**123**	97	372	100	342	102	352	98	339	97	331
34	17	176	17	156	17	156	17	152	17	151	**124**	56	340	54	303	54	308	52	303	54	302
35	33	190	33	161	33	171	33	169	33	158	**125**	246	387	244	366	243	381	245	374	243	375
36	24	171	24	160	24	168	24	174	24	154	**126**	154	382	154	365	154	368	154	368	154	343
37	40	186	40	163	40	164	40	175	45	152	**127**	146	373	146	363	146	371	146	366	146	344
38	54	182	54	169	54	165	54	160	54	163	**128**	256	379	244	377	248	389	254	361	264	356
39	44	191	44	175	44	165	44	167	44	168	**129**	168	384	168	375	168	369	168	356	168	357
40	99	201	99	184	99	173	99	182	99	177	**130**	225	389	225	367	225	368	217	352	225	353
41	77	182	75	177	67	172	77	166	79	163	**131**	109	368	109	347	104	354	106	347	107	344

42	239	203	239	186	278	183	239	185	278	177	**132**	306	406	299	380	306	395	306	385	306	365
43	220	197	220	187	220	188	220	181	220	179	**133**	316	425	316	384	316	373	316	393	316	375
44	67	188	79	177	79	168	77	162	77	162	**134**	112	381	110	353	112	353	112	323	112	327
45	374	206	384	191	394	187	394	187	376	183	**135**	479	413	523	391	512	409	512	399	508	382
46	239	198	279	187	279	187	279	181	239	184	**136**	314	393	306	378	314	376	314	382	314	370
47	260	198	260	184	220	186	260	185	260	176	**137**	324	424	324	368	324	377	316	384	324	376
48	334	206	334	204	334	193	374	184	334	188	**138**	522	409	526	405	526	387	524	370	524	384
49	259	194	258	187	220	183	220	196	259	178	**139**	540	432	513	432	541	405	526	406	550	374
50	215	198	215	184	191	181	215	182	215	175	**140**	577	424	577	383	577	392	561	376	577	377
51	108	201	118	181	108	187	108	177	120	180	**141**	187	419	187	380	193	369	193	378	187	378
52	326	212	326	198	326	201	326	197	326	196	**142**	470	449	470	413	470	420	462	420	478	413
53	331	213	298	204	320	198	298	204	298	199	**143**	481	416	471	406	492	397	484	418	480	396
54	108	208	118	182	118	183	128	181	108	174	**144**	190	407	190	382	200	368	190	369	190	363
55	520	225	520	205	596	205	520	204	596	197	**145**	760	463	740	435	751	424	763	411	739	404
56	326	213	326	196	365	200	326	194	362	207	**146**	485	411	473	418	469	404	473	404	473	408
57	298	217	337	191	336	200	298	201	320	197	**147**	486	430	478	416	482	404	473	423	483	419
58	426	215	426	210	426	199	448	211	426	216	**148**	746	448	782	435	774	421	766	423	766	421
59	368	214	344	200	332	202	332	204	344	196	**149**	486	421	482	405	485	399	467	408	472	386
60	380	220	422	197	380	203	380	201	380	208	**150**	1634	469	1611	438	1576	450	1634	422	1634	425
61	35	204	35	204	38	186	35	184	37	186	**151**	62	487	62	448	61	446	62	431	62	420
62	68	230	68	200	68	201	76	202	68	207	**152**	226	560	226	504	226	495	226	522	229	500
63	61	219	61	209	61	202	61	200	61	195	**153**	131	501	131	477	132	478	130	455	131	446
64	37	208	39	181	33	181	33	181	33	177	**154**	137	505	128	509	127	506	127	489	127	465
65	68	230	76	206	68	201	65	198	68	199	**155**	68	498	68	433	64	474	64	444	68	438
66	61	232	66	205	61	202	71	195	61	192	**156**	137	541	145	467	133	477	137	488	137	483
67	106	233	94	204	94	203	90	202	106	199	**157**	133	520	133	507	133	493	129	468	129	470
68	91	239	83	232	91	208	83	203	91	203	**158**	313	563	313	531	318	539	338	492	318	480
69	101	234	101	219	101	217	101	207	115	199	**159**	184	517	194	504	190	504	190	487	184	488
70	129	242	129	223	129	216	129	202	129	203	**160**	188	528	188	498	188	490	188	464	190	487
71	83	233	83	219	83	208	83	216	90	205	**161**	111	569	111	503	118	484	111	478	108	503
72	269	243	269	245	263	231	264	230	269	228	**162**	346	560	336	524	350	515	336	517	356	517
73	241	255	247	226	241	235	247	232	253	227	**163**	315	589	315	529	333	540	315	583	315	528
74	84	227	84	211	84	217	86	208	84	210	**164**	332	588	335	548	332	524	332	560	332	527
75	263	259	263	234	272	232	263	228	263	232	**165**	117	537	117	511	123	479	113	468	117	484
76	241	251	250	232	241	230	241	228	241	220	**166**	318	617	333	544	321	567	321	539	338	517
77	452	269	420	246	420	252	420	242	420	225	**167**	342	596	338	567	354	534	338	542	331	519
78	208	252	208	240	208	228	209	212	212	218	**168**	564	626	530	573	539	582	536	587	536	553
79	392	283	392	254	392	242	404	251	424	239	**169**	544	593	544	542	568	560	544	557	544	525
80	250	254	250	235	250	232	250	229	250	230	**170**	558	617	589	521	558	556	558	559	558	527
81	141	254	142	231	142	235	142	224	142	220	**171**	173	560	173	574	173	535	173	555	173	521
82	399	285	400	257	382	251	390	251	380	245	**172**	355	638	399	587	371	563	394	543	405	537
83	348	267	348	270	348	247	348	246	349	238	**173**	427	592	434	550	434	548	439	553	434	553
84	142	271	142	230	142	216	142	218	142	225	**174**	450	603	448	623	457	603	462	557	457	562
85	382	278	382	260	390	240	382	241	382	276	**175**	157	580	179	545	179	538	179	506	159	496
86	361	267	366	251	356	245	348	246	348	241	**176**	718	609	718	608	742	592	718	594	718	556
87	588	284	596	254	588	252	588	265	588	250	**177**	784	653	800	582	800	597	784	591	784	569
88	560	283	547	257	550	257	541	269	554	254	**178**	522	630	522	592	522	553	522	583	522	535
89	328	264	333	252	333	250	345	244	333	242	**179**	377	603	377	575	351	570	377	568	377	532
90	517	278	524	254	514	241	514	259	528	255	**180**	816	626	804	606	804	574	804	589	804	570

PSO/SA Heuristic Procedure 4

Problem #	$c_1 = c_2 = 0.25$ Min. Cost	$c_1 = c_2 = 0.25$ Total Time in Seconds	$c_1 = c_2 = 0.30$ Min. Cost	$c_1 = c_2 = 0.30$ Total Time in Seconds	$c_1 = c_2 = 0.35$ Min. Cost	$c_1 = c_2 = 0.35$ Total Time in Seconds	$c_1 = c_2 = 0.40$ Min. Cost	$c_1 = c_2 = 0.40$ Total Time in Seconds	$c_1 = c_2 = 0.45$ Min. Cost	$c_1 = c_2 = 0.45$ Total Time in Seconds
1	7	12	7	12	7	12	7	13	7	11
2	14	15	14	13	14	13	14	12	14	12
3	9	16	9	13	9	14	9	11	9	12
4	7	13	7	12	7	13	7	12	7	11
5	16	13	16	13	16	13	16	16	16	12
6	14	15	14	14	14	15	14	21	14	13
7	9	13	9	13	9	13	9	21	9	12
8	16	13	16	14	16	13	16	23	16	12
9	11	11	11	13	11	13	11	21	11	12
10	17	13	17	13	17	14	17	21	17	13
11	18	13	18	14	18	14	18	20	18	12
12	61	14	61	18	61	15	61	28	61	15
13	49	16	49	15	49	16	49	26	49	14
14	48	16	48	16	48	16	48	32	48	16
15	61	15	61	17	61	16	61	31	61	20
16	49	15	49	15	49	15	49	27	49	18
17	92	17	92	17	92	17	92	29	92	25
18	92	17	92	17	92	17	92	32	92	19
19	44	16	44	16	44	15	44	29	44	20
20	54	15	54	15	54	16	54	23	54	17
21	35	15	35	16	35	16	35	21	35	15
22	95	16	95	17	95	17	95	28	95	16
23	82	18	82	17	82	19	82	25	82	16
24	35	17	35	16	35	15	35	26	35	15
25	95	17	95	18	95	17	95	26	95	17
26	82	22	82	17	82	18	82	27	82	16
27	142	31	142	17	142	18	142	29	142	18
28	142	23	142	19	142	17	142	28	142	17
29	98	19	98	17	98	18	98	32	98	16
30	113	25	113	20	113	17	113	35	113	17
31	20	297	21	287	22	305	19	464	17	286
32	36	341	37	319	39	337	36	413	36	322
33	28	310	27	306	31	289	27	337	27	344
34	21	269	21	285	23	258	19	268	21	279
35	38	294	36	332	46	257	38	292	39	290
36	35	249	29	330	32	271	29	269	32	298
37	45	326	44	353	52	273	44	286	44	321
38	62	293	58	298	66	309	60	297	60	294
39	49	349	49	323	61	283	49	281	49	293
40	116	356	111	343	124	321	111	334	111	317
41	75	314	77	320	79	284	79	289	78	315

Problem #	$c_1 = c_2 = 0.25$ Min. Cost	$c_1 = c_2 = 0.25$ Total Time in Seconds	$c_1 = c_2 = 0.30$ Min. Cost	$c_1 = c_2 = 0.30$ Total Time in Seconds	$c_1 = c_2 = 0.35$ Min. Cost	$c_1 = c_2 = 0.35$ Total Time in Seconds	$c_1 = c_2 = 0.40$ Min. Cost	$c_1 = c_2 = 0.40$ Total Time in Seconds	$c_1 = c_2 = 0.45$ Min. Cost	$c_1 = c_2 = 0.45$ Total Time in Seconds
91	42	376	42	372	42	348	42	379	42	374
92	80	390	80	401	79	373	80	393	80	381
93	75	403	75	398	75	386	75	382	75	388
94	42	384	42	369	42	364	42	365	42	361
95	80	394	80	383	80	388	80	381	80	382
96	76	390	76	384	76	375	76	393	76	382
97	107	429	111	410	108	397	107	413	109	386
98	114	407	114	383	114	388	114	404	114	395
99	101	401	101	393	101	444	101	388	101	389
100	165	423	164	403	165	425	164	418	164	400
101	86	437	86	415	86	420	86	422	86	422
102	260	460	260	447	260	482	260	469	260	476
103	236	465	236	479	236	485	236	456	236	476
104	86	421	86	421	86	407	86	413	86	429
105	260	457	260	475	260	454	260	476	260	473
106	236	498	236	475	236	466	236	447	236	446
107	410	482	410	495	410	495	410	465	410	482
108	212	462	212	450	212	450	212	484	212	464
109	218	458	218	448	218	444	218	452	218	456
110	430	464	430	498	430	480	430	476	430	450
111	142	438	139	463	142	444	139	448	139	448
112	373	493	373	501	379	480	373	516	379	482
113	356	501	362	474	356	450	356	523	362	485
114	139	439	139	440	143	452	139	444	139	454
115	603	523	603	524	603	493	603	420	603	493
116	373	487	373	501	373	478	373	486	373	489
117	356	520	356	481	367	475	368	483	356	511
118	598	500	598	500	612	502	598	505	598	511
119	528	525	528	515	528	500	536	521	537	484
120	762	529	751	518	751	503	751	506	751	499
121	53	602	50	627	54	616	54	593	53	611
122	111	661	111	717	108	625	111	627	108	653
123	108	669	101	609	111	617	107	659	107	638
124	56	606	56	645	58	553	56	599	58	595
125	255	698	271	725	274	701	271	687	269	707
126	160	657	165	717	154	656	170	667	173	648
127	158	681	170	678	165	636	157	682	159	681
128	256	717	264	682	274	677	240	666	276	687
129	171	600	177	675	174	669	182	644	173	668
130	225	694	228	680	231	674	231	651	231	689
131	109	650	113	655	109	614	111	649	109	647

42	275	319	278	344	265	309	278	314	275	338	**132**	317	730	316	676	311	650	316	699	308	707
43	259	344	259	335	259	298	246	340	259	390	**133**	326	708	321	711	321	690	321	712	327	724
44	79	346	77	322	77	292	79	324	79	324	**134**	114	672	116	632	112	622	116	666	116	621
45	384	393	384	371	394	320	384	421	394	357	**135**	531	719	523	744	523	713	531	713	531	749
46	270	354	271	350	275	317	279	327	279	330	**136**	314	744	314	714	314	714	325	703	314	747
47	260	396	260	348	251	312	260	329	260	309	**137**	324	750	324	722	322	720	324	721	329	713
48	394	472	384	362	374	328	389	359	384	321	**138**	522	759	534	750	526	742	540	742	534	767
49	247	379	258	360	254	320	245	371	258	316	**139**	550	757	550	744	531	730	527	737	550	731
50	215	344	215	337	215	314	213	330	215	301	**140**	577	791	561	760	577	737	577	766	577	763
51	120	345	118	353	120	345	118	359	118	341	**141**	197	708	197	696	195	707	197	689	190	709
52	360	350	366	381	372	457	362	369	364	386	**142**	489	806	470	775	485	756	483	776	482	767
53	339	379	331	378	331	369	354	370	339	359	**143**	495	812	495	767	492	724	492	726	495	740
54	118	346	118	359	118	343	128	344	118	332	**144**	202	691	200	692	196	666	196	707	194	687
55	578	397	604	387	596	374	604	383	556	386	**145**	779	810	787	816	780	756	783	767	783	795
56	364	399	364	473	382	357	367	371	367	366	**146**	493	754	488	730	481	771	492	755	477	744
57	332	367	336	386	354	368	332	373	339	370	**147**	504	757	487	746	502	721	498	768	491	723
58	466	380	466	394	468	371	437	360	466	387	**148**	782	793	792	781	766	754	794	771	786	759
59	362	369	362	403	368	370	350	337	362	383	**149**	492	745	482	767	493	718	490	713	485	754
60	422	389	422	378	422	380	415	361	415	382	**150**	1611	799	1629	803	1634	794	1629	814	1611	833
61	38	333	38	344	38	338	38	342	39	347	**151**	66	973	62	939	62	876	62	947	62	913
62	68	395	76	349	76	364	76	368	75	376	**152**	229	1024	229	1070	229	965	229	1024	229	1019
63	65	333	69	370	71	379	69	348	65	361	**153**	141	968	131	970	131	954	131	973	131	1008
64	41	322	41	360	39	322	39	352	39	338	**154**	127	1004	127	1014	137	909	127	987	137	983
65	79	367	78	373	75	326	78	407	76	366	**155**	70	939	70	908	68	867	68	862	70	931
66	71	394	71	360	72	329	66	384	71	362	**156**	145	956	145	978	137	955	137	981	137	904
67	110	451	106	397	110	366	106	390	110	378	**157**	133	1003	141	1000	133	928	133	996	140	896
68	91	353	89	395	93	376	95	390	87	382	**158**	327	1053	330	1093	330	1016	318	1042	318	1042
69	116	358	107	391	116	377	116	373	116	378	**159**	184	986	184	1052	184	941	184	959	184	896
70	143	377	131	400	143	395	129	380	129	396	**160**	198	1052	188	1013	198	884	198	967	188	970
71	86	358	90	402	83	385	89	401	83	378	**161**	111	1011	118	953	111	858	111	1054	111	905
72	269	386	282	434	264	417	281	428	264	435	**162**	336	1115	336	1008	336	1016	356	1095	336	1289
73	247	452	249	439	247	416	251	438	260	448	**163**	322	1095	333	1028	315	954	315	1154	328	1025
74	94	398	86	390	84	379	90	386	90	373	**164**	332	1070	332	1256	332	1087	332	1211	332	951
75	270	444	274	417	263	424	292	433	272	428	**165**	123	920	117	1057	123	849	117	909	123	1125
76	267	426	241	426	250	415	260	437	260	435	**166**	321	1036	321	958	340	979	321	1017	339	1078
77	428	454	436	458	440	443	420	456	452	428	**167**	338	1074	338	1001	338	1007	338	1249	356	1122
78	212	423	208	425	208	427	229	385	208	421	**168**	536	1101	536	1167	564	1072	536	1113	536	1005
79	404	455	405	487	392	443	404	397	424	451	**169**	544	1071	572	1107	552	1003	544	1085	544	1148
80	276	448	250	416	250	421	263	402	250	427	**170**	558	1092	589	1102	558	1396	605	1034	558	1225
81	147	401	146	411	145	403	151	419	149	413	**171**	173	1121	173	1057	181	1092	179	1029	180	1057
82	397	447	393	448	400	457	393	446	400	448	**172**	413	1118	413	1258	415	992	406	1183	406	1102
83	346	459	366	439	349	427	368	454	349	452	**173**	434	1200	448	1220	454	997	434	1079	448	1254
84	146	416	150	417	142	411	150	425	150	416	**174**	471	1089	457	1062	452	1397	457	1149	457	1087
85	392	450	406	446	400	452	397	446	392	447	**175**	179	1000	179	946	185	1393	179	950	187	974
86	388	444	366	446	366	462	366	437	361	448	**176**	718	1131	718	1055	718	1080	718	1055	718	1499
87	620	453	620	479	640	460	652	443	640	459	**177**	784	1205	822	1047	784	1024	816	1135	784	1547
88	550	466	554	458	554	461	560	446	560	455	**178**	548	1109	538	1054	522	1027	522	1159	522	1507
89	345	432	347	422	342	445	335	448	333	448	**179**	377	1072	377	1060	397	1286	377	1107	395	991
90	507	448	518	469	528	455	519	464	528	462	**180**	804	1177	804	1114	804	1277	804	1073	804	1234

PSO/SA Heuristic Procedure 5

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
1	7	5	7	5	7	4	7	6	7	5
2	14	5	14	6	14	6	14	5	14	6
3	9	5	9	5	9	5	9	5	9	5
4	7	5	7	5	7	5	7	5	7	5
5	16	6	16	6	16	6	16	6	16	6
6	14	5	14	5	14	5	14	6	14	5
7	9	5	9	5	9	5	9	5	9	5
8	16	6	16	6	16	6	16	6	16	6
9	11	5	11	5	11	5	11	5	11	5
10	17	6	17	6	17	6	17	6	17	6
11	18	6	18	6	18	6	18	6	18	6
12	61	6	61	7	61	7	61	6	61	7
13	49	7	49	6	49	6	49	7	49	6
14	48	7	48	7	48	7	48	7	48	7
15	61	7	61	7	61	7	61	7	61	7
16	49	6	49	6	49	7	49	7	49	7
17	92	7	92	8	92	7	92	7	92	7
18	92	8	92	7	92	7	92	7	92	7
19	44	6	44	6	44	7	44	7	44	6
20	54	7	54	7	54	6	54	7	54	7
21	35	7	35	7	35	7	35	7	35	7
22	95	7	95	7	95	8	95	7	95	7
23	82	8	82	8	82	7	82	8	82	8
24	35	6	35	6	35	7	35	7	35	7
25	95	8	95	8	95	7	95	7	95	7
26	82	7	82	7	82	8	82	8	82	7
27	142	8	142	8	142	7	142	8	142	8
28	142	8	142	8	142	8	142	8	142	8
29	98	7	98	8	98	8	98	7	98	8
30	113	8	113	7	113	8	113	8	113	7
31	22	107	19	108	22	107	19	111	19	108
32	39	112	37	114	40	114	36	116	41	114
33	27	111	27	113	31	112	31	111	35	111
34	17	109	23	107	19	112	23	109	23	109
35	36	113	43	112	44	114	41	118	41	115
36	29	110	38	112	29	112	29	116	38	112
37	44	114	49	116	49	116	49	119	48	117
38	60	119	54	119	60	118	68	120	78	119
39	59	119	44	118	58	122	65	122	59	118
40	127	125	116	126	124	125	128	126	128	129
41	79	123	79	126	79	130	77	126	79	129

	$c_1 = c_2 = 0.25$		$c_1 = c_2 = 0.30$		$c_1 = c_2 = 0.35$		$c_1 = c_2 = 0.40$		$c_1 = c_2 = 0.45$	
Problem #	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds	Min. Cost	Total Time in Seconds
91	43	145	42	144	42	146	42	141	43	145
92	79	153	79	153	80	153	79	154	79	152
93	75	153	76	150	77	153	75	151	75	153
94	44	143	42	144	44	143	44	142	44	144
95	80	154	80	155	80	152	82	153	80	154
96	76	153	76	152	78	153	78	153	78	154
97	107	161	107	157	107	155	109	161	112	160
98	114	157	114	158	114	159	116	159	116	158
99	101	155	101	154	101	159	101	155	101	159
100	165	165	169	164	164	164	169	165	164	167
101	86	164	86	164	86	165	86	161	86	166
102	260	185	260	184	260	182	260	184	260	184
103	236	182	236	182	236	181	236	181	236	184
104	86	164	86	162	86	163	86	162	86	164
105	260	182	260	183	260	182	260	183	260	182
106	236	181	236	183	236	181	236	180	236	182
107	410	193	410	190	410	190	410	192	410	192
108	212	178	212	178	212	181	212	179	212	180
109	218	179	218	179	218	180	218	179	218	180
110	430	191	430	189	430	190	430	190	430	190
111	142	178	142	177	142	178	142	179	142	178
112	380	195	379	195	379	195	380	196	380	195
113	363	195	362	196	363	193	362	195	362	197
114	143	178	139	179	145	176	145	178	145	176
115	603	208	613	203	612	204	603	204	612	203
116	385	194	385	196	384	197	383	196	385	195
117	366	195	368	195	368	195	368	192	368	194
118	618	203	618	201	612	201	612	202	598	202
119	536	216	536	203	528	201	537	203	541	201
120	762	214	762	205	762	207	762	207	762	206
121	54	232	54	221	51	221	54	226	55	221
122	111	257	106	243	108	235	100	242	111	240
123	107	243	97	239	112	237	108	244	107	242
124	58	228	58	230	56	225	58	231	58	223
125	275	266	261	264	263	263	256	261	263	262
126	169	257	166	252	160	247	171	252	169	254
127	166	255	168	246	161	249	165	256	166	254
128	264	264	264	267	274	266	264	263	266	259
129	165	253	172	249	185	252	173	253	181	252
130	221	268	230	260	231	264	217	257	225	256
131	107	248	112	249	109	247	109	252	109	245

42	278	147	271	144	278	144	278	146	265	149	**132**	299	286	316	271	311	268	317	274	321	269
43	259	145	259	150	259	143	253	149	259	148	**133**	321	278	316	276	326	275	331	275	321	277
44	77	132	79	132	79	132	77	131	77	135	**134**	112	247	114	251	116	260	112	246	112	248
45	374	146	385	148	385	149	384	148	334	151	**135**	532	292	499	286	532	279	531	282	531	291
46	270	145	279	146	279	153	279	148	279	146	**136**	325	284	319	278	314	269	314	271	322	272
47	260	142	260	147	255	141	260	145	260	149	**137**	331	278	329	277	324	273	331	272	334	276
48	394	142	394	148	394	141	374	149	394	146	**138**	526	288	534	284	534	282	524	286	526	279
49	220	137	258	148	259	144	251	142	258	144	**139**	550	300	550	288	550	282	548	290	550	296
50	191	135	215	139	215	140	215	144	215	142	**140**	577	302	588	292	577	282	577	282	602	289
51	127	142	127	136	124	144	127	141	128	137	**141**	197	275	196	277	197	267	197	265	196	268
52	362	153	382	152	372	152	377	154	359	149	**142**	482	304	485	292	484	294	487	291	486	290
53	354	152	341	155	354	150	331	154	344	153	**143**	491	307	491	293	492	297	492	305	492	291
54	128	139	120	136	112	138	120	145	128	140	**144**	204	271	196	286	200	263	196	270	200	280
55	604	159	520	161	596	165	596	171	575	156	**145**	779	313	783	300	787	300	787	306	783	304
56	365	154	372	152	372	157	382	159	372	153	**146**	485	295	485	295	483	298	490	289	477	291
57	339	154	354	151	354	156	354	157	344	157	**147**	502	299	493	305	500	291	495	288	498	291
58	503	151	466	152	463	163	503	167	501	161	**148**	790	308	774	308	798	302	790	300	794	303
59	387	150	387	147	368	162	387	165	386	159	**149**	493	301	492	299	497	302	482	297	490	288
60	380	150	445	152	445	160	445	166	450	159	**150**	1623	326	1629	325	1611	319	1629	322	1608	319
61	39	129	38	130	38	130	35	135	36	129	**151**	64	329	62	318	66	332	64	325	62	328
62	69	140	69	140	69	141	68	143	76	141	**152**	229	389	226	365	229	386	229	400	229	386
63	61	136	61	139	62	140	69	141	69	139	**153**	131	352	131	340	131	357	131	362	131	361
64	39	132	39	134	39	131	35	134	35	134	**154**	137	356	127	350	133	342	133	346	127	344
65	76	141	78	143	81	141	78	147	78	141	**155**	68	320	68	321	70	344	70	337	68	315
66	71	138	69	138	71	138	71	147	72	140	**156**	137	349	144	344	137	366	137	369	137	386
67	102	145	94	145	110	146	110	148	110	145	**157**	141	346	140	349	133	343	141	354	133	360
68	95	143	93	145	89	146	90	150	93	144	**158**	329	381	318	362	318	370	318	386	318	385
69	117	145	116	146	116	146	117	150	117	146	**159**	194	358	184	363	184	355	184	367	184	368
70	140	150	139	147	131	153	129	151	143	151	**160**	198	370	188	356	188	356	188	355	188	352
71	90	146	91	150	84	149	84	146	83	149	**161**	111	357	119	367	115	361	111	345	117	358
72	281	167	280	166	269	165	287	170	269	168	**162**	354	395	336	387	336	398	336	379	336	396
73	259	167	271	164	252	167	260	171	260	167	**163**	333	395	315	399	333	400	315	402	315	401
74	94	150	94	149	84	152	90	154	90	150	**164**	350	386	332	378	350	404	332	394	350	404
75	292	166	263	166	271	168	290	173	272	168	**165**	117	350	117	364	123	359	117	375	117	345
76	258	167	256	168	260	166	256	167	248	166	**166**	321	389	321	385	339	388	321	397	321	385
77	436	175	468	175	452	175	420	179	428	175	**167**	338	385	338	398	345	400	338	396	338	392
78	224	163	208	163	208	165	224	167	209	164	**168**	536	413	544	404	536	429	564	415	565	412
79	418	174	424	177	416	178	392	180	424	175	**169**	572	397	544	399	544	416	544	429	544	411
80	269	166	263	165	272	166	272	167	263	166	**170**	568	406	558	409	558	415	558	419	558	403
81	145	162	144	161	145	159	144	160	148	160	**171**	182	372	181	374	179	395	173	401	181	406
82	402	178	395	177	399	176	397	178	393	175	**172**	406	413	415	426	413	416	406	431	406	407
83	365	175	365	176	359	177	381	175	362	177	**173**	448	423	434	409	457	423	449	423	449	427
84	150	162	146	160	146	161	148	161	152	161	**174**	471	417	469	420	457	431	457	423	457	429
85	400	176	387	176	414	177	400	178	397	176	**175**	181	382	185	383	179	388	181	383	181	389
86	368	176	376	176	368	177	366	178	366	177	**176**	718	423	718	424	718	460	718	431	718	446
87	604	186	616	183	604	185	620	184	640	184	**177**	784	438	784	455	784	463	784	435	784	440
88	560	183	567	182	560	181	569	182	547	186	**178**	522	425	522	416	548	427	522	422	522	430
89	343	174	343	175	343	173	351	174	333	175	**179**	377	405	377	426	377	431	377	418	377	413
90	529	182	536	180	506	182	524	183	532	181	**180**	804	434	827	425	804	432	804	456	824	433

PSO/SA Heuristic Procedure 6

Problem #	$c_1 = c_2 = 0.25$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.30$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.35$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.40$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.45$ Min. Cost	Total Time in Seconds
1	7	7	7	7	7	7	7	7	7	8
2	14	8	14	8	14	8	14	8	14	8
3	9	8	9	8	9	8	9	8	9	7
4	7	8	7	7	7	7	7	7	7	8
5	16	8	16	8	16	8	16	8	16	8
6	14	9	14	8	14	8	14	8	14	8
7	9	8	9	8	9	8	9	8	9	8
8	16	11	16	8	16	8	16	8	16	8
9	11	9	11	8	11	8	11	8	11	8
10	17	10	17	8	17	8	17	8	17	8
11	18	9	18	8	18	9	18	8	18	9
12	61	12	61	10	61	10	61	10	61	9
13	49	11	49	9	49	10	49	9	49	10
14	48	11	48	10	48	9	48	9	48	10
15	61	10	61	9	61	10	61	10	61	10
16	49	10	49	10	49	10	49	9	49	9
17	92	11	92	10	92	10	92	10	92	10
18	92	11	92	10	92	10	92	10	92	10
19	44	10	44	9	44	9	44	10	44	10
20	54	11	54	10	54	10	54	9	54	10
21	35	10	35	9	35	10	35	9	35	9
22	95	12	95	11	95	10	95	11	95	11
23	82	11	82	10	82	10	82	10	82	10
24	35	10	35	10	35	10	35	10	35	10
25	95	11	95	10	95	11	95	10	95	10
26	82	12	82	11	82	11	82	10	82	11
27	142	12	142	11	142	11	142	11	142	11
28	142	11	142	11	142	11	142	11	142	11
29	98	11	98	10	98	10	98	11	98	11
30	113	12	113	11	113	11	113	10	113	11
31	20	180	19	164	23	161	20	166	19	173
32	37	194	37	185	37	190	39	177	39	175
33	31	207	27	183	27	177	27	177	27	183
34	23	173	23	158	21	160	19	161	21	167
35	33	175	39	178	41	172	38	170	41	192
36	27	179	34	173	32	175	32	162	30	173
37	44	195	44	185	45	172	44	177	44	193
38	68	182	72	176	60	178	68	183	60	183
39	59	173	59	181	58	179	49	175	52	173
40	124	190	111	182	124	193	123	180	131	196
41	79	194	79	192	79	189	79	190	79	203

Problem #	$c_1 = c_2 = 0.25$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.30$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.35$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.40$ Min. Cost	Total Time in Seconds	$c_1 = c_2 = 0.45$ Min. Cost	Total Time in Seconds
91	42	215	42	218	42	209	42	212	42	210
92	80	238	80	226	80	231	79	230	79	234
93	75	226	75	230	75	226	75	221	75	223
94	42	216	42	215	42	214	42	220	42	212
95	80	233	80	228	80	224	80	222	80	232
96	76	228	76	237	76	223	76	223	76	230
97	107	244	107	232	107	234	109	233	107	237
98	114	225	114	239	114	235	114	236	114	237
99	101	232	101	233	101	222	101	224	101	235
100	169	242	169	249	162	241	165	238	164	246
101	86	263	86	256	86	248	86	254	86	245
102	260	285	260	290	260	271	260	276	260	285
103	236	277	236	281	236	276	236	283	236	279
104	86	260	86	259	86	246	86	259	86	267
105	260	280	260	276	260	262	260	276	260	280
106	236	313	236	287	236	268	236	279	236	280
107	410	300	410	287	410	284	410	303	410	286
108	212	299	212	279	212	277	212	267	212	263
109	218	272	218	265	218	275	218	262	218	292
110	430	282	430	289	430	282	430	285	430	290
111	139	274	139	266	139	273	142	265	142	272
112	379	281	379	289	373	303	380	290	373	324
113	356	281	356	316	356	297	356	299	362	300
114	139	276	139	262	139	271	139	259	139	282
115	603	296	612	314	603	308	603	308	603	321
116	373	285	383	308	373	290	383	310	373	294
117	366	289	356	314	356	313	366	299	356	312
118	598	304	620	299	618	303	598	318	598	306
119	537	301	528	294	528	308	528	310	538	302
120	751	330	751	315	751	319	751	301	762	318
121	50	351	53	355	55	330	53	353	54	378
122	109	374	115	396	109	349	105	377	105	377
123	106	370	104	385	107	356	107	369	107	371
124	56	353	58	377	56	347	58	357	60	348
125	275	447	268	410	259	400	256	406	267	406
126	170	381	165	407	170	394	163	381	175	417
127	160	388	164	379	161	379	157	393	167	400
128	270	420	270	418	264	383	270	418	272	419
129	168	376	177	383	179	389	173	413	177	394
130	229	407	231	394	225	397	228	413	231	403
131	113	381	109	396	111	384	109	384	112	382

42	278	203	258	199	278	196	278	200	278	206	**132**	311	419	317	424	311	408	311	399	311	426
43	256	202	239	198	254	197	259	197	259	205	**133**	326	421	326	424	321	406	327	412	316	414
44	77	199	79	185	79	193	79	190	79	201	**134**	116	393	114	383	112	373	112	401	116	395
45	365	217	394	221	384	220	374	209	385	219	**135**	523	443	531	438	523	428	523	429	523	446
46	279	205	279	197	279	194	270	202	279	198	**136**	320	423	314	428	314	413	314	420	319	432
47	260	201	260	201	260	196	260	196	260	201	**137**	329	432	324	418	329	392	324	430	329	425
48	394	217	394	210	384	212	394	208	394	221	**138**	526	439	534	442	534	436	526	454	526	450
49	247	209	259	203	259	208	259	206	259	214	**139**	550	434	534	448	550	436	550	433	550	443
50	215	199	215	196	215	195	215	193	215	198	**140**	577	441	587	439	591	431	577	434	587	458
51	124	201	128	206	127	205	120	203	127	212	**141**	196	414	193	430	195	420	197	413	195	410
52	359	236	382	216	369	215	382	215	382	216	**142**	482	436	478	440	482	437	470	441	485	449
53	338	225	354	221	354	211	344	217	354	232	**143**	495	461	487	443	496	443	492	460	495	447
54	128	204	118	214	128	202	120	197	118	213	**144**	202	414	200	411	204	394	204	408	196	423
55	604	233	596	228	556	232	603	226	556	240	**145**	787	462	783	463	780	453	782	465	784	456
56	366	227	366	223	375	221	355	214	381	215	**146**	488	429	488	463	488	428	488	441	488	451
57	339	220	339	212	339	216	353	216	332	223	**147**	506	444	503	460	503	440	500	460	495	452
58	463	221	501	233	506	220	448	231	473	226	**148**	782	445	784	460	788	444	782	460	786	457
59	387	226	381	242	383	225	387	213	383	220	**149**	482	469	482	441	490	447	487	438	485	452
60	443	231	450	225	415	230	450	225	443	225	**150**	1565	475	1611	488	1629	481	1629	400	1658	484
61	35	210	38	213	38	208	38	195	35	206	**151**	66	549	66	513	62	511	62	510	62	529
62	76	232	76	223	76	217	76	215	68	223	**152**	229	597	246	570	229	569	246	605	229	590
63	68	228	62	222	69	215	69	225	62	229	**153**	131	576	131	559	139	555	131	550	131	579
64	37	216	41	208	39	200	39	209	39	217	**154**	127	558	127	519	127	591	137	566	127	560
65	78	225	78	234	78	225	75	215	79	233	**155**	70	506	68	555	68	506	68	526	68	512
66	71	218	69	224	71	220	69	230	72	224	**156**	137	580	145	574	137	583	137	561	137	569
67	110	219	102	227	110	224	110	219	102	224	**157**	133	552	138	577	133	565	133	560	133	601
68	87	230	93	215	91	233	91	222	93	226	**158**	318	562	327	582	318	548	318	594	318	611
69	103	231	114	229	118	228	116	216	115	242	**159**	194	569	184	567	184	549	184	537	194	595
70	143	234	143	236	145	234	142	239	131	233	**160**	198	565	188	575	188	567	198	568	188	582
71	90	246	83	233	90	228	89	230	91	239	**161**	111	570	111	536	111	552	111	562	111	552
72	282	274	287	254	269	257	274	257	263	268	**162**	336	627	354	596	336	589	336	605	356	653
73	241	260	241	268	247	261	241	264	249	274	**163**	333	606	333	629	315	613	315	598	333	654
74	90	229	92	238	84	230	90	236	96	234	**164**	332	619	339	599	332	608	332	616	350	658
75	282	265	270	261	271	255	285	261	263	261	**165**	123	584	117	537	123	564	117	530	117	591
76	260	262	241	262	248	264	250	245	250	263	**166**	328	659	321	610	321	614	339	600	321	595
77	436	275	448	277	448	283	448	275	436	256	**167**	338	611	338	618	357	631	350	616	345	623
78	224	277	209	259	224	249	208	251	208	249	**168**	536	608	536	597	536	612	536	657	561	637
79	424	292	404	267	405	268	404	264	421	282	**169**	544	658	574	612	560	641	544	683	544	622
80	275	273	250	261	272	257	263	254	268	278	**170**	558	620	558	673	589	640	558	613	605	623
81	141	256	146	244	145	250	149	264	145	263	**171**	179	594	182	651	182	584	179	627	180	614
82	397	316	400	266	400	259	380	292	390	271	**172**	406	636	406	642	406	668	414	644	413	611
83	363	280	365	283	372	273	348	303	374	289	**173**	434	656	442	631	434	659	448	680	454	655
84	142	259	142	255	150	240	154	267	154	252	**174**	472	643	457	628	457	656	457	634	457	645
85	400	287	400	269	402	272	400	278	416	272	**175**	181	618	181	568	179	590	181	572	179	600
86	382	289	366	278	348	265	392	279	359	275	**176**	718	646	718	673	718	641	718	654	718	712
87	596	299	620	283	628	274	592	291	612	284	**177**	784	683	784	673	784	666	811	708	784	657
88	569	309	547	283	558	267	541	278	544	293	**178**	522	662	546	626	522	645	540	649	540	652
89	343	269	343	274	338	257	346	271	336	269	**179**	377	640	392	650	377	631	392	635	377	636
90	503	273	528	275	528	267	511	273	514	274	**180**	804	650	827	665	824	669	804	690	804	679

Vita

Hadeel Y. Al Sayegh was born in 1982, in Dubai, United Arab Emirates. She attended Al Mawakeb School in Dubai before moving to Canada. She graduated from St. Patrick's High School in 2000. She enrolled in the University of Ottawa in Ontario, Canada, from which she graduated with magna cum laude, in 2004. Her degree was a Bachelor of Science in Information and Management Systems from the School of Information Technology and Engineering (SITE).

In 2008, Ms. Al Sayegh began a master's program in Engineering Systems Management at the American University of Sharjah. She earned the Master of Science degree in Engineering Systems Management in 2011.