University of Louisville
ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

5-2010

Preventive maintenance and replacement scheduling : models and algorithms.

Kamran S. Moghaddam
University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

Recommended Citation
Moghaddam, Kamran S., "Preventive maintenance and replacement scheduling : models and algorithms." (2010). *Electronic Theses and Dissertations.* Paper 994.
https://doi.org/10.18297/etd/994

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

PREVENTIVE MAINTENANCE AND REPLACEMENT SCHEDULING:
MODELS AND ALGORITHMS

By

Kamran S. Moghaddam
B.S., Applied Mathematics, University of Tehran, 2001
M.S., Industrial Engineering, Tehran Polytechnic, 2003

A Dissertation
Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements for the

Doctor of Philosophy

Department of Industrial Engineering
University of Louisville
Louisville, Kentucky, USA

May 2010

©Copyright 2010 by Kamran S. Moghaddam
All Rights Reserved

PREVENTIVE MAINTENANCE AND REPLACEMENT SCHEDULING:

MODELS AND ALGORITHMS

By

Kamran S. Moghaddam
B.S., Applied Mathematics, University of Tehran, 2001
M.S., Industrial Engineering, Tehran Polytechnic, 2003

A Dissertation Approved on

April 23, 2010

By the following Dissertation Committee

Dr. John S. Usher, Committee Chair ____________________

Dr. Gerald W. Evans ____________________

Dr. Sunderesh S. Heragu ____________________

Dr. Gail W. DePuy ____________________

Dr. Ali M. Shahhosseini ____________________

DEDICATION

This dissertation is dedicated to my parents

Houshang and Afsaneh

My brothers

Shahram and Shaheen

And my beloved girlfriend

Anahita

ACKNOWLEDGMENTS

I would like to thank Dr. John S. Usher, without whose guidance this dissertation would have been impossible. John has been generous and supportive, academically, professionally, and personally, and is a true role model to me.

I want to express my gratitude to Dr. Sunderesh S. Heragu for giving me a post-doctoral research opportunity. I would like to thank my dissertation committee, Dr. Gerald W. Evans, Dr. Sunderesh S. Heragu, Dr. Gail W. DePuy, and Dr. Ali M. Shahhosseini, who have been of great help throughout my graduate career and job-searching process. I appreciate the time that Dr. Gail W. DePuy took to read and edit my job-searching documents. I especially want to thank Dr. Ali M. Shahhosseini and his wife Dr. Samin Rezania for their help and support after my eye injury and during the recovery period after eye surgeries.

I would like to thank my mother, Afsaneh, my departed father, Houshang, and my brothers Shahram and Shaheen. This achievement is due in no small part to their love, support, and generosity.

My advisors during my graduate years at Tehran Polytechnic, Dr. Abbas Seifi and Dr. Jamshid Mousavi, were formative in my academic development, and I thank them for helping me develop as a student and researcher. I also wish to thank the rest of the faculty and my fellow graduate students in the Department of Industrial Engineering at the University of Louisville for creating a productive environment of intellectual growth and for supporting me as a student, instructor, and researcher.

Most especially, I want to thank my girlfriend, Anahita, an amazing friend and partner from whom I have learned so much about generosity, dedication, and forgiveness; and for her understanding and patience during those times when there was no light at the end of anything. I hope I can repay some of the support, encouragement, and love she has given me.

ABSTRACT

PREVENTIVE MAINTENANCE AND REPLACEMENT SCHEDULING: MODELS AND ALGORITHMS

Kamran S. Moghaddam

April 23, 2010

Preventive maintenance is a broad term that encompasses a set of activities aimed at improving the overall reliability and availability of a system. Preventive maintenance involves a basic trade-off between the costs of conducting maintenance/replacement activities and the cost savings achieved by reducing the overall rate of occurrence of system failures. Designers of preventive maintenance schedules must weigh these individual costs in an attempt to minimize the overall cost of system operation. They may also be interested in maximizing the system reliability, subject to some sort of budget constraint.

In this dissertation, we present a complete discussion about the problem definition and review the literature. We develop new nonlinear mixed-integer optimization models, solve them by standard nonlinear optimization algorithms, and analyze their computational results. In addition, we extend the optimization models by considering engineering economy features and reformulate them as a multi-objective optimization model. We optimize this model by generational and steady state genetic algorithms as well as by a simulated annealing algorithm and demonstrate the computational results obtained by implementation of these algorithms. We perform a sensitivity analysis on the parameters of the optimization models and present a comparison between exact and metaheuristic algorithms in terms of computational efficiency and accuracy. Finally, we present a new mathematical function to model age reduction and improvement factor parameter used in optimization models. In addition, we develop a practical procedure to estimate the effect of maintenance activity on failure rate and effective age of multi component systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Preventive Maintenance and Replacement Scheduling

Preventive maintenance is a broad term that encompasses a set of activities aimed at improving the overall reliability and availability of a system. All types of systems, from conveyors to cars to overhead cranes, have prescribed maintenance schedules set forth by the manufacturer that aim to reduce the risk of system failure. Preventive maintenance activities generally consist of inspection, cleaning, lubrication, adjustment, alignment, and/or replacement of sub-components that wear-out. Regardless of the specific system in question, preventive maintenance activities can be categorized in one of two ways, component maintenance or component replacement. An example of component maintenance would be maintaining proper air pressure in the tires of an automobile. Note that this activity changes the aging characteristics of the tires and, if done correctly, ultimately decreases their rate of occurrence of failure. An example of component replacement would be simply replacing one or more of the tires with new ones.

Obviously, preventive maintenance involves a basic trade-off between the costs of conducting maintenance/replacement activities and the cost savings achieved by reducing the overall rate of occurrence of system failures. Designers of preventive maintenance schedules must weigh these individual costs in an attempt to minimize the overall cost of system operation. They may also be interested in maximizing the system reliability, subject to some sort of budget limitation. Other criteria such as availability and demand satisfaction might be considered as the objective functions, but they will not be studied in this dissertation. The main problem is to find the best sequence of maintenance and replacement actions for each component in the system in each period over a planning horizon such that total costs are minimized subject to a constraint on reliability of the system or the overall reliability of the system is maximized subject to a constraint on budget of the system.

1.2. Research Contributions

In this dissertation, new optimization models, designed to find the optimal preventive maintenance and replacement schedules, are developed and solved via exact, and heuristic algorithms. In addition, a new mathematical age reduction and improvement factor model is developed. These models can be considered as the main research contributions. In particular, the following contributions are made:

1. Two optimization models will be constructed based on extensions of previous work in particular, by Usher *et al* (1998). The optimization models are solved using a standard nonlinear mixed-integer programming algorithm. These models also provide a general framework to achieve optimal preventive maintenance and replacement policies and, with modifications, can be used as basic closed-form models for any type of system.
2. A multi-objective optimization model is developed based on a set of basic assumptions and engineering economy considerations. This model is optimized

via generational and steady state genetic algorithms as well as by a simulated annealing algorithm, which provide Pareto optimal solutions.

3. A sensitivity analysis on parameters of optimization models is performed and an extensive comparison of computational performance and accuracy of exact and metaheuristics algorithms is presented.
4. A new mathematical model for estimating age reduction and the improvement factor parameter used in optimization models is constructed and analyzed. In addition, a practical procedure is developed to estimate age reduction and the improvement factor parameter in maintainable and repairable systems.

1.3. Outline

The remainder of this dissertation is organized as follows. In Chapter 2, a comprehensive literature review of models, algorithms and, applications of preventive maintenance and replacement scheduling is presented. In Chapter 3, system configuration and formulation of the optimization models are presented and their computational results are analyzed. Chapter 4 includes an extension of the Chapter 3 optimization models by introducing engineering economy parameters into a multi-objective optimization model. This model has been optimized by multi-objective generational and steady state genetic algorithms as well as by a multi-objective simulated annealing algorithm, and the computational results obtained by implementation of these algorithms are demonstrated.

Chapter 5 deals with a sensitivity analysis on the parameters of the optimization models and also presents a comparison between of exact and heuristic algorithms in terms of computational efficiency and accuracy. Chapter 6 reviews current age reduction and improvement factor models and introduces a new mathematical function and a practical procedure to estimate age reduction and the improvement factor parameter. Finally, in Chapter 7, conclusions and potential directions for future research are presented.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

This chapter has three main sections. The first section presents a complete review of various optimization models and algorithms related to preventive maintenance and replacement scheduling. Section 2.3 presents a review of key work that utilizes simulation models of preventive maintenance and replacement scheduling. In Section 2.4, approaches that develop and use age reduction and improvement factor models are presented. We also review the applications of preventive maintenance and replacement models in a wide variety of systems such as in manufacturing and production systems, service systems, and power systems. Finally, we discuss potential research areas and summarize the reviewed papers.

2.2. Optimization Models

2.2.1. Analytical Methods

Analytical methods have been broadly used as a standard optimization approach to achieve optimal maintenance and replacement schedules in engineering problems. Canfield (1986) studies preventive maintenance optimization models by focusing on different aspects of the failure function on systems reliability. He mentions that preventive maintenance actions do not change or affect deteriorating behavior of failure rate, so the assumed failure function is unchanged with maintenance actions. He assumes an increasing failure rate based on the Weibull distribution function for his study and determines an optimal cost of maintenance policies by defining the average cost-rate of system operation and applying analytical method as the solution approach. McClymonds and Winge (1987) present methods to achieve optimal preventive maintenance schedules for nuclear power plants, though they have not been applied successfully. They consider plant availability and reliability as the objective functions and develop models based on assigning resources to preventive and corrective maintenance activities.

Martin (1988) presents a preventive maintenance optimization model, which has been developed, and implemented by Columbia Hospital in Milwaukee based on plant technology and safety management standards. The hospital designed this program in order to use the optimal preventive maintenance plan for its electrical distribution equipment to improve safety, serviceability, reliability and total cost. Hsu (1991) develops an optimization model in order to determine optimal preventive maintenance schedules for a serial multi-station manufacturing system. He mentions that most models use a simulation approach at that time but his model is focused on a mathematical programming approach. The computational results of his study show that the operating features of the stations are interrelated and one must investigate the effect of preventive maintenance activities on all stations at the same time.

Jayabalan and Chaudhuri (1992) present two different preventive maintenance scheduling models for maintaining bus engines in a public transit network based on minimization of total cost over a finite planning horizon. They construct the models based on the concept of mean time to failure (MTTF) of the engines and assume an upper bound for the failure rates. The first model is based on different Weibull failure functions between preventive maintenance activities and the second model assumes that the each preventive maintenance action reduces the effective age of the system by a certain amount. The authors present computational results and show the effectiveness of the models in a real case study. Westman and Hanson (2000) develop a mathematical model to determine the mean time to failure (MTTF) as a function of uptime for a workstation in a multi-stage manufacturing system. The authors assume that the uptime of the workstation has an increasing rate and is reduced if preventive maintenance actions are performed. They mention that this methodology captures the flexibility and multi-stage properties of manufacturing systems and can generate preventive maintenance policies.

Fard and Nukala (2004) study and review the application of different stochastic process such as homogenous Poisson process (HPP), non-homogenous Poisson process (NHPP), branching Poisson process (BPP), and superimposed renewal process (SRP) in preventive maintenance scheduling problems. They present current methods based on non-homogenous processes for modeling and optimization of single and multi-component systems. They assume that maintenance actions do not affect the failure rate of system; hence, they suggest that non-homogenous Poisson process can be applied and used to model the failure rate of repairable service systems.

Ying *et al.* (2005) develop an integrated optimization model that simultaneously considers preventive maintenance and production scheduling decision variables. Their model minimizes total tardiness of jobs and makes a 30% reduction in expected total tardiness of jobs. Pongpech *et al.* (2006) present an optimization model that minimizes total maintenance costs and penalty costs for used equipment under lease. They assume a Weibull distribution function for failure rate of equipment, develop a 4-parameter model, and develop a 4-stage algorithm based on an analytical approach to solve it. They apply their model to several numerical examples with different contract assumptions and find optimal policy in each situation.

Panagiotidou and Tagaras (2007) develop an optimization model that optimizes preventive maintenance schedules in a manufacturing process. The authors consider two different states for components, in-control or out-of-control, and before complete failure. They treat the time to shift and the time to failure as random variables and express them with Weibull and Gamma distribution functions. In addition, they combine age-based and condition-based concepts into the optimization model with the minimization of total cost and solve it by applying Karush-Kuhn-Tucker (KKT) conditions of optimality to obtain an optimal preventive maintenance schedule. Finally, they present several numerical examples to demonstrate the effectiveness of their methodology.

Shirmohammadi *et al.* (2007) develop an age-based nonlinear optimization model to determine an optimal preventive maintenance schedule for a single-component system. They define two types of decision variables, time between preventive replacements and cut-off age, and assume an expected cost of failures, maintenance, replacement costs, and total cycle cost in the cost function and consider cost per unit time as the objective function. In order to solve the optimization model and show the effectiveness of the proposed approach, they utilize MAPLE and solve the model for a numerical example by setting different values for an improvement factor, which is assumed as a constant in the model.

2.2.2. Exact Algorithms

Westman *et al.* (2001) formulate a mathematical model to find an optimal production schedule via a Gaussian Poisson function with state dependent Poisson process. They consider the total cost of production and maintenance scheduling as the objective function and use a stochastic dynamic programming approach, and demonstrate application of the model in a numerical example.

Yao *et al.* (2001) present a two-layer hierarchical model that optimizes the preventive maintenance schedules in semiconductor manufacturing operations. They develop a Markov decision process and optimize this model via a mixed-integer linear programming model. They define profit of cluster tools production as the objective function to be maximized and consider a time window for preventive maintenance activities and limitation of resources as nonlinear constraints. In order to achieve a global optimum, they transform the nonlinear functions into linear functions and use EasyModeler and OSL as the optimization software. In addition, they utilize AutoSched AP as the simulation software in order to construct a simulation model to evaluate the performance of the optimization model in a real case study with 11 preventive maintenance tasks in a one-week planning horizon and compare the obtained optimal results with the actual preventive maintenance plan. Later Yao *et al.* (2004) extend their previous model to be more general, apply this extended model to a production line of a semiconductor manufacturing system, and show the application of it via numerical examples.

Han *et al.* (2004) develop a nonlinear optimization model to minimize the total cost of maintenance and replacement actions under reliability constraints for production machine in a production system. Their model considers the Weibull distribution as the failure function of the machine and can be used as a decision support system for job shop scheduling. Jayakumar and Asgarpoor (2004) present a linear programming model in order to optimize the maintenance policy for a component with deterioration and random failure rate. They determine optimal mean times of minor and major preventive maintenance actions based on maximizing the availability of the component. They utilize MAPLE and LINGO to solve the linear programming model of their Markov decision process.

Zhao *et al.* (2005) present an age-based preventive maintenance optimization model for a gas turbine power plant. They develop a model with profit instead of cost as the objective function and considered power plant performance, reliability and market dynamics. In order to determine the effects of economics on maintenance costs and frequencies, they utilize a sequential approach and show its effectiveness by using real data of based load combined cycle power plant with a gas turbine unit. Canto (2006) presents an optimization model to schedule a preventive maintenance of a real power plant over a long-term planning horizon. He considers the total cost of various operations as the objective function and uses Bender's decomposition to solve a mixed-integer linear programming model.

Budai *et al.* (2006) present two mixed-integer linear programming models for preventive maintenance scheduling problems. The authors assume the total cost including possession costs, maintenance costs, and the penalty costs of early consecutive maintenance activities as the objective function for both models. They present and prove a theorem about the NP-hard structure of the preventive maintenance scheduling problems and use GAMS software to implement the optimization models. They use CPLEX as the optimization software to find an optimal preventive maintenance schedule. They apply their model to a case study of railway maintenance scheduling. In addition, they develop four heuristic optimization algorithms, two for each model, and compare the computational results obtained from exact algorithms in CPLEX with the results achieved from heuristic algorithms and mention the advantages of each solution methodology.

Another excellent study in this area is by Tam *et al.* (2006), who develop three nonlinear optimization models: one that minimizes total cost subject to satisfying a required reliability, one that maximizes reliability at a given budget, and one that minimizes the expected total cost including expected breakdown outages cost and maintenance cost. They utilize MS-Excel Solver as the optimization software that uses a generalized reduced gradient algorithm to solve the nonlinear optimization models. Using these models, they determine optimal maintenance intervals for a multi-component system but their models consider only maintenance actions for components and do not consider replacement actions.

Robelin and Madanat (2006) develop a maintenance optimization model for bridge decks via a Markov chain process. In this paper, they classify optimization models into two categories, (1) physically based deterioration models with a limited number of decision variables, and (2) simpler deterioration models with more and sophisticated decision variables. They apply a Markov chain methodology with states based on history of deterioration and maintenance actions and utilize dynamic programming as the solution approach to solve a Markov decision process. As a case study, they apply their approach to optimize the maintenance policy of bridges.

Alardhi *et al.* (2007) present a binary integer linear programming model in order to find the best preventive maintenance schedule in separated and linked cogeneration plants. The researchers define the availability of the power and desalting equipments as the objective function to be maximized, and consider the maintenance time window, maintenance completion duration, logical operational, resource limitation, maintenance crew availability, efficiency measures, and demand as the set of constraints. They apply their model in two co-generation plants with 7 units and 42 pieces of equipment in Kuwait, over a 52-week planning horizon, and utilize LINGO as the optimization software to optimize the model. In addition, they perform a sensitivity analysis on the model to assess the robustness and analyze the effect of expanding the planning horizon, reducing the resources, and increasing the demand on the maintenance strategies.

Kuo and Chang (2007) develop an integrated maintenance scheduling and production planning optimization model for a single machine based on a cumulative damage process and the effect of preventive maintenance strategies on production schedules in order to minimize total tardiness. The authors express that in the optimal strategy if jobs have a certain process time with different respective due dates, the optimal production schedule sorts the jobs by earliest due date and if jobs have certain due dates with different process time, it sorts them by shortest process time. In addition, they mention that the optimal maintenance policy is a constraint on the production schedule when the machine shuts down due to cumulative damage failure process. The computational results achieved by dynamic programming show that by increasing the number of jobs the effect of jobs due dates on the optimal maintenance policy is decreased.

2.2.3. Heuristics and Meta-Heuristics Algorithms

Genetic algorithms, as a major optimization approach, have been presented in several research papers. Usher *et al.* (1998) present an optimization maintenance and replacement model for a single-component system. They determine an optimal preventive maintenance schedule for the system subject to deterioration by considering the time value of money in all future costs, the cost of the increasing rate of occurrence of failure over time and the use of an improvement factor to provide for the case of imperfect maintenance actions. In addition, they provide a comparison of computational results among random search, genetic algorithm, and branch and bound algorithms.

One of the most notable studies in the area of reliability and maintenance optimization for multi-state multi-component systems is found in Levitin and Lisnianski (2000). They define a multi-state system in which all or some of the components have different performance levels, from proper functioning to complete failure and the reliability of the system as its ability of satisfying the demand levels. They formulate an optimization model to determine preventive maintenance schedules that affect the effective age of components. Their model is based on minimization of cost subject to a required level of reliability. They apply a universal generating function technique and use a genetic algorithm to determine the best maintenance strategy. Levitin and Lisnianski (2000) present additional research in which an optimization model was developed in order to find an optimal replacement schedule in multi-state series-parallel systems. They consider an increasing failure rate based on the expected number of failures during time intervals and define the summation of maintenance activities cost along with cost of unsupplied demand due to failures of components as the objective function. Finally, they utilize a universal generating function approach and apply a genetic algorithm to find an optimal maintenance policy.

Wang and Handschin (2000) develop a new genetic algorithm by modifying the basic operators, crossover and mutation operators of a standard genetic algorithm based on the specific characteristics of a preventive maintenance scheduling problem for power systems. They improve the computational complexity of their genetic algorithm by considering a code-specific and constraint-transparent integrated coding method to achieve faster convergence and to prevent production of infeasible solutions. As the implementation methodology, an object oriented programming approach is applied and the effectiveness of the new genetic algorithm shown via theoretical analysis and simulation results to compare with a traditional genetic algorithm.

Tsai *et al.* (2001) consider two activities, imperfect maintenance, and replacement, in their preventive maintenance optimization model. They model imperfect maintenance activities based on the concept of an improvement factor, which is determined by a quantitative assessment procedure. They use a genetic algorithm to find an optimal preventive maintenance schedule while the system unit-cost life is considered as the objective function. As a case study, they test a mechatronic system to show the effectiveness of their proposed model and algorithm.

Cavory *et al.* (2001) present an optimization model to schedule preventive maintenance tasks of all machines in a single-product manufacturing production line. They assume that each machine should be assigned to each operator and considered the total throughput of the line as the objective function to be maximized. At the first step, they formulate the optimization model and analyze it via an analytical approach. Then, they used C++ as a programming environment and applied a genetic algorithm in order to find the best combination of preventive maintenance tasks. In addition, they construct an experimental design to set and analyze the parameters of the genetic algorithm. Then, they utilize the Taguchi method and statistical analysis to validate the results. Finally, an application of the proposed approach is performed in an actual production line of car engines.

Leou (2003) presents an optimization model to find an optimal preventive maintenance schedule in a multi-component system. He considers the total cost of operations and maintenance activities along with reliability as the main criteria of the system and transfers them into the objective function by defining the degree of violation from required reliability. In addition, he defines the maintenance crew and duration of maintenance as the constraints of the system. He applies his optimization model in a case study with six electric generators and utilizes a genetic algorithm as the optimization methodology to determine the best preventive maintenance schedule.

Han *et al.* (2003) consider the recursive nature of the failure rate between preventive maintenance cycles and develop a nonlinear optimization model based on repair cost, preventive maintenance cost, and production loss cost in a production system. They apply a genetic algorithm as the optimization technique and mention that their model can be considered in decision support systems for maintenance and job shop scheduling. Bris *et al.* (2003) consider cost and availability as the systems criteria in their research. They optimize a mathematical model including cost in the objective function and availability as the constraint by using a genetic algorithm to find the best preventive maintenance schedule. They use a time-dependent Birnbaum importance factor to generate the ordered sequence of inspection times and utilize MATLAB to calculate the system availability via a Monte Carlo simulation approach.

Adzakpa *et al.* (2004) present an application of combined maintenance scheduling and job assignment model of distribution systems. They develop an optimization model that considers total cost of maintenance actions as the objective function and availability in a given time-window and precedence among consecutive standby jobs and their emergency as constraints of the model. They show that their model is NP-hard to solve and because of that, they use a heuristic optimization algorithm to solve the model. Li and Qian (2005) present a real time preventive maintenance optimization model for cluster tools in a semiconductor manufacturing system. They consider the standpoint of the system and used a genetic algorithm as the solution procedure.

Samrout *et al.* (2005) use an ant colony algorithm to solve the problem that was previously optimized via a genetic algorithm. They define maintenance and inspection periods for series of components and use MATLAB as the programming environment to solve their model and compare the computational results with the results obtained by genetic algorithm. Sortrakul *et al.* (2005) present an optimization model of integrated preventive maintenance scheduling and production planning for a single machine. The authors mention that these problems have been tackled separately in several papers but they have not been considered together in real manufacturing systems. They consider the total weighted expected job completion time as the objective function and optimize the combinatorial optimization model via a genetic algorithm. As the result, they express the advantages and effectiveness of their approach, which can be used to solve actual manufacturing problems.

Cassady and Kutanoglu (2005) develop and present an integrated preventive maintenance and production scheduling mathematical model for a single-machine.

They consider total weighted expected completion time as the objective function that should be minimized. Their model allows multiple maintenance activities and explicitly captures the risk of not performing maintenance actions. They employ a heuristic approach to solve the model and compare obtained computational results of an integrated model with the results achieved from solving preventive maintenance scheduling and job scheduling problems independently.

El-Ferik and Ben-Daya (2006) present an age-based hybrid model for imperfect preventive maintenance scheduling problem. The authors review different policies and the models developed by other researchers and propose a new sequential age-based analytical model. They assume that the imperfect preventive maintenance activities reduce the effective age of the system but increase the failure rate and presented mathematical formulations to determine the adjustment factors for both failure rate and age reduction coefficient. They construct an optimization model based on their analytical models, consider the total cost as the objective function, and solve the optimization model via a new heuristic algorithm in a numerical example.

Duarte *et al.* (2006) present a model and a heuristic algorithm for maintenance scheduling of a system with a series of components. In this research, they assume that all components have linearly increasing failure rates with a constant improvement factor for imperfect maintenance. In addition, they consider the total cost as the objective function and the total downtime as the main constraint. In terms of maintenance activities, they define preventive and corrective maintenance for each component. Finally, their algorithm optimizes the interval of time between maintenance actions for each component over a planning horizon.

Limbourg and Kochs (2006) propose several techniques to represent the decision variables in preventive maintenance scheduling models that use heuristics and meta-heuristics optimization algorithms. They test various non-standard approaches and compare them to binary representations by a heuristic algorithm and the computational results show the effectiveness of their approach. In addition, they apply some modified crossover and mutation procedures in a genetic algorithm and show the improvement in performance of the algorithm in terms of computational time and accuracy.

Additional research on the application of genetic algorithms to maintenance optimization has been done by Lapa *et al.* (2006). They consider flexible intervals between maintenance actions and mention the advantage of this assumption over the common methodologies of continuous fitting of the schedules. They develop a mathematical model that includes preventive and corrective maintenance actions and the associated cost with them, outage times, reliability of the system, and probability of imperfect maintenance. Because their model is a nonlinear large-scale optimization model, they utilize a genetic algorithm as the solution procedure. In addition and as a case study, they apply their model to a high-pressure injection system to measure the effectiveness of their methodology.

Shum and Gong (2007) recently present an application of a genetic algorithm to optimize preventive maintenance schedules of a production machine. They consider maintenance and replacement frequency along with purchasing strategy and the size of the maintenance workforce as the decision variables and total cost as the objective function. They examine the effect of these costs on the optimal maintenance schedule in a numerical example. Other meta-heuristics have been used as the combinatorial optimization techniques to solve maintenance scheduling problems. Zhou *et al.* (2007) demonstrate an age based preventive maintenance scheduling model combined with production planning optimization model in order to maximize availability of a production machine. The authors use a heuristic algorithm to obtain an optimal schedule that minimizes the makespan. They also apply a simulation approach to validate the heuristic algorithm and to show its effectiveness in solving flow shop scheduling problems of integrated production planning with preventive maintenance scheduling.

2.2.4. Hybrid Models and Algorithms

Kim *et al.* (1994) combine a genetic algorithm with a simulated annealing in order to optimize a large-scale and long-term preventive maintenance and replacement scheduling problem. In their research, the acceptance probability of a simulated annealing method is considered as a measure for individual survival in the genetic algorithm. By using this approach, they achieve a near optimal solution in a short period of time compared to the computational time of a simple genetic algorithm. As a case study, they optimize a long-term maintenance scheduling problem of a thermal system and show the effectiveness of their model.

Tan and Kramer (1997) develop a general framework for preventive maintenance optimization problems in chemical process operations. They assume a Weibull distribution function for failure rate and consider different maintenance activities that can be performed. They develop a methodology that combines a Monte Carlo simulation with a genetic algorithm to solve opportunistic maintenance problems with a non-deterministic objective function. They apply their approach to two case studies to compare the results obtained from their proposed model with the results achieved from an analytic approach, and the Monte Carlo simulation with a neural network. Finally, they mention the advantages of their approach over other approaches.

Marseguerra *et al.* (2002) develop a condition-based maintenance scheduling model for multi-component systems and use a Monte Carlo simulation model to predict the degradation level in a continuously monitored system. They apply a genetic algorithm to optimize the degradation level after maintenance actions in a multi-objective optimization model with profit and availability as the objective functions. In addition, they consider a simulation model to describe the dynamics of a stress-dependent degradation process in load-sharing components. Based on the computational results, they mention that the combination of a genetic algorithm with Monte Carlo simulation is an effective approach to solve combinatorial maintenance scheduling optimization models.

Charles *et al.* (2003) present a preventive maintenance optimization model in order to minimize total maintenance costs in a production system. In this paper, they consider productive maintenance, corrective maintenance and preventive maintenance actions along with production operations as well as the related associated costs. They assume a Weibull distribution function for failure rate and utilize MELISSA C++ as discrete-event production-oriented simulation software to evaluate different scenarios. As a case study, they analyze a prototype semiconductor manufacturing workshop to demonstrate the proposed approach and mention that this model has general structure that can be applied for other kind of manufacturing systems.

Shalaby *et al.* (2004) develop an optimization model for preventive maintenance scheduling of multi-component and multi-state systems. They define sequence of preventive maintenance activities as decision variables and the summation of preventive maintenance, minimal repair, and downtime costs as the objective function. In addition, they consider system reliability, minimum intervals between maintenance actions, and crew availability as the constraints into the model. Finally, a combination of genetic algorithm and simulation was utilized to optimize the model. Allaoui and Artiba (2004) present a combination of simulation and optimization models in order to solve the NP-hard hybrid flow shop scheduling problem with maintenance constraints and multiple objective functions based on flow time and due date. In addition, they consider setup times, cleaning times, and transportation times in the model and mention that the performance of the algorithm can be affected by the number of breakdown times. Finally, they prove that the effectiveness of the simulated annealing algorithm is better than other heuristic algorithms with the same conditions.

Suresh and Kumarappan (2006) develop an optimization model and use a genetic algorithm combined with simulated annealing. The authors define customer satisfaction at the objective function and apply their method to determine an optimal preventive maintenance schedule in a power system. They mention that the method could produce better solutions if some changes and modification were made into the solution procedure. As a case study, they test the method on 62-unit state electrical system of Victoria and show the advantages of the their proposed approach. Samrout *et al.* (2006) present another paper about the combination of an ant colony algorithm and a genetic algorithm to optimize a large-scale preventive maintenance scheduling problem. They divide the objective function of their problem into two sections and then utilize each algorithm to improve each section separately.

They mention that using hybrid algorithm in a large-scale problem is more efficient than using a simple algorithm.

Jin *et al.* (2006) develop a preventive maintenance optimization model for a multi-component production process. They define a combination of mechanical service, repair, and replacement activities for each component and use Markov decision process to present the transition function of probability for maintenance activities over the planning horizon. In addition, they consider required reliability of the system as a constraint and total preventive maintenance cost as the objective function of the model. As the solution procedure, a simulation approach was utilized to find an optimal schedule. The authors describe that considering the combination of preventive maintenance activities can reduce more cost in comparison with the situation that different activities are considered separately.

Ruiz *et al.* (2007) present comprehensive research in the area of integrating preventive maintenance scheduling and production planning. They define three different policies for preventive maintenance schedules; preventive maintenance at fixed predefined time intervals, preventive maintenance for maximizing equipment availability, and maintaining a minimum reliability threshold over the planning horizon. The minimization of the total manufacturing time of the sequence is considered as the main criterion. The authors apply six different adaptations of heuristic and meta-heuristic algorithms to evaluate the last two policies for two sets of problems and mention that ant colony and genetic algorithm solve these problems effectively. Finally, they conclude that integrated preventive maintenance scheduling and production planning optimization problems along with meta-heuristic algorithms can be successfully applied in flowshop problems. In addition, they suggest that one can define more criteria and consider the problem as a multi-objective optimization model.

2.2.5. Multi-Objective Models and Algorithms

Multi-objective maintenance scheduling optimization models have been presented in several papers. Kralj and Petrovic (1995) present a novel approach in preventive maintenance scheduling of thermal generating systems. The authors develop a large-scale multi-objective combinatorial optimization model with three objective functions and a set of constraints. They consider minimization of total fuel costs, maximization of reliability in terms of expected unserved energy, and minimization of technological concerns as the objective functions. In addition, they define maintenance duration, maintenance continuity, maintenance season, maintenance sequence of thermal units of the same class, limitation on simultaneous maintenance of thermal units, and limitation on total capacity on maintenance due to labor and resources as the constraints of the model. They develop a multi-objective preventive maintenance scheduling software based on a multi-objective branch-and-bound algorithm implemented in FORTRAN. Finally, the researchers apply their methodology to a real system of 8 power plants with 21 thermal units with 11 maintenance classes over 31 weeks as the planning horizon.

Chareonsuk *et al.* (1997) develop a multi-criteria preventive maintenance optimization model to find an optimal preventive maintenance interval of components in a production system. In this study, the authors consider an age-based failure rate for components by fitting a Weibull distribution function to data and define expected total cost per unit time and the reliability of the production system as the main criteria. In following, they utilize a preference ranking organization method for enrichment evaluations (PROMETHEE) as the solution approach and define alternative decisions as the preventive maintenance intervals. By using this approach, they can aggregate preferences of alternatives by combining the weighted values of the preference functions of the complete set of criteria. As a case study, they apply their methodology in a paper factory and used PROMCALC as the optimization software. Finally, they mention the advantage of their approach in which decision makers and managers can input various criteria into the model and perform sensitivity analysis on the optimal solutions.

Leng *et al.* (2006) present an integrated preventive maintenance scheduling and production planning multi-objective optimization model for a single machine. They use a chaotic particle swarm optimization algorithm to solve the model and show its application and effectiveness via numerical examples. Konak *et al.* (2006) present a comprehensive study on multi-objective genetic algorithms and their applications in reliability optimization problems. They review 55 research papers and demonstrate the recent techniques and methodologies.

Quan *et al.* (2007) develop a novel multi-objective genetic algorithm in order to optimize preventive maintenance scheduling problems. They define the problem as a multi-objective optimization problem by considering the minimization of workforce idle time and the minimization of maintenance time and mention that there is a tradeoff between the objective functions. As the solution procedure, they use utility theory instead of dominance-based Pareto search to determine the non-inferior solutions and show the advantage of this method via a numerical example.

Verma and Ramesh (2007) integrate systems and sub-systems of a large engineering plant into higher modular assemblies and apply a multi-objective preventive maintenance scheduling approach. They model this problem as a constrained nonlinear multi-objective mathematical program with reliability, cost, and non-concurrence of maintenance periods and maintenance start time into the objective functions and use a genetic algorithm to optimize the model.

Taboada *et al.* (2008) present a recent study in this area. They develop a multi-objective genetic algorithm in order to solve multi-state reliability design problems. The authors utilize the universal moment generating function to measure the reliability and availability criteria in the system. They apply their approach into two examples; the first one is a system of five units connected in series in which each component has two states, functioning properly, or failure and the second one is a system of three units connected in series. In this system, each component has multiple states with different levels of performance, which range from maximum capacity to total failure. They utilize MATLAB as the programming environment, and show the effectiveness of their approach in terms of computational times and obtained non-inferior solutions.

2.3. Simulation Models

2.3.1. Monte Carlo Simulation

Bottazi *et al.* (1992) present the results of a systematic collection of actual failure times and preventive and corrective maintenance activities of 900 buses over a period of five years. They create an updatable database to estimate the failure distribution functions and to evaluate the influence of systematic preventive and corrective maintenance actions. They consider the total cost and availability as the objective functions and apply a Monte Carlo simulation approach to evaluate the model. They compare different maintenance policies and present computational results of their model.

Billinton and Pan (2000) also develop a simulation model, which is based on the Monte Carlo simulation approach, to determine the total failure frequency and the optimum maintenance interval for a parallel-redundant system. The authors present a modified distribution function and assume an exponential distribution function for component useful life and a Weibull distribution function for the wear out period. The procedure includes construction of a mathematical model and definition of the stopping rule in simulation for a parallel-redundant system. They state that if the shape parameter of the Weibull distribution function increases, the optimum maintenance interval decreases. Finally, they show that a two-component parallel-redundant system has a structure, which can be considered for minimal cut set analysis that is used for evaluation of power systems reliability.

Zhou *et al.* (2005) present an approach for sequential preventive maintenance scheduling based on the concept of age reduction due to imperfect maintenance actions. They consider an assumption for the time of imperfect maintenance actions based on required reliability of the system. They utilize a hybrid recursive method based on an assumed constant improvement factor and increasing failure rate and develop an optimization model with a maintenance cost rate in the life cycle of the system as the objective function. Finally, they apply Monte Carlo simulation and describe how their computational results can be used in decision support systems of maintenance scheduling problems.

Marquez *et al.* (2006) develop a simulation model to find the best preventive maintenance strategy in semiconductor manufacturing plants. The authors model the effective age of equipment, availability of equipment, maintenance activity backlog, and preventive maintenance policies and consider different wafer production scenarios in a Monte Carlo continuous time simulation model. They analyze and compare different maintenance strategies on the status of manufacturing equipments and operating conditions of the wafer production flow. Furthermore, they describe how the combination of the effective age concept with availability-based models increases the throughput and provides better results than the simple age-based models.

2.3.2. Discrete-Event and Continuous Simulation

Goel *et al.* (1973) present a simulation model and develop a statistical analysis that considers three different types of preventive maintenance activities by defining stochastic and deterministic decision variables as well as unavailability and cost as the main objectives. In addition, they make a 2-level sequential fractional factorial design in order to facilitate their simulation model. By designing the simulation model based on experimental design approach, their model finds the best set of preventive maintenance schedules for ground electronics systems.

Burton *et al.* (1989) develop a simulation model to evaluate the performance of a job shop. In this research, the effectiveness of the preventive maintenance scheduling under different conditions such as shop load, job sequencing rule, maintenance capacity, and strategy is determined and presented. Krishnan (1992) develops a simulation model to evaluate maintenance schedules for an automated production line in a steel rolling mill plant. He considers three different maintenance policies as opportunistic, failure, and block with the percent of availability as the objective function. He shows that the existing maintenance policy only includes the failure and block maintenance actions. By using the historical data of maintenance activities in the simulation model, an optimal preventive maintenance schedule is obtained in the form of a checklist.

Mathew and Rajendran (1993) present a simulation model in order to determine the frequency of the shutdown for periodic system overhaul, preventive and corrective maintenance, and inspections in a sugar manufacturing plant. They utilize a time-dependent simulation model to minimize the total cost including maintenance costs and downtime losses. Paz *et al.* (1994) develop a two-stage knowledge base for a maintenance supervisor assistant system. This knowledge base interacts with maintenance managers on a periodic basis to select the proper preventive maintenance plan for the next period. The first stage deals with an object-oriented computer simulation model to monitor different preventive maintenance schedules that include preventive maintenance polices, staffing policies, downtime costs, simultaneous downtime practices, travel time impacts, and blocking situations as the systems specifications. In addition, they consider overall machine availability, critical machine availability, worker utilization, cost of maintenance activities, and work order completion time as the systems criteria. At the second stage, they make a knowledge engineering environment to use the computational results obtained from a simulation model and send feedback to the first stage.

Joe *et al.* (1997) develops a simulation model in order to evaluate different preventive maintenance strategies for a fleet of vehicles in the St. Louis metropolitan police department. He utilizes GPSS as the simulation software, analyzes several strategies to improve the effectiveness and efficiency of operations, and presents the best policy. Savsar (1997) develops a simulation model in order to investigate effect of different preventive maintenance strategies in a just-in-time production system. He constructs a simulation model of a 5-station production system and considers throughput rate, average equipment utilizations, and total work-in-process as the performance measures of the production system. After running the simulation model and analyzing the computational results, he mentions that preventive and corrective maintenance policies have a high impact on the performance measures in just-in-time production systems and by combining the maintenance activities and just-in-time operations one can improve the effectiveness of the this kind of systems.

Mohamed-Salah *et al.* (1999) develop a simulation model in order to achieve opportunistic maintenance strategies in a multi-component production line. The authors consider two different strategies and define total cost as the function of preventive and corrective maintenance activities as well as fixed cost due to any stop or failure in production line. The first strategy assumes that the maintenance activities are allowed on all non-failed components if the difference between the expected preventive time of non-failed components and the failure instant of failed components is less than certain value. The second one considers that the maintenance activities are allowed on all non-failed components if the difference between the expected preventive time of non-failed components and the preventive time or corrective instant of failed components is less than certain value. They utilize PROMODEL and describe that the cost function has a unique optimum. Finally, they express that the optimal interval of maintenance for the different strategies is 5.5 and 3.5 days respectively.

Cassady *et al.* (1999) develop an integrated production control chart and preventive maintenance scheduling model to reduce the total operating cost of manufacturing systems. The researchers formulate an economic model that includes product inspection costs, process downtime costs and poor quality costs and analyze it via a simulation model. In addition, they construct a simulation-optimization model in order to evaluate and optimize the parameters of control chart and preventive maintenance strategy. They demonstrate their approach in a numerical example and show the feasibility and effectiveness of their methodology.

Greasley (2000) presents a simulation model to find an optimal maintenance planning in train maintenance depot for an underground transportation facility in the United Kingdom. He develops a simulation model based on two different situations. The first situation assumes there is no random arrival and the second one considers random arrivals and investigates the effect of the arrival on service level performance measures. He utilizes ARENA as the simulation software and shows the effectiveness of the maintenance policies obtained by the simulation model. Chan (2001) presents a simulation model to analyze the effects of preventive maintenance policies on buffer size, inventory sorting rules, and process interruptions in a flow line of a push production system. He presents the performance of the production system under different operational conditions and preventive maintenance policies.

Duffuaa *et al.* (2001) present a generic conceptual simulation model for maintenance scheduling systems. They define this simulation model by constructing seven modules including an input module, maintenance load module, planning and scheduling module, materials and spares module, tools and equipment module, quality module, and finally, a performance measure module. The authors mention that this model could be used to develop a discrete event simulation model using commercial simulation software. In addition, they suggest that by applying this model one can evaluate the need for contract maintenance and effect of availability of spare parts on performance measures in the system.

Devulapalli *et al.* (2002) develop a simulation model in order to determine the best preventive maintenance policies for bridge management systems. They utilize STROBOSCOPE software and examine conditions of bridges under different strategies. They apply their model to a set of bridges in Virginia and argue that the model can be used to provide various maintenance policies for bridge management systems. Alfares (2002) presents a simulation model to evaluate preventive maintenance schedules of components in a detergent-packing line and considers two different situations in his model. The first situation assumes a constant time interval that is not affected by maintenance actions or unexpected failures. In the second situation, the time interval is affected and restarted by maintenance actions or unexpected failures. In order to minimize the total cost, he develops a simulation model to determine the best maintenance schedule of components for each situation.

Houshyar *et al.* (2003) present a simulation model to evaluate the impact of preventive maintenance scheduling on the production rate of a manufacturing machine. They utilize PROMODEL software to develop a simulation model and consider two different scenarios for the simulation run. They use statistical analysis on the simulation outputs in order to determine the impact of recommended annual preventive maintenance schedule on the production throughput of the machine. Finally, they mention that the preventive maintenance policy does not affect the production rate but can reduce annual maintenance costs of the system.

Sawhney *et al.* (2004) present a simulation model to determine maintenance strategies of a manufacturing system. Their model is constructed to integrate reactive and proactive maintenance schedules in order to increase productivity of operations in the lean manufacturing structure. Preventive maintenance optimization is also used in semiconductor manufacturing. Rezg *et al.* (2004) present an integrated preventive maintenance and inventory control simulation model in a multi-component production line. The authors define preventive and corrective maintenance activities along with inventory control variables and parameters to develop approximate analytical models for the single machine under different scenarios. In addition, they utilize PROMODEL software to construct an age-based simulation model and apply a genetic algorithm to optimize the variables of the simulation model and evaluate different production scenarios. Finally, they test their methodology on three numerical examples of a production line and compare the computational results with results obtained from analytical approaches. They mention that applying combination of maintenance scheduling production planning policies leads to a significant reduction of the total cost of the system.

Han *et al.* (2004) develop a finite time horizon model to achieve preventive maintenance scheduling of manufacturing equipment based on setback based residual factors and use simulation approach to evaluate the model. They mention the consistency of computational results and show that simulation approach is a useful and effective method to solve such models. Rezg *et al.* (2005) present another paper in this area. He and his colleagues develop an integrated age-based preventive maintenance and inventory control simulation model in a manufacturing system with just-in-time configuration. They present two approaches; the first one is a mathematical model to determine the average cost per unit time and the second one is a combination of simulation model and experimental design methods. They use MAPLE to solve the analytical model, utilize PROMODEL for simulation, and use STATGRAPHICS to analyze the data for experimental design and regression analysis. The authors mention that both approaches could give approximately the same results. The existing differences are attributed to approximation assumptions considered in the analytical model that was eliminated in the simulation model.

Hagmark and Virtanen (2007) present one of the most recent studies on application of simulation in preventive maintenance scheduling problems. They develop a simulation model to determine the level of reliability, availability and corrective and preventive maintenance at the early stage of design. Their method considers repair time delays and effect of preventive maintenance on the system failure observed by condition monitoring and diagnostic resources.

Yin *et al.* (2007) recently propose a simulation model in order to analyze dynamic structure of maintenance scheduling in complex systems. The researchers consider various subsystems such as preventive maintenance subsystem, defects subsystem, condition-based subsystem, failure subsystem, corrective maintenance subsystem, and performance subsystem and utilized SIMULINK environment to build up the model. They analyze the structure of components and the relation of their constraints in a maintenance system and present the advantages of the model over classical stochastic process methods in a numerical example. In addition, they mention that obtained simulation results express the dynamic nature of maintenance systems.

Li and Zuo (2007) recently develop a simulation model to determine and evaluate the impact of preventive and corrective maintenance activities on the total cost of inventories in a production system. They apply a simulation approach as the solution methodology to find the optimal number of failures and the optimal level of safety stock simultaneously and mention that combining preventive and corrective maintenance scheduling with production planning can reduce the large amount of total operating cost in the system.

2.4. Age Reduction and Improvement Factor Models

Nakagawa (1988) presents notable research for models that utilize an improvement factor. His work has been referenced by many researchers. He develops two analytical models in order to find an optimal preventive maintenance schedule based on an assumption of increasing failure rate over time. The first model, called a preventive maintenance hazard rate model, calculates the average failure cost of minimal repairs along with costs of preventive maintenance and replacement actions under the assumption that preventive maintenance actions reduce the next effective age to zero. He also assumes the failure rate is increased by increasing the frequency of preventive maintenance actions. Furthermore, this model assumes that maintenance activities take place at fixed intervals between each predetermined replacement. The second model, called an age reduction preventive maintenance model, considers the average failure cost of minimal repairs as well as costs of preventive maintenance and replacement actions by assuming that the effective age of component is reduced by an improvement factor after performing minimal repairs. In order to find an optimal schedule, both models are optimized by calculus methods. He applies the models in a numerical example and describes that based on obtained computational results the second model is more practical than the first.

Jayabalan and Chaudhuri (1992) propose another often-referenced work on age reduction and improvement factors models. They develop an optimization model and a branching algorithm that minimizes the total cost of preventive maintenance and replacement activities. They assume a constant improvement factor and define a required failure rate. In addition, they assume a zero failure cost and do not consider the time value of money for future costs. Their algorithm determines an optimal schedule of maintenance actions before each replacement action in order to minimize the total cost in a finite planning horizon. They utilize FORTRAN programming environment to implement the algorithm and prove its effectiveness via several numerical examples.

Dedopoulos and Smeers (1998) develop a nonlinear optimization model to find the best preventive maintenance schedule by considering the degree of age reduction as the variable in the model. The researchers assume a constant improvement factor but a variable amount of age reduction, which depends on the schedule of preventive maintenance actions. They define the amount of age reduction, time and duration of preventive maintenance activities as the decision variables and consider fixed and variable costs for maintenance actions. They present the variable cost as a function of the amount of age reduction and duration of action and the effective age of the component. Moreover, they present the failure rate in each period as a recursive function of age reduction from a previous period and consider the net profit as the objective function in the model. They implement the model in GAMS programming environment and use GAMS/MINOS optimization software. Finally, the effectiveness of the model is shown via three numerical examples.

Martorell *et al.* (1999) present an age-dependent preventive maintenance model based on the surveillance parameters, improvement factor, and environmental and operational conditions of the equipment in a nuclear power plant. They consider risk and cost as the main criteria of the model based on the age of the system, and perform a sensitivity analysis to show the effect of the parameters on the preventive maintenance policies. They discuss how the results obtained from their model are different than those from other models that do not consider the improvement factor parameter and working conditions.

Lin *et al.* (2001) combine the models developed by Nakagawa (1988) and present hybrid models in which effects of each preventive maintenance action are considered in two ways; one for its immediate effects and the other one for the lasting effects when the equipment is put to use again. The authors construct two models that reflect the concept of maintainable and non-maintainable failure modes. In the first model, they assume that preventive maintenance and replacement time are independent decision variables and consider the mean cost rate as the objective function that should be minimized. In the second model, they assume that preventive maintenance activities are performed whenever the failure rate of the system exceeds the certain level and same as the first model, the mean cost rate is considered as the objective function. Finally, they present numerical examples to show the application of the developed models and mention that for a system with Weibull failure rate optimal schedules can be achieved analytically, but for the general case, it cannot be solved by analytic methods.

Cheng and Chen (2003) consider the improvement factor as a variable of total number of preventive maintenance actions performed over the planning horizon, and the cost ratio of preventive maintenance to replacement actions. They assume different types of restoration effects based on the cost ratio of maintenance and replacement actions and propose three different models. They consider total number of preventive maintenance actions as the decision variable and develop an objective function to minimize the total cost of the system. By using a numerical analysis method, they mention that the proposed improvement factor model provides a variety of options to evaluate the restoration effect of a deteriorating system.

Xi *et al.* (2005) develop a sequential preventive maintenance optimization model over a finite planning horizon. They define a recursive hybrid failure rate based on the improvement factor concept and increasing failure rate in order to estimate the systems reliability in each period of the planning horizon. In addition, they consider the total cost of preventive maintenance activities and assume that the mean cost in each period is a function of required reliability and the improvement factor parameter. Finally, they utilize a simulation approach to optimize the model and mention that the computational results can be used in a maintenance decision support system for job shop scheduling problems.

Jaturonnatee *et al.* (2006) develop an analytical model in order to find an optimal preventive maintenance schedule of leased equipment by minimizing the total cost function. They define maintenance actions as preventive and corrective, each with associated costs, and then consider the concept of reduction in failure intensity function along with penalty costs due to violation of leased contact issues. They present a numerical example for a system with Weibull failure rate, solve the model analytically, and examine the effect of penalty terms on the optimal preventive maintenance policies.

Bartholomew-Biggs *et al.* (2006) present several preventive maintenance scheduling models that consider the effect of imperfect maintenance on effective age of component. The researchers develop optimization models that minimize the total cost of preventive maintenance and replacement activities. In this study, they assume a known failure rate to express the expected failures as a function of age and consider age reduction in the effective age, based on the concept of an improvement factor. They develop a new mathematical programming formulation to achieve optimal maintenance schedules and utilize automatic differentiation as numerical approach, instead of analytical approach, to compute the gradients and hessians in the optimization procedure, which is a global minimization of non-smooth performance function. Finally, the effectiveness of the proposed model and algorithm is shown in several numerical examples.

One of the recent works on methods for estimating age reduction factor is presented by Che-Hua (2007). In this research, he determines an optimal preventive maintenance plan for a deteriorating single-component system via minimizing the expected cost over a finite planning horizon. He develops a mathematical model for estimating improvement factor to measure the restoration of component under the minimal repair. The proposed improvement factor is a function of effective age of component, the number of preventive maintenance actions, and the cost ratio of maintenance action to the replacement action. Finally, the researcher could obtain an optimal preventive maintenance schedule for a case study with the Weibull hazard function by applying a particle swarm optimization method.

Cheng *et al.* (2007) present a paper about models to estimate the degradation rate of the age reduction factor. They present two optimization models, which minimize the cost subject to required reliability. The first model has a periodic preventive maintenance time interval for every replacement and the second one contains the maintenance schedule where the time interval between the final maintenance and replacement is not constant.

Lim and Park (2007) present three analytical preventive maintenance models that consider the expected cost rate per unit time as the objective function. In this research, they assume that each preventive maintenance activity reduces the starting effective age but does not change the failure rate. They consider the improvement factor as the function of number of preventive maintenance activities. They also assume that the failure function corresponds to a Weibull distribution function and develop a mathematical formulation for three different situations; preventive maintenance period is known, number of preventive maintenance is known, and number and period of preventive maintenance is unknown. They obtain an optimal preventive maintenance and replacement schedule by taking an analytical approach and apply them to a numerical example to show an application of their models.

2.5. Chapter Summary

In this chapter, recent work pertaining to methods and applications of preventive maintenance and replacement scheduling were reviewed. They were categorized as optimization models, simulation models, and age reduction and improvement factor models. Table 2.1 shows the summary of the reviewed articles.

Table 2.1. Summary of reviewed articles

Author(s)	Year	Objective(s)	Method(s)/Algorithm(s)	Application(s)	Section
Canfield	1986	Min total maintenance cost	Analytical method	General system	2.2.1
McClymonds and Winge	1987	Max availability and reliability	Analytical method	Nuclear power plants	2.2.1
Martin	1988	Min total cost and Max reliability	Analytical method	Health-care	2.2.1
Hsu	1991	Min total maintenance cost	Analytical method	Serial production system	2.2.1
Jayabalan and Chaudhuri	1992	Min total maintenance cost	Analytical method	Bus engines in a public transit network	2.2.1
Westman and Hanson	2000	Determine optimal mean time to failure	Analytical method	Multi-stage manufacturing system	2.2.1
Fard and Nukala	2004	Min total maintenance cost	Analytical method	Service systems	2.2.1

Author(s)	Year	Objective(s)	Method(s)/Algorithm(s)	Application(s)	Section
Ying et al.	2005	Min total tardiness of jobs	Analytical method	Production scheduling	2.2.1
Pongpech et al.	2006	Min total maintenance and penalty costs	Analytical method	Maintenance strategies for used equipment under lease	2.2.1
Panagiotidou and Tagaras	2007	Min total maintenance cost	Karush-Kuhn-Tucker (KKT) method	Manufacturing process	2.2.1
Shirmohammadi et al.	2007	Min total maintenance cost	Analytical method	Single-component system	2.2.1
Westman et al.	2001	Min total cost of production and maintenance scheduling	Stochastic dynamic programming	Multi-stage manufacturing system	2.2.2
Yao et al.	2001	Max profit of cluster tools production	Mixed-integer linear programming	Semiconductor manufacturing	2.2.2
Yao et al.	2004	Max profits from tool availability	Mixed-integer linear programming	Semiconductor manufacturing	2.2.2
Han et al	2004	Min total cost of maintenance and replacement	Nonlinear programming	Production machine	2.2.2
Jayakumar and Asgarpoor	2004	Max availability	Linear programming and Markov decision processes	General system	2.2.2
Zhao et al.	2005	Max power plant performance and reliability	A sequential approach	Gas turbine power plant	2.2.2
Canto	2006	Min total maintenance, start-up, and production cost	Mixed-integer linear programming model by Benders' decomposition	Power plant	2.2.2
Budai et al	2006	Min total possession, maintenance and a penalty costs	Mixed-Integer linear programming	Railway Industry	2.2.2
Robelin and Madanat	2006	Max facility level	Markov chain and dynamic programming	Bridge maintenance	2.2.2
Tam et al.	2006	Min total maintenance cost/Max reliability	Nonlinear programming by generalized reduced gradient	General multi-component system	2.2.2
Alardhi et al.	2007	Max availability	Binary integer linear programming	Co-generation plants	2.2.2
Kuo and Chang	2007	Min total tardiness of jobs	Dynamic programming	Prodcution machine	2.2.2
Usher et al.	1998	Min total maintenance cost	Genetic algorithm	Single-component system	2.2.3
Levitin and Lisnianski	2000	Min total maintenance cost	Genetic algorithm	General multi-state multi-component	2.2.3
Levitin and Lisnianski	2000	Min total maintenance cost	Genetic algorithm	General multi-state series-parallel systems	2.2.3
Wang and Handschin	2000	Min maintenance time interval	Genetic algorithm	Power systems	2.2.3
Tsai et al.	2001	Min total maintenance cost	Genetic algorithm	Mechatronic system	2.2.3

Author(s)	Year	Objective(s)	Method(s)/Algorithm(s)	Application(s)	Section
Cavory et al.	2001	Max total throughput of the line	Genetic algorithm	Production line of car engines	2.2.3
Leou	2003	Min total maintenance costs	Genetic algorithm	Series of electric generators	2.2.3
Han et al.	2003	Min total production and maintenance costs	Genetic algorithm	Decision support systems for maintenance and job-shop scheduling	2.2.3
Bris et al.	2003	Min total maintenance cost	Genetic algorithm	General series-parallel systems	2.2.3
Adzakpa et al.	2004	Min total maintenance cost	Heuristic algorithm	Distributed system	2.2.3
Li and Qian	2005	Min system standpoints	Heuristic algorithm	Semiconductor manufacturing	2.2.3
Samrout et al.	2005	Min total maintenance cost	Ant colony algorithm	General series-parallel systems	2.2.3
Sortrakul et al.	2005	Min total weighted expected job completion time	Genetic algorithm	Integrated preventive maintenance scheduling and production planning in a single machine	2.2.3
Cassady and Kutanoglu	2005	Min total weighted expected completion time	Heuristic algorithm	Integrated preventive maintenance scheduling and production planning in a single machine	2.2.3
El-Ferik and Ben-Daya	2006	Min total maintenance cost	Heuristic algorithm	General system	2.2.3
Duarte et al.	2006	Min total maintenance cost	Heuristic algorithm	General series system of components	2.2.3
Limbourg and Kochs	2006	Evaluate effect of different methods	Several evolutionary algorithms	Representation of the schedule to evolutionary algorithms	2.2.3
Lapa et al.	2006	Min total maintenance cost	Genetic algorithm	high-pressure injection system	2.2.3
Shum and Gong	2007	Min total maintenance costs	Genetic algorithm	Production machine	2.2.3
Zhou et al.	2007	Max availability	Heuristic algorithm	Integrated preventive maintenance scheduling and production planning	2.2.3
Kim et al.	1994	Min total operations and maintenance costs	Genetic algorithm with simulated annealing	Thermal system	2.2.4
Tan and Kramer	1997	Min total maintenance cost	Monte Carlo simulation with a genetic algorithm	Chemical process operations	2.2.4
Marseguerra et al.	2002	Max profit and max availability	Monte Carlo simulation with a genetic algorithm	load-sharing components	2.2.4
Charles et al.	2003	Min total maintenance cost	Simulation-optimization	Production system	2.2.4
Shalaby et al.	2004	Min total maintenance cost	Genetic algorithm with simulation	General multi-component and multi-state systems	2.2.4
Allaoui and Artiba	2004	Min total tardiness of jobs	Simulated annealing with simulation	Flow shop scheduling	2.2.4

Author(s)	Year	Objective(s)	Method(s)/Algorithm(s)	Application(s)	Section
Suresh and Kumarappan	2006	Max customer satisfaction	Genetic algorithm with simulated annealing	Power system	2.2.4
Samrout et al.	2006	Min total maintenance cost	Ant colony algorithm and genetic algorithm	Large-scale system	2.2.4
Jin et al.	2006	Min total maintenance cost	Simulation-optimization	Multi-component production process	2.2.4
Ruiz et al.	2007	Min total manufacturing time	Ant colony algorithm and genetic algorithm	Integrated preventive maintenance scheduling and production planning	2.2.4
Kralj and Petrovic	1995	Min fuel cost, Max reliability, Min technological concerns	Multi-objective branch-and-bound algorithm	Thermal generating systems	2.2.5
Chareonsuk et al.	1997	Min total maintenance cost, Max reliability	Preference ranking organization method for enrichment evaluations	General system	2.2.5
Leng et al.	2006	Min total weighted expected completion time	Chaotic particle swarm optimization algorithm	Integrated preventive maintenance scheduling and production planning	2.2.5
Konak et al.	2006	Review paper	Multi-objective genetic algorithm	General reliability optimization problems	2.2.5
Quan et al.	2007	Min workforce idle time and Min maintenance time	Multi-objective genetic algorithm	General system	2.2.5
Verma and Ramesh	2007	Min total maintenance cost, Max reliability	Multi-objective genetic algorithm	Large engineering plant	2.2.5
Taboada et al.	2008	Max reliability, Max availability	Multi-objective genetic algorithm	Multi-state reliability design	2.2.5
Bottazi et al.	1992	Min total cost, Max availability	Monte Carlo simulation	Public Transit	2.3.1
Billinton and Pan	2000	Optimize maintenance intervals	Monte Carlo simulation	Power systems	2.3.1
Zhou et al.	2005	Min total maintenance cost	Monte Carlo simulation	Decision support systems for general systems	2.3.1
Marquez et al.	2006	Max throughput	Monte Carlo simulation	Semiconductor manufacturing	2.3.1
Goel et al.	1973	Min unavailability and logistics support costs	Experimental design	Electronics systems	2.3.2
Burton et al.	1989	Evaluate the performance of a job shop	Simulation	Job-shop Scheduling	2.3.2
Krishnan	1992	Max availability	Simulation	Automated production line in a steel rolling mill plant	2.3.2
Mathew and Rajendran	1993	Min total maintenance and downtime costs	Simulation	Sugar manufacturing plant	2.3.2
Paz et al.	1994	Min total maintenance cost	Simulation	Production Line	2.3.2

Author(s)	Year	Objective(s)	Method(s)/Algorithm(s)	Application(s)	Section
Joe et al.	1997	Max effectiveness and efficiency of facility operations	Simulation	Vehicle maintenance of St. Louis metropolitan police department	2.3.2
Savsar	1997	Evaluate effect of maintenance strategies	Simulation	Just-in-time production system	2.3.2
Mohamed-Salah et al.	1999	Min total maintenance cost	Simulation	Multi-component production line	2.3.2
Cassady et al.	1999	Min total operating cost of manufacturing systems	Simulation-Optimization	Integrated preventive maintenance scheduling and production planning	2.3.2
Greasley	2000	Max service level performance measures	Simulation	Train maintenance	2.3.2
Chan	2001	Evaluate effect of maintenance strategies	Simulation	Production system	2.3.2
Duffuaa et al.	2001	Evaluate effect of maintenance strategies	Generic conceptual simulation model	General system	2.3.2
Devulapalli et al.	2002	Evaluate effect of maintenance strategies	Simulation	Bridge management systems	2.3.2
Alfares	2002	Min total maintenance cost	Simulation	Detergent-packing production line	2.3.2
Houshyar et al.	2003	Evaluate effect of maintenance strategies	Simulation	Production rate of a manufacturing machine	2.3.2
Sawhney et al.	2004	Min total operations and maintenance costs	Simulation	Semiconductor manufacturing	2.3.2
Rezg et al.	2004	Evaluate effect of maintenance strategies	Simulation-optimization	Integrated preventive maintenance scheduling and inventory control	2.3.2
Han et al.	2004	Evaluate effect of maintenance strategies	Simulation	Manufacturing system	2.3.2
Rezg et al.	2005	Evaluate effect of maintenance strategies	Simulation, optimization, and experimental design	Integrated preventive maintenance scheduling and inventory control in a JIT manufacturing system	2.3.2
Hagmark and Virtanen	2007	Max reliability, Max availability	Simulation	General system	2.3.2
Yin et al.	2007	Evaluate effect of maintenance strategies	Simulation	General system	2.3.2
Li and Zuo	2007	Optimize number of failures level of safety stock	Simulation	Production system	2.3.2
Nakagawa	1988	Min total maintenance cost	Analytical method	General system	2.4

Author(s)	Year	Objective(s)	Method(s)/Algorithm(s)	Application(s)	Section
Jayabalan and Chaudhuri	1992	Min total maintenance cost	Branching algorithm	General system	2.4
Dedopoulos and Smeers	1998	Max net profit	Nonlinear programming	General system	2.4
Martorell et al.	1999	Evaluate effect of maintenance strategies	Sensitivity analysis	Nuclear power plant	2.4
Lin et al.	2001	Min total maintenance cost	Analytical method	General system	2.4
Cheng and Chen	2003	Min total maintenance cost	Analytical method	General system	2.4
Xi et al.	2005	Min total maintenance cost	Simulation	Decision support system for job shop scheduling	2.4
Jaturonnatee et al.	2006	Min total maintenance cost	Analytical method	General system	2.4
Bartholomew-Biggs et al.	2006	Min total maintenance cost	Differential equations method	General system	2.4
Che-Hua	2007	Min total maintenance cost	Particle swarm optimization method	General system	2.4
Cheng et al.	2007	Min total maintenance cost	Analytical method	General system	2.4
Lim and Park	2007	Min expected cost rate per unit time	Analytical method	General system	2.4

We found that most studies focus on single-component systems or on simple and specific systems, which is not always applicable for real and general systems. These studies provide solution methodologies and sophisticated algorithms but most developed models can be applied only into specific systems such as production systems or power plant systems. We also found that there is a lack of general modeling approach in the literature that could be applied in a wide variety of systems. In addition, not much work has been done in the area of age reduction and improvement factor models and most researchers have assumed a constant improvement factor or just presented simple models. Hence, the main contribution of this research is to define a general configuration for multi-component systems, design different maintenance actions, and develop mathematical formulation to determine optimal preventive maintenance and replacement schedules. We consider the realistic dependency between components that affects maintenance and replacement decisions, and show how to develop time-based patterns of maintenance and repair actions that minimizes the total cost of those actions including the cost of unexpected failures and maximizes the overall reliability of the system. Because we use the concept of age reduction and an improvement factor in these models, we also develop a mathematical model to estimate the improvement factor for imperfect maintenance activities.

CHAPTER 3

OPTIMIZATION MODELS AND EXACT ALGORITHMS

3.1. Introduction

This chapter will present a new modeling approach to find optimal preventive maintenance and replacement schedules for multi-component systems. We construct new closed-form optimization models based on cost and reliability characteristics of the system and solve them using a standard optimization procedures. These models provide a general framework that can be applied and used in a wide variety of systems. Computational results show the feasibility of the proposed approach.

3.2. System Configuration

Consider a new repairable and maintainable system of N components, each subject to deterioration. Each component i is assumed to have an increasing rate of occurrence of failure (ROCOF), $v_i(t)$, where t denotes actual time, $(t > 0)$. In this research, we assume that component failures follow the well-known non-homogeneous Poisson process (NHPP), with the increasing rate of occurrence of failure given as:

$$v_i(t) = \lambda_i \cdot \beta_i \cdot t^{\beta_i - 1} \quad \text{for } i = 1, \ldots, N \tag{3.1}$$

where λ_i and β_i are the scale and the shape parameters of component i respectively. The non-homogeneous Poisson process is similar to the homogeneous Poisson process (HPP) with the exception that the failure rate is a function of time. For more on this well-known stochastic process see, Ascher and Feingold (1984).

We seek to establish a schedule of future maintenance and replacement actions for each component over the period [0, T]. The interval [0, T] is segmented into J discrete intervals, each of length T/J. At the end of period j, the system is either, maintained, replaced, or no action is taken. We assume that maintenance or replacement activities in period j reduce the "effective age" of the system and thus the rate of occurrence of failure. For simplicity we also assume that these activities are instantaneous, i.e., the time required to replace or maintain is negligible, relative to the size of the interval, and thus is assumed to be zero, however, we do impose a cost associated with repair or maintenance actions.

To account for the instantaneous changes in system age and system failure rate, we introduce the following notation. Let $X_{i,j}$ denote the effective age of component i at the start of period j, and $X'_{i,j}$ denotes the age of component i at the end of period j. It is clear that:

$$X'_{i,j} = X_{i,j} + \frac{T}{J} \quad \text{for } i = 1,\dots,N;\ j = 1,\dots,T \tag{3.2}$$

3.2.1. Maintenance

Consider the case where component i is maintained in period j. For simplicity, we assume that the maintenance activity occurs at the end of the period. The maintenance action effectively reduces the age of component i at the start of the next period. That is:

$$X_{i,j+1} = \alpha_i \cdot X'_{i,j} \quad \text{for } i = 1,\ldots,N;\ j = 1,\ldots,T \text{ and } (0 \le \alpha_i \le 1) \tag{3.3}$$

The term α is an "improvement factor", similar to that proposed by Malik (1979) and Jayabalan and Chaudhuri (1992). This factor allows for a variable effect of maintenance action on the aging of the system. When $\alpha = 0$, the effect of maintenance action is to return the system to a state of "good-as-new". When $\alpha = 1$, maintenance action has no effect, and the system remains in a state of "bad-as-old". We will discuss more about age reduction and improvement factor models and develop a new model in Chapter 6.

Note that the maintenance action at the end of period j results in an instantaneous drop in the rate of occurrence of failure of component i, as shown in Figure 3.1. Thus at the end of period j, the rate of occurrence of failure for component i is $v_i(X'_{i,j})$. At the start of period $j+1$ we find that the rate of occurrence of failure drops to $v_i(X'_{i,j})$.

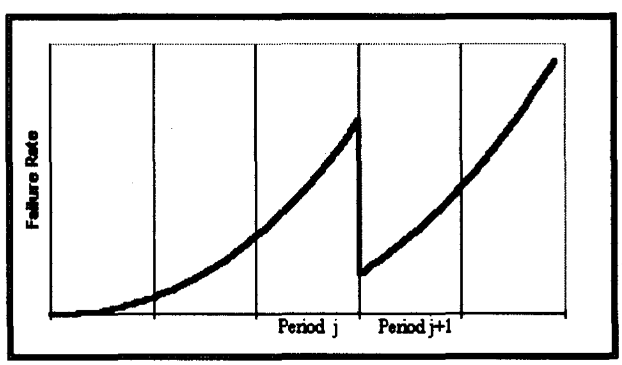

Figure 3.1. Effect of period-*j* maintenance on component ROCOF

3.2.2. Replacement

If component i is replaced at the end of period j, we find that:

$$X_{i,j+1} = 0 \quad \text{for } i = 1,\ldots,N;\ j = 1,\ldots,T \tag{3.4}$$

i.e., the system is returned to a state of "good-as-new". The rate of occurrence of failure of component i instantaneously drops from $v_i(X'_{i,j})$ to $v_i(0)$ as shown in Figure 3.2.

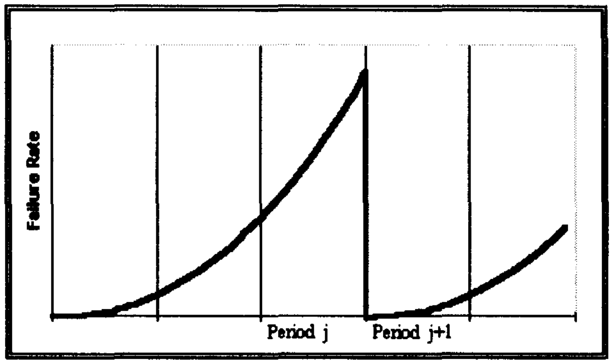

Figure 3.2. Effect of period-*j* replacement on system ROCOF

3.2.3. Do Nothing

If no action is performed in period *j*, we see no effect on the rate of occurrence of failure of component *i*, and we find that:

$$X'_{i,j} = X_{i,j} + \frac{T}{J} \quad \text{for } i = 1,\ldots,N; j = 1,\ldots,T \tag{3.5}$$

$$X_{i,j+1} = X'_{i,j} \quad \text{for } i = 1,\ldots,N; j = 1,\ldots,T \tag{3.6}$$

$$v_i(X_{i,j+1}) = v_i(X'_{i,j}) \quad \text{for } i = 1,\ldots,N; j = 1,\ldots,T \tag{3.7}$$

3.2.4. Cost of Preventive Maintenance and Replacements

For a new system, we seek to find cost associated with a given schedule of future maintenance and replacement activities. The cost associated with all component-level maintenance and replacement actions in period *j*, will be a function of the all the actions taken during that period.

3.2.4.1. Failure Cost

When we view the future periods of operation for a system, we must account for inevitable costs due to unplanned component failures. From our vantage point, at the start of period j, however, we do not know when such failures will occur. However, we know that if the system carries a high rate of occurrence of failure through a period, then we are at risk of experiencing high number, and hence, high cost of unexpected failures. Conversely, a low rate of occurrence of failure in period j should yield a low cost of failure. To account for this, we propose the computation of the expected number of failures in each period for each component in the system. (We depart here from the approach found in Usher *et al.* (1998) where an average failure rate concept was used with a cost constant.) Here we compute the expected number of failures of component i in period j, as:

$$E[N_{i,j}] = \int_{X_{i,j}}^{X'_{i,j}} v_i(t)dt \quad \text{for } i = 1,\ldots,N; j = 1,\ldots,T \tag{3.8}$$

Under the non-homogeneous Poisson process assumption, we find the expected number of component i failures in period j to be:

$$E[N_{i,j}] = \int_{X_{i,j}}^{X'_{i,j}} \lambda_i \cdot \beta_i \cdot t^{\beta_i - 1} dt = \lambda_i \left(X'_{i,j}\right)^{\beta_i} - \lambda_i \left(X_{i,j}\right)^{\beta_i} \quad \text{for } i = 1,\ldots,N; j = 1,\ldots,T \tag{3.9}$$

We assume that the cost of each failure is F_i (in units of \$/failure event), which in turn allows us to compute, $F_{i,j}$ the cost of failures attributable to component i in period j as:

$$F_{i,j} = F_i \cdot E[N_{i,j}] = F_i \cdot \lambda_i \left(\left(X'_{i,j}\right)^{\beta_i} - \left(X_{i,j}\right)^{\beta_i}\right) \quad \text{for } i = 1,\ldots,N; j = 1,\ldots,T \tag{3.10}$$

Hence regardless of any maintenance or replacement actions (which are assumed to occur at the end of the period) in period j, there is still a cost associated with the possible failures that can occur during the period.

3.2.4.2. Maintenance Cost

If maintenance is performed on component i in period j, a maintenance cost constant M_i is incurred at the end of the period.

3.2.4.3. Replacement Cost

If component i is replaced in period j we assume that the replacement cost is the initial purchase price of the component i, denoted R_i.

3.2.4.4. Fixed Cost

For a multi-component system, and the cost structure defined above, the problem can be shown to reduce to a simple problem of finding an optimal sequence of maintenance, replacement, or do-nothing actions for each component, independent of all other components. That is, one could simply find the best sequence of actions for component 1 regardless of the actions taken to component 2 and so on. This would result in N independent optimization problems. In that case, a system of N components over T time periods, has $N \times 3^T$ possible maintenance schedules.

Such a modeling approach seems unrealistic, as there should be some overall system cost penalty when an action is taken on any component in the system. It would seem that there should be some logical advantage to combining maintenance and replacement actions. For example, while the system is shut down to replace one component, it may make sense to go ahead and perform maintenance or replacement of some other components, even if they are not at their individual optimum point where maintenance or replacement would ordinarily be performed. Under this scenario, the optimal time to perform maintenance or replacement actions on individual components is dependent upon the decision made for other components. As such, we propose that a fixed cost of "downtime", Z, be charged in period j if any component (one or more) is maintained or replaced in that period. Consideration of this fixed cost makes the problem much more interesting, and more difficult to solve, as the optimal sequence of actions must be determined simultaneously for all components in the system. It can be concluded that in this situation the scheduling problem has $3^{N \times T}$ possible solutions.

3.2.4.5. Total Cost

From our vantage point at the start of the planning horizon, we wish to determine the set of activities, i.e., maintenance, replacement, or do nothing, for each component in each period such that total cost is minimized. In order to have $X'_{i,j}$, age of component i at the end of period j by using equation (3.2) first, we define $m_{i,j}$ and $r_{i,j}$ as binary variables of maintenance and replacement actions for component i in period j as:

$$m_{i,j} = \begin{cases} 1 & \text{if component } i \text{ at period } j \text{ is maintained} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

$$r_{i,j} = \begin{cases} 1 & \text{if component } i \text{ at period } j \text{ is replaced} \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

Then, we construct the following recursive function of $X_{i,j}, X'_{i,j}, m_{i,j}, r_{i,j}, \alpha_i$ with a constraint:

$$\begin{cases} X_{i,j} = (1-m_{i,j-1})(1-r_{i,j-1})X'_{i,j-1} + m_{i,j-1}(\alpha_i \cdot X'_{i,j-1}) \\ X'_{i,j} = X_{i,j} + \dfrac{T}{J} \end{cases} \tag{3.13}$$

$$m_{i,j} + r_{i,j} \leq 1 \tag{3.14}$$

In addition, we assume the initial age for each component at the start of the planning horizon is equal to zero:

$$X_{i,1} = 0 \ \text{ for } \ i = 1,\ldots,N \tag{3.15}$$

If component replacement occurs in the previous period then $r_{i,j-1} = 1,\ m_{i,j-1} = 0$, so $X_{i,j} = 0$. If a component is maintained in the previous period then $r_{i,j-1} = 0,\ m_{i,j-1} = 1$ so $X_{i,j} = \alpha_i \cdot X'_{i,j-1}$ and finally if we do nothing, $r_{i,j-1} = 0,\ m_{i,j-1} = 0$, and $X_{i,j} = X'_{i,j-1}$ which corresponds to our basic assumptions given in Section 3.1. From our definitions of each type of cost, we can derive the following total cost function as:

$$Total\ Cost = \sum_{i=1}^{N}\sum_{j=1}^{T}\left[F_i \cdot \lambda_i\left((X'_{i,j})^{\beta_i} - (X_{i,j})^{\beta_i}\right) + M_i \cdot m_{i,j} + R_i \cdot r_{i,j}\right] + \sum_{j=1}^{T}\left[Z\left(1 - \prod_{i=1}^{N}\left(1 - (m_{i,j} + r_{i,j})\right)\right)\right] \tag{3.16}$$

This objective function computes the total cost as a summation of component costs in each period based on any maintenance or replacement cost, the system "downtime" cost, and the cost of the expected number of unexpected failures. It is certainly possible to compute a more accurate economic measure of these costs, such as Net Present Value (NPV), using a suitable interest rate. One could also include the effects of inflation, by adding an inflation rate in the calculation of future costs. While these may make the model more accurate, we have avoided those minor refinements for the sake of notational simplicity.

3.3. Optimization Models

3.3.1. Model 1 - Minimizing total cost subject to reliability constraint

In this model, we attempt to minimize the total cost subject to a constraint in which some minimum level of system reliability over the planning horizon is achieved and assume that components are arranged in series. It is important to note that other system configurations (parallel, series-parallel, parallel-series, k-out-of-n, complex, etc) can be modeled just by modifying and adapting the reliability function, which reflect the configuration of the parallel, series-parallel, parallel-series, k-out-of-n, or other complex systems, but for the sake of simplicity, we consider only series systems in this research.

One may also be interested in determining the system reliability (probability of operating without failure survival over the planning horizon). Based on the assumption on a non-homogeneous Poisson process, we define the reliability of component i in period j (the probability of surviving component i to the end of period j given survival to the start of period j) as follows:

$$R_{i,j} = e^{-\left[\int_{X_{i,j}}^{X'_{i,j}} v_i(t)dt\right]} = e^{-\left[\lambda_i\left((X'_{i,j})^{\beta_i} - (X_{i,j})^{\beta_i}\right)\right]} \quad \text{for } i = 1,\dots, N;\ j = 1,\dots, T \tag{3.17}$$

Therefore, the probability of the series system of components surviving the entire planning horizon is:

$$Reliability = \prod_{i=1}^{N}\prod_{j=1}^{T} e^{-\left[\lambda_i\left((X'_{i,j})^{\beta_i} - (X_{i,j})^{\beta_i}\right)\right]} \tag{3.18}$$

Then we formulate the following nonlinear mixed-integer programming model that minimizes the total cost subject to a required reliability of the system:

$$Min\ TotalCost = \sum_{i=1}^{N}\sum_{j=1}^{T}\left[F_i \cdot \lambda_i\left(\left(X'_{i,j}\right)^{\beta_i} - \left(X_{i,j}\right)^{\beta_i}\right) + M_i \cdot m_{i,j} + R_i \cdot r_{i,j}\right] + \sum_{j=1}^{T}\left[Z\left(1 - \prod_{i=1}^{N}\left(1 - \left(m_{i,j} + r_{i,j}\right)\right)\right)\right]$$

s.t.:

$$X_{i,1} = 0 \qquad i = 1,\dots,N$$

$$X_{i,j} = (1 - m_{i,j-1})(1 - r_{i,j-1})X'_{i,j-1} + m_{i,j-1}(\alpha_i \cdot X'_{i,j-1}) \qquad i = 1,\dots,N \text{ and } j = 2,\dots,T$$

$$X'_{i,j} = X_{i,j} + \frac{T}{J} \qquad i = 1,\dots,N \text{ and } j = 1,\dots,T$$

$$m_{i,j} + r_{i,j} \le 1 \qquad i = 1,\dots,N \text{ and } j = 1,\dots,T$$

$$\prod_{i=1}^{N}\prod_{j=1}^{T} e^{-\left[\lambda_i\left(\left(X'_{i,j}\right)^{\beta_i} - \left(X_{i,j}\right)^{\beta_i}\right)\right]} \ge R_{series}$$

$$m_{i,j}, r_{i,j} = 0 \text{ or } 1 \qquad i = 1,\dots,N \text{ and } j = 1,\dots,T$$

$$X_{i,j}, X'_{i,j} \ge 0 \qquad i = 1,\dots,N \text{ and } j = 1,\dots,T$$

(3.19)

3.3.2. Model 2 - Maximizing reliability subject to budgetary constraint

Here we modify the formulation and introduce a budgetary constraint. The objective of this model is to maximize the overall reliability of the system, through our choice of maintenance and replace decisions, such that we do not exceed the budgeted total cost. This model can be formulated as:

$$Max\ \ Reliability = \prod_{i=1}^{N}\prod_{j=1}^{T} e^{-\left[\lambda_i\left(\left(X'_{i,j}\right)^{\beta_i} - \left(X_{i,j}\right)^{\beta_i}\right)\right]}$$

s.t.:

$$X_{i,1} = 0 \qquad i = 1,\dots, N$$

$$X_{i,j} = (1 - m_{i,j-1})(1 - r_{i,j-1})X'_{i,j-1} + m_{i,j-1}(\alpha_i \cdot X'_{i,j-1}) \qquad i = 1,\dots, N \text{ and } j = 2,\dots, T$$

$$X'_{i,j} = X_{i,j} + \frac{T}{J} \qquad i = 1,\dots, N \text{ and } j = 1,\dots, T$$

$$m_{i,j} + r_{i,j} \le 1 \qquad i = 1,\dots, N \text{ and } j = 1,\dots, T$$

$$\sum_{i=1}^{N}\sum_{j=1}^{T}\left[F_i \cdot \lambda_i\left(\left(X'_{i,j}\right)^{\beta_i} - \left(X_{i,j}\right)^{\beta_i}\right) + M_i \cdot m_{i,j} + R_i \cdot r_{i,j}\right] + \sum_{j=1}^{T}\left[Z\left(1 - \prod_{i=1}^{N}\left(1 - \left(m_{i,j} + r_{i,j}\right)\right)\right)\right] \le B$$

$$m_{i,j}, r_{i,j} = 0 \text{ or } 1 \qquad i = 1,\dots, N \text{ and } j = 1,\dots, T$$

$$X_{i,j}, X'_{i,j} \ge 0 \qquad i = 1,\dots, N \text{ and } j = 1,\dots, T$$

(3.20)

3.4. Solution Approach

Based on the nonlinear and mixed-integer structure of the preventive maintenance and replacement scheduling optimization models presented in Section 3.3, we apply integer programming approaches along with nonlinear optimization techniques to solve the models. We utilize both Microsoft Excel Solver[1] and LINGO[2] software to solve the nonlinear mixed-integer optimization models for each model.

The Microsoft Excel Solver tool uses the simplex method with bounds on the variables, and the branch-and-bound (BB) method for linear and integer problems and generalized reduced gradient algorithm (GRG) for nonlinear optimization.

For models with general and binary integer restrictions, LINGO includes an integer solver that works in conjunction with the linear, nonlinear, and quadratic solvers based on branch-and-bound algorithm. For linear models, the integer solver includes preprocessing and dozens of constraint "cut" generation routines that can greatly improve solution times on large classes of integer models. For nonlinear programming models, the primary underlying technique used by LINGO's optional nonlinear solver is based upon a generalized reduced gradient algorithm. However, to help get to a good feasible solution quickly; LINGO also incorporates successive linear programming. The nonlinear solver takes advantage of sparsity for improved speed and more efficient memory usage. Local search solvers are generally designed to search only until they have identified a local optimum. If the model is non-convex, other local optima may exist that yield significantly better solutions. Rather than stopping after the first local optimum is found, the global solver will search until the global optimum is confirmed. The global solver converts the original non-convex,

---

[1] http://office.microsoft.com
[2] http://www.lindo.com

nonlinear problem into several convex, linear sub-problems. Then, it uses the branch-and-bound technique to exhaustively search over these sub-problems for the global solution.

3.5. Computational Results

In order to illustrate the models numerically, and the proposed solution procedure, we develop a representative data set shown in Table 3.1. In addition, we assume Z = \$800 as the fixed cost, R = 50% as the required reliability for Model 1, B = \$15000 as the given budget for Model 2, and 36 months as the planning horizon. It is useful to mention that for the example problem, the nonlinear mixed-integer optimization models presented in section 3.3 have 1420 variables, 720 of which are binary and 1062 constraints, 352 of which are nonlinear. LINGO programs of nonlinear mixed-integer optimization models are presented in Appendix A.

Table 3.1. Parameters of the numerical example

Component	λ	β	α	Failure Cost (\$)	Maintenance Cost (\$)	Replacement Cost (\$)
1	0.00022	2.20	0.62	250	35	200
2	0.00035	2.00	0.58	240	32	210
3	0.00038	2.05	0.55	270	65	245
4	0.00034	1.90	0.50	210	42	180
5	0.00032	1.75	0.48	220	50	205
6	0.00028	2.10	0.65	280	38	235
7	0.00015	2.25	0.75	200	45	175
8	0.00012	1.80	0.68	225	30	215
9	0.00025	1.85	0.52	215	48	210
10	0.00020	2.15	0.67	255	55	250

Excel Solver is able to solve smaller problems. For example, a test problem with 2 components and 12 months took only 17 minutes on a laptop computer (Intel/Core 2, 1.67 GHz and 2 GB RAM). However, the example problem described above, with 10 components and 36 periods could not be solved in reasonable time. Using LINGO, we were able to solve the example problem for both models in approximately 4.5 hours and 1.5 hours respectively. The objective function value for the optimum solution in the Model 1 is \$13797.10 and the overall reliability of the system with this optimal solution is 50.00% equal to required reliability of the model. For the second model, the system reliability is maximized and found to be 49.92% and the total consumed budget is equal to \$14989.74. The optimal schedules for these two models are presented in Tables 3.2 and 3.3 respectively.

Table 3.2. Optimal maintenance and replacement schedule that minimizes total cost (Reliability=50.00% and Cost=\$13797.10)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	R	-	-	M	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-
2	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	M	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-
3	-	-	-	-	M	R	-	-	-	-	M	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-
4	-	-	-	-	M	M	-	-	-	-	R	-	-	-	-	-	R	-	-	M	-	-	-	M	-	-	-	-	-	R	-	-	-	-	-	-
5	-	-	-	-	-	M	-	-	-	-	M	-	-	-	-	-	R	-	-	-	-	-	-	M	-	-	-	-	-	M	-	-	-	-	-	-
6	-	-	-	-	M	M	-	-	-	-	R	-	-	-	-	-	R	-	-	M	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-
7	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-
8	-	-	-	-	-	M	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	M	-	-	-	-	-	M	-	-	-	-	-	-
9	-	-	-	-	M	-	-	-	-	-	M	-	-	-	-	-	R	-	-	M	-	-	-	M	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	M	-	-	-	-	-	R	-	-	-	-	-	-

Table 3.3. Optimal maintenance and replacement schedule that maximizes reliability (Budget=\$14989.74 and Reliability=49.92%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	M	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
2	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	M	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
3	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	M	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
4	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	M	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
5	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	M	-	-	-	R	M	-	-	-	-	-	M	-	-	-	-	-	-	-	-	-
6	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	M	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
7	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
8	-	M	-	-	-	-	-	-	-	-	M	-	-	-	-	M	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
9	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-
10	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-

Note that in both models most of maintenance and replacement actions tend to occur in the same period, which reflects the effect of the fixed cost Z. As we can see that in both models the reliability is around 50%, but the optimal total cost in the first model is 7% lower than consumed budget in the second model. It is also interesting to note that once a repair action occurs, it is often followed by a period of inactivity. Such observations can perhaps lead to the development of simple heuristic solution procedures in following on work.

Another interesting aspect of this type of modeling is that one can analyze the effective age of each component. Maintenance managers could use the model to track the effective age of the components and then utilize the information to initiate additional monitoring activities. For example, after a component reaches a certain level of effective age, additional monitoring, tests or inspections might be warranted to assist in the detection of imminent failure.

The minimum, maximum, and average effective age of each component are shown in Tables 3.4 and 3.5. Notice that the minimum effective age of each component is equal to zero at the beginning of planning horizon. Hence, minimum effective ages of components are shown from the second month on. Note that most components were replaced at some time during the planning horizon. The effective age for the components ranges from roughly 0-15 months with an average age of about 4 months.

Table 3.4. Effective age of components in Model 1

Component	Minimum Effective Age (month)	Maximum Effective Age (month)	Average Effective Age (month)
1	0.0	6.0	2.9
2	0.0	6.0	2.9
3	0.0	8.8	3.4
4	0.0	8.8	3.5
5	0.0	10.5	5.4
6	0.0	7.8	3.2
7	0.0	7.0	3.0
8	0.0	15.1	7.1
9	0.0	14.9	6.0
10	0.0	9.0	3.6

Table 3.5. Effective age of components in Model 2

Component	Minimum Effective Age (month)	Maximum Effective Age (month)	Average Effective Age (month)
1	0.0	9.0	3.5
2	0.0	9.0	3.5
3	0.0	9.0	3.5
4	0.0	9.0	3.4
5	0.0	12.1	4.5
6	0.0	9.0	3.5
7	0.0	9.0	3.5
8	0.0	12.2	5.8
9	0.0	9.0	3.5
10	0.0	9.0	3.5

Figures 3.3.1 through 3.3.10 and Figures 3.4.1 through 3.4.10 show the effective age of each component. As we can see, when a component is maintained the effective age of that component drops based on the amount of improvement factor, α_i, presented in Table 3.1. For example based on the effective age presented in Figure 3.3.1, component 1 does not receive any maintenance action for the first 4 months, but it is replaced at the 5th month, maintained at the 10th month and so on. This causes the effective age drops to zero and component 1 works as a new one at the beginning of the next month.

Another important feature presented in Figures 3.3 and 3.4 is the effect of failure rate on the number and frequency of maintenance and replacement actions of components over a planning horizon. For example, compare the variations in the effective age of components 7 and 9 in Figures 3.3.7 and 3.3.9. It can be seen that component 7 is just replaced and there is no maintenance action is performed on this component. On the other hand, component 9 is just maintained and it is replaced once at month 17. This is related to values of λ and β for each component. In Table 3.1, component 7 has 0.00015 and 2.25 and component 9 has 0.00025 and 1.85 for parameters λ and β, which means that component 7 has a higher failure rate and greater probability to fail than component 9. Therefore, it is necessary that component 7 receive more replacement actions than component 9 in order to satisfy the required reliability or to maximize the system's reliability.

Appendix B presents the detailed computational results of optimization models. Tables B.1 and B.4 show the expected number of failures and Tables B.2 and B.5 present the reliability of components over the planning horizon. We can see that expected number of failures for all components is too low and reliability of all components is higher than 99%; this is due to the optimal preventive maintenance and replacement schedule that keeps the components and the system in excellent condition.

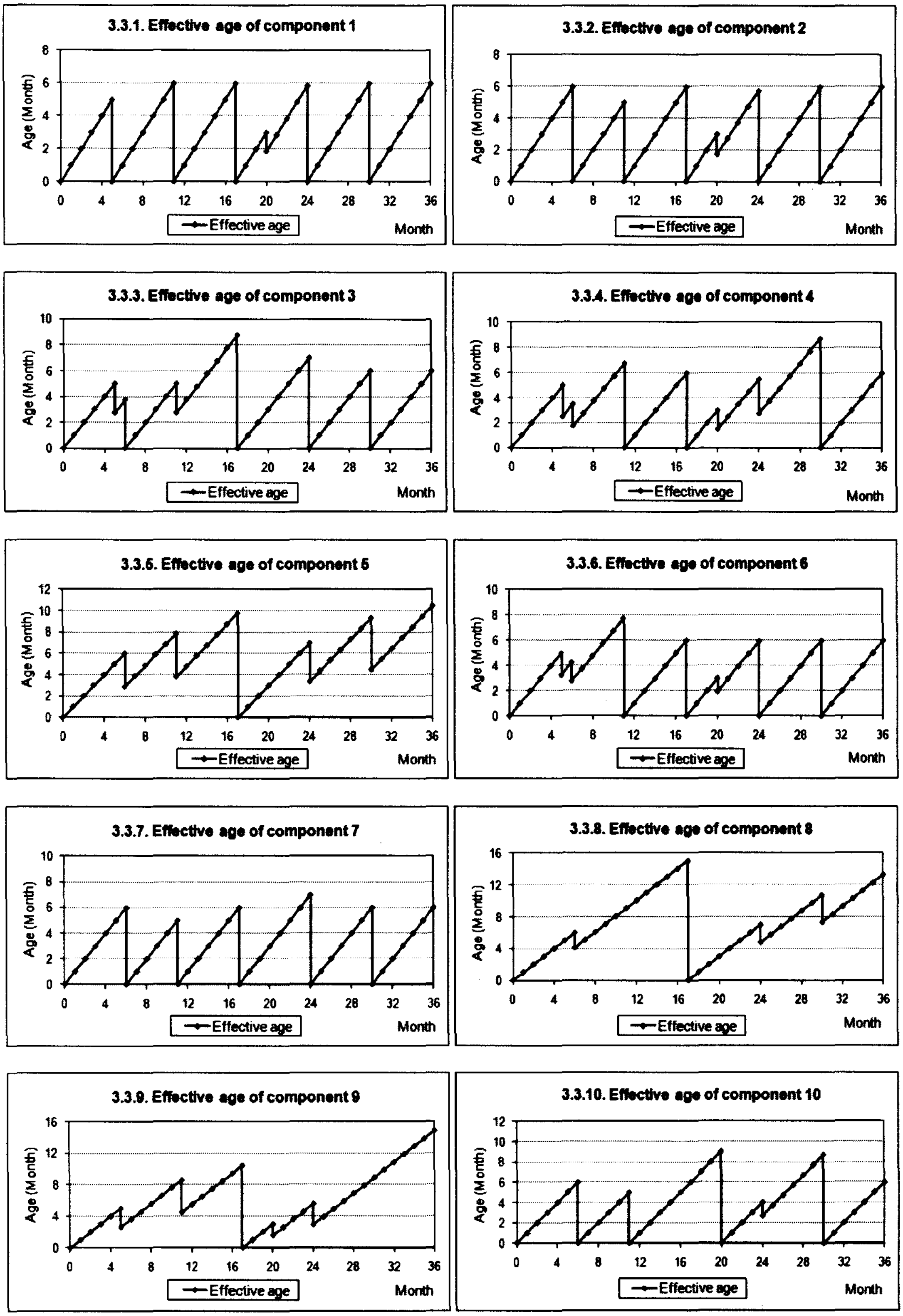

Figure 3.3. Effective age of components in Model 1

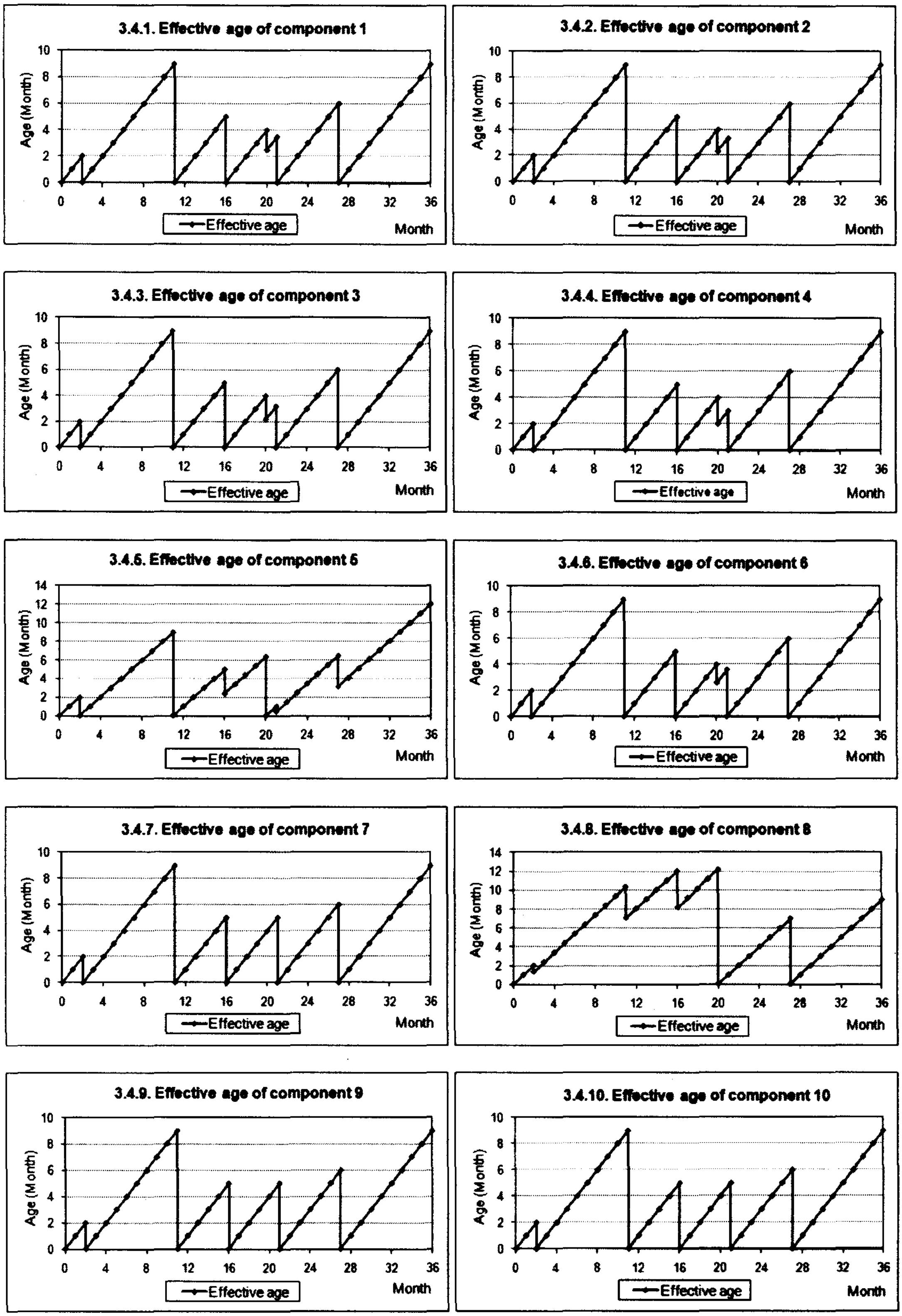

Figure 3.4. Effective age of components in Model 2

3.6. Chapter Summary

This chapter presented basic assumptions and framework for the formulation of preventive maintenance and replacement scheduling problem in order to find the best sequence of actions for each component in the system over a planning horizon such that total costs are minimized or the overall reliability of the system is maximized. Two nonlinear mixed-integer programming models were developed and optimized by generalized reduced gradient and branch-and-bound algorithms using LINGO software. The application and effectiveness of the optimization models to find the best preventive maintenance and replacement schedule in multi-component systems were presented via a numerical example. Furthermore, the computational results of both models were analyzed and advantages of the proposed approach were shown.

CHAPTER 4

OPTIMIZATION MODELS AND METAHEURISTIC ALGORITHMS

4.1. Introduction

In Chapter 3, we presented two nonlinear mixed-integer programming models that were optimized using generalized reduced gradient and branch-and-bound algorithms in LINGO software. Because of the computational complexity of nonlinear mixed-integer programming models to solve real large-scale problems, we intend to apply metaheuristic methods to tackle the problem. In this chapter, we present a new multi-objective optimization model to find an non-dominated preventive maintenance and replacement schedule of multi-component systems, which is an extension of proposed models in Chapter 3. Two types of metaheuristic algorithms are adapted and modified to solve the multi-objective optimization model. Computational results show the feasibility and effectiveness of the proposed approaches.

4.2. Engineering Economics Parameters

Based on the equations (3.9) and (3.10), we assume that the general effect of inflation increases the cost of failures over time, at a rate of *inffailure* percent per period. Thus we find, $F_{i,j}$, the cost of failures attributable to component i in period j as:

$$F_{i,j} = F_i \cdot \lambda_i \left((X'_{i,j})^{\beta_i} - (X_{i,j})^{\beta_i} \right) (1 + inffailure)^j \quad \text{for } i = 1,\dots,N;\ j = 1,\dots,T \tag{4.1}$$

In addition, we assume a separate inflation rates, $infm$, $infr$, and $infz$ for maintenance, replacement and fixed costs increases over time, and find that the associated costs of maintenance activities of component i in period j as follows:

$$M_{i,j} = M_i (1 + infm)^j \quad \text{for } i = 1,\dots,N;\ j = 1,\dots,T \tag{4.2}$$

$$R_{i,j} = R_i (1 + infr)^j \quad \text{for } i = 1,\dots,N;\ j = 1,\dots,T \tag{4.3}$$

$$Z_j = Z (1 + infz)^j \left(1 - \prod_{i=1}^{N} \left(1 - (m_{i,j} + r_{i,j}) \right) \right) \quad \text{for } i = 1,\dots,N;\ j = 1,\dots,T \tag{4.4}$$

Note that $m_{i,j}$ and $r_{i,j}$ are binary variables of maintenance and replacement actions for component i in period j and they cannot be equal to one simultaneously. The equation (4.4) mentions that if a component is maintained or replaced in each period, the defined fixed cost will be charged. From our definitions of each type of cost and by using standard time value of money concepts and an interest rate int, we can find the total net present worth (NPW) of the failure, maintenance, replacement, and fixed costs over the planning horizon with the length of T periods.

4.3. Multi-Objective Optimization Model

By considering engineering economics parameters, we can extend the objective function of the total cost that should be minimized. Finally, the multi-objective optimization model corresponds to the cost and reliability functions can be expressed as:

$$Min\ Total\ Cost = \sum_{j=1}^{T}\left(\left(\sum_{i=1}^{N}\begin{bmatrix} F_i \cdot \lambda_i\left(\left(X'_{i,j}\right)^{\beta_i} - \left(X_{i,j}\right)^{\beta_i}\right)\left(1 + inffailure\right)^j \\ + M_i\left(1 + infm\right)^j \cdot m_{i,j} + R_i\left(1 + infr\right)^j \cdot r_{i,j} \end{bmatrix}\right) + Z\left(1 + infz\right)^j\left(1 - \prod_{i=1}^{N}\left(1 - \left(m_{i,j} + r_{i,j}\right)\right)\right)\right)\left(1 + int\right)^{-j}$$

$$Max\ Reliability = \prod_{i=1}^{N}\prod_{j=1}^{T} e^{-\left[\lambda_i\left(\left(X'_{i,j}\right)^{\beta_i} - \left(X_{i,j}\right)^{\beta_i}\right)\right]}$$

$s.t.:$

$$\begin{aligned} & X_{i,1} = 0 && i = 1,\dots,N \\ & X_{i,j} = (1 - m_{i,j-1})(1 - r_{i,j-1})X'_{i,j-1} + m_{i,j-1}(\alpha_i \cdot X'_{i,j-1}) && i = 1,\dots,N \text{ and } j = 2,\dots,T \\ & X'_{i,j} = X_{i,j} + \frac{T}{J} && i = 1,\dots,N \text{ and } j = 1,\dots,T \\ & m_{i,j} + r_{i,j} \le 1 && i = 1,\dots,N \text{ and } j = 1,\dots,T \\ & m_{i,j}, r_{i,j} = 0 \text{ or } 1 && i = 1,\dots,N \text{ and } j = 1,\dots,T \\ & X_{i,j}, X'_{i,j} \ge 0 && i = 1,\dots,N \text{ and } j = 1,\dots,T \end{aligned} \tag{4.5}$$

In the above optimization model, $m_{i,j}$ and $r_{i,j}$ are binary variables of maintenance and replacement actions for component i in period j. The first set of constraints shows that the initial age for each component is equal to zero. The second set mentions that if a component is replaced in the previous period then $r_{i,j-1} = 1$, $m_{i,j-1} = 0$, so $X_{i,j} = 0$ and if a component is maintained in the previous period then $r_{i,j-1} = 0$, $m_{i,j-1} = 1$ so $X_{i,j} = \alpha_i \cdot X'_{i,j-1}$. Finally if we do nothing, $r_{i,j-1} = 0$, $m_{i,j-1} = 0$, and $X_{i,j} = X'_{i,j-1}$. The other constraints correspond to our system configuration presented in Chapter 3.

4.4. Genetic Algorithms

Genetic Algorithms (GAs) were developed and introduced by John Holland (1975). Genetic algorithm is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search metaheuristics. They are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. They have been designed as general search strategies and optimization methods working on populations of feasible solutions. Based on population search approach, genetic algorithms are able to solve multi-objective optimization problems. A generic single-objective genetic algorithm can be easily modified to search a new set of multiple non-dominated solutions. The ability of genetic algorithm to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces.

4.4.1. Representation of Solutions

The first step in any genetic algorithm implementation is to develop an encoding of the solution. In order to represent the solution of the proposed preventive maintenance and replacement scheduling problem with do nothing, maintenance and replacement actions as a chromosome used by genetic algorithms, we define an array with length of $N \times T$ for N components and T periods where each cell in that array contains 0, 1 or 2 corresponds to three different actions.

4.4.2. Fitness Functions

A fitness function is a particular type of objective function that quantifies the optimality of a solution (that is, a chromosome) in a genetic algorithm so that particular solution may be ranked against all the other solutions. An ideal fitness function correlates closely with the algorithm's goal, and yet may be computed quickly. Since the optimization model presented in (4.5) is a multi-objective optimization model, we consider three different fitness functions in order to represent the model as a single-objective optimization model and to evaluate and compare different Pareto optimal fronts (also known as "*trade off curve*").

$$Fitness_1 = w_1(Total\ Cost / Cost_{max}) + w_2(-Reliability) \tag{4.6}$$

$$Fitness_2 = (-Reliability) + (1/Cost_{max}) \times |Total\ Cost - Given\ budget| \tag{4.7}$$

$$Fitness_3 = (Total\ Cost / Cost_{max}) + |Reliability - Required\ Reliability| \tag{4.8}$$

Note that the above fitness functions are all subject to minimization. The first fitness function, $Fitness_1$, is based on the weighted summation of the normalized total cost and reliability functions with the condition of $w_1 + w_2 = 1$; for more information see Cohon (1978). The weighted summation strategy converts the multi-objective problem into a single-objective problem by constructing a weighted sum of all the objectives. In order to normalize the total cost function, we defined $1/Cost_{max}$ as the normalization coefficient. This coefficient is the maximum amount of total cost that the system could incur when all components are replaced in each period over the planning horizon. The second fitness function, $Fitness_2$, considers maximizing the reliability function and minimizing a penalty term of the total cost. The penalty term is based on violated values of the total cost of maintenance and replacement activities and the given budget in the system. Since the violated values have larger amount in comparison with reliability values, we normalize the violated values by using normalization coefficient. The third fitness function, $Fitness_3$, minimizes the total cost and absolute values of subtraction of overall reliability and required reliability of the system. As before, we considered $1/Cost_{max}$ as the normalization coefficient to normalize total cost term in order to make a same magnitude for both parts.

The idea used in the second and third fitness functions is a special case of goal programming method called goal attainment method developed by Gembicki (1974). This involves expressing a set of design goals, which is associated with a set of objectives. The problem formulation allows the objective functions to be under- or overachieved and enables the designer to be relatively imprecise about initial design goals. The relative degree of under- or overachievement of the goals is controlled by a vector of weighting coefficients, and is expressed as a standard optimization problem. The goal attainment method provides a convenient intuitive interpretation of the design problem, which is a solvable using standard optimization procedure.

4.4.3. Crossover Procedures

The crossover procedures create a new solution as the offspring of pair of selected solutions (parent solutions). The offspring should inherit some useful properties of both parents in order to facilitate its propagation throughout the population. We employed and tested several common crossover procedures, but we found that they do not work very well and generate poor solutions that result to slow and premature convergence of the genetic algorithm. Therefore, based on the especial structure of the problem we designed two new crossover procedures to overcome ineffectiveness of the tested crossover procedures as follow:

a) Two-Point Inverse Crossover: In this type crossover, first we generate two random numbers between 1 and $N \times T$, then make an offspring from selected parents in which all elements outside the position of those random numbers are copied from the first parent but in an inverse order and inside elements are copied from the second parents. If the chosen parents are identical, this type of crossover makes a different offspring, which is not the same to its parents.

b) NT-Point Crossover: In this type crossover, the even genes are copied from the first parent and odd genes are copied from the second one.

Based on the structure of the obtained solutions in genetic algorithms iterations, we designed that if two selected solutions are equal to each other, then the algorithm uses Two-Point Inverse Crossover, and if the selected solutions are not same, the algorithm uses NT-Point Crossover to produce new solutions.

4.4.4. Mutation Procedure

The mutation procedure is applied to the offspring solutions. It makes changes into the solution encoding string by modifying some of the string elements.

Based on the especial structure of the proposed preventive maintenance and replacement scheduling optimization model in which if even one maintenance or replacement action is performed in a period, the whole system encounters a fixed cost, we define a special type of mutation procedure. In this type of mutation, a random number between 1 and $N \times T$ is generated, then the corresponding gene is changed to 1 or 2 if it is equal to 0, or it is changed to 0 if it is equal to 1 or 2, and do same procedure in the same period for other components. This kind of mutation procedure produces schedules in which maintenance and replacement activities tend to occur in the same periods across all components.

4.4.5. Generational Genetic Algorithm

In the generational genetic algorithm (GGA), the entire population is replaced in each generation. The generational genetic algorithm uses two populations at the reproduction stage. One population contains the parents to be selected and the second one is generated to hold their progeny. The generational genetic algorithm is as follows, see Goldberg (1989) and Lisnianski and Levitin (2003):

Begin Generational Genetic Algorithm
  g=0
  Produce initial population P(g)
  Determine the fitness values of members in P(g)
  While GA termination condition is not satisfied, do
    g=g+1
    Select solutions from P(g-1) for P(g) based on their fitness value with the probability of $p_{selection}$ as the selected parents
    Make an offspring from selected parents from P(g-1) with the probability of $p_{crossover}$
    Mutate solutions from P(g-1) with the probability of $p_{mutation}$
    Determine the fitness values of the new generated solutions in P(g)
  End while
End Generational Genetic Algorithm

4.4.6. Steady State Genetic Algorithm

The steady state genetic algorithm (SSGA) uses the same population for both parents and their progeny. When the generic operation on the parents is completed, the new offspring takes the place of the members of the previous generation within that population. The steady state genetic algorithm is as follows, see Whitley (1989) and Lisnianski and Levitin (2003):

Begin Steady State Genetic Algorithm
  Produce initial population P
  Determine the fitness values of members in P
  While GA termination condition is not satisfied, do
    While genetic cycle termination condition is not satisfied, do

Make an offspring from selected parents
Mutate the produced offspring with the probability of $p_{mutation}$
Determine the fitness values of the new produced solution
Replace the new produced solution with the worst solution in P if its fitness value is better than the fitness value of the worst solution
Discard identical solutions in P
End while
Update P with new produced solutions
End while
End Steady State Genetic Algorithm

4.5. Implementation of the Genetic Algorithms

In order to illustrate the optimization model numerically, and the proposed solution procedure, we used data set presented in Table 3.1 and assume Z = \$800 as the fixed cost and a 36-month planning horizon. In addition, we set the genetic algorithm parameters for both generational and steady state genetic algorithms as presented in Table 4.1. Finally, we consider inflation rates for failure, maintenance, replacement, and fixed costs equal to 1%, 1.5%, 2%, and 1% respectively and 3% as an interest rate for engineering economy parameters. We utilized MATLAB R2008a[1] programming environment to develop the generational and steady state genetic algorithm as well as to calculate the fitness functions. We investigated the computational efficiency of the algorithms in terms of CPU time. The computational time is about slightly less than 6 minutes for both algorithms on a laptop computer (Intel/Core 2, 1.67 GHz and 2 GB RAM). Appendix C presents the MATLAB programs of fitness functions, crossover and mutation procedures, and generational and steady state genetic algorithms.

[1] www.mathworks.com

Table 4.1. Parameters of Genetic Algorithms

Generational GA		Steady State GA	
Number of Generations	500	Number of Generations	1
Population Size	2000	Genetic Cycle	500
Probability of Selection	0.20	Number of Iterations	100
Probability of Crossover	0.40	Population Size	2000
Probability of Mutation	0.40	Probability of Mutation	0.20

4.5.1. Computational Results of Fitness Function 1

We run both generational and steady state genetic algorithms with the first fitness function for the set of weights for both objective functions and achieve Pareto optimal solutions (also known as *"non-dominated solutions"*) shown in Table 4.2. We achieved the extreme points as $37334.28 for the total cost and 91.03% as the maximum reliability in a case of having only reliability function as the objective function in the optimization model. We also found $454.85 as the minimum total cost and 2.22% as the systems reliability in a case that system has only total cost as the objective function.

Table 4.2. Pareto optimal solutions of fitness function 1 with GAs

Weights		Generational GA		Steady State GA	
W1	W2	Cost	Reliability	Cost	Reliability
0.0	1.0	$ 37,334.28	91.03%	$ 37,334.28	91.03%
0.1	0.9	$ 37,334.28	91.03%	$ 37,229.57	90.98%
0.2	0.8	$ 33,585.74	89.89%	$ 32,586.72	90.08%
0.3	0.7	$ 28,004.50	88.63%	$ 27,426.80	88.32%
0.4	0.6	$ 20,127.67	84.43%	$ 21,414.99	85.48%
0.5	0.5	$ 14,602.70	80.23%	$ 16,697.21	81.97%
0.6	0.4	$ 10,599.07	74.85%	$ 12,694.47	77.29%
0.7	0.3	$ 9,080.44	71.71%	$ 9,638.40	72.86%
0.8	0.2	$ 6,240.55	62.93%	$ 6,979.54	65.36%
0.9	0.1	$ 3,581.16	48.79%	$ 2,602.64	39.80%
1.0	0.0	$ 454.85	2.22%	$ 454.85	2.22%

Figure 4.1 represents the Pareto optimal front of the first fitness function obtained by generational and steady state genetic algorithms. Figures 4.2 and 4.3

illustrate cost and reliability progress show the cost and reliability progress during the iterations of the algorithms for $w_1 = 80\%$ and $w_1 = 20\%$. As we can see, the convergence of the steady state genetic algorithm is somewhat faster than the convergence of the generational genetic algorithm but the quality of final solution resulting from the generational genetic algorithm is slightly better than from the steady state genetic algorithm.

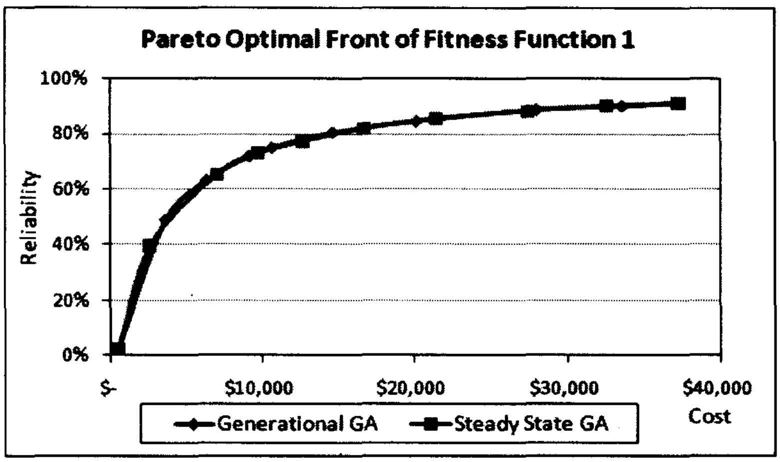

Figure 4.1. Pareto optimal front of fitness function 1 with GAs

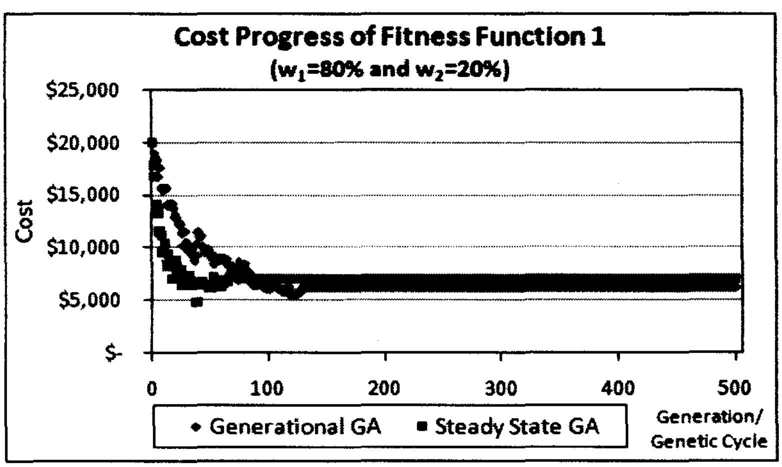

Figure 4.2. Cost progress of fitness function 1 with GAs

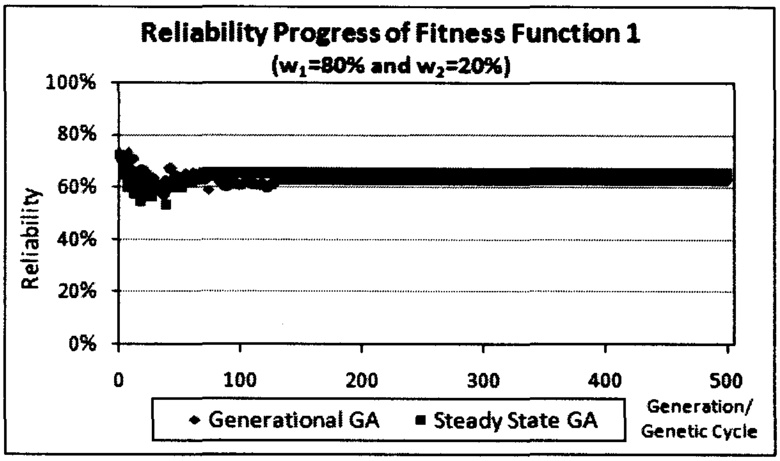

Figure 4.3. Reliability progress of fitness function 1 with GAs

Tables 4.3 and 4.4 show an example of non-dominated preventive maintenance and replacement schedules with fitness function 1 for 0.8 and 0.2 as the weights for cost and reliability objective functions. With these weights, the values of objective functions are \$6240.55 and 62.93% obtained by the generational genetic algorithm and are \$6979.54 and 65.36% achieved by the steady state genetic algorithm. It should be mentioned that all of replacement actions tend to occur in the same month, which reflects the effect of the fixed cost Z. It is also interesting to note that once a replacement action occurs, it is always followed by a period of inactivity.

Table 4.3. Non-dominated preventive maintenance and replacement schedule Fitness function 1, GGA (w_1=80% and w_2=20%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
2	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
3	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
4	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
5	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
6	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
7	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
8	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
9	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
10	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-

Table 4.4. Non-dominated preventive maintenance and replacement schedule Fitness function 1, SSGA (w_1=80% and w_2=20%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
2	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
3	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
4	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
5	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
6	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
7	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
8	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
9	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-
10	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-

4.5.2. Computational Results of Fitness Function 2

We optimize the model (4.5) with fitness function 2 and by considering different budget levels in the system and obtain Pareto optimal solutions presented in Table 4.5. Based on the extreme points, we considered different budget levels range from \$400 to \$20000 in the system for the second fitness function.

Table 4.5. Pareto optimal solutions of fitness function 2 with GAs

Given Budget	Generational GA		Steady State GA	
	Cost	Reliability	Cost	Reliability
\$ 400.00	\$ 454.85	2.22%	\$ 454.85	2.22%
\$ 2,000.00	\$ 2,000.61	14.94%	\$ 2,000.12	18.88%
\$ 4,000.00	\$ 4,000.23	42.14%	\$ 4,000.07	35.61%
\$ 6,000.00	\$ 6,000.13	58.00%	\$ 6,000.03	56.95%
\$ 8,000.00	\$ 7,999.97	64.98%	\$ 7,999.87	62.38%
\$ 10,000.00	\$ 9,999.96	69.07%	\$ 9,999.98	66.39%
\$ 12,000.00	\$ 11,998.88	75.24%	\$ 11,999.70	72.31%
\$ 14,000.00	\$ 14,000.02	77.98%	\$ 13,999.10	75.42%
\$ 16,000.00	\$ 15,999.56	80.23%	\$ 16,000.65	78.92%
\$ 18,000.00	\$ 17,999.98	83.56%	\$ 17,999.33	81.25%
\$ 20,000.00	\$ 20,000.40	85.12%	\$ 19,999.93	83.11%

Figure 4.4 shows the Pareto optimal front of fitness function 2 obtained by the genetic algorithms. As it can be seen, both Pareto fronts are relatively similar to each other. The cost and reliability progress in terms of number of generations and

genetic cycles in the generational and steady state genetic algorithms are also shown in Figures 4.5 and 4.6. It is clear that the convergence of the steady state genetic algorithm is little bit faster than the convergence of generational genetic algorithm at the beginning iterations.

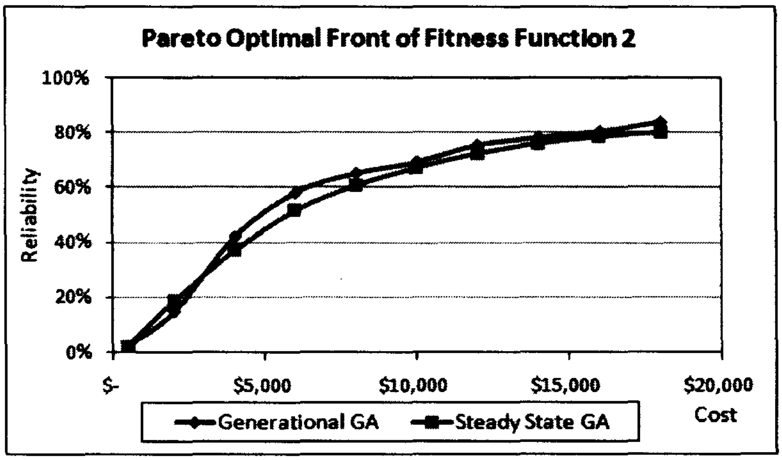

Figure 4.4. Pareto optimal front of fitness function 2 with GAs

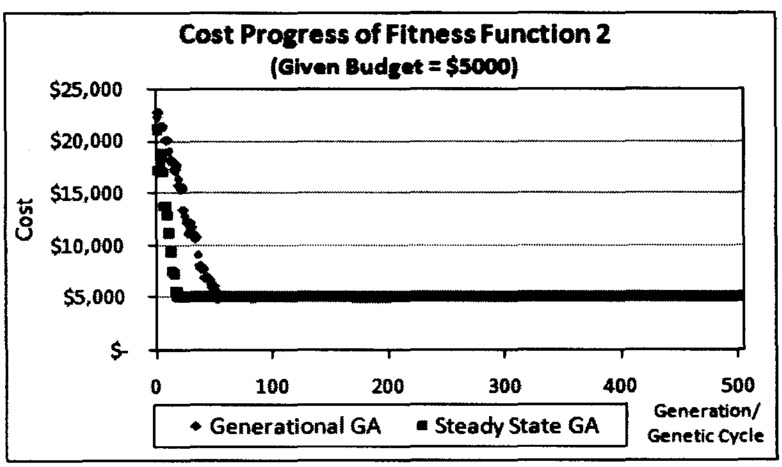

Figure 4.5. Cost progress of fitness function 2 with GAs

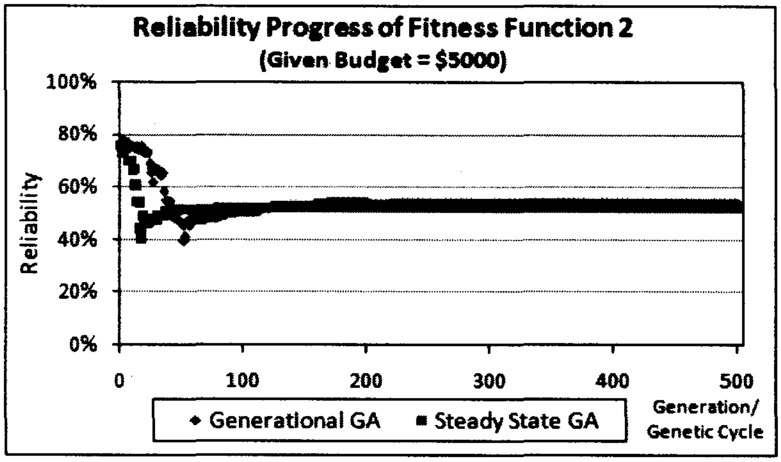

Figure 4.6. Reliability progress of fitness function 2 with GAs

Tables 4.6 and 4.7 show an example of non-dominated preventive maintenance and replacement schedules with fitness function 2 for a $5000 as given budget. With this level of budget, the reliability of the system achieved by the generational and steady state genetic algorithms is 54.07% and 51.88% respectively. As we can see, in this situation, all of maintenance and replacement actions take place in the same period which reflects the effect of fixed cost and once maintenance or replacement action occurs, it is often followed by a period of inactivity.

Table 4.6. Non-dominated preventive maintenance and replacement schedule Fitness function 2, GGA (Budget=$5000 and Reliability=54.07%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	R	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
2	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
3	-	-	-	-	R	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
4	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
5	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
6	-	-	-	-	R	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
7	-	-	-	-	R	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
8	-	-	-	-	R	-	-	M	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
9	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
10	-	-	-	-	R	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-

Table 4.7. Non-dominated preventive maintenance and replacement schedule Fitness function 2, SSGA (Budget=$5000 and Reliability=51.88%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	R	-	-	R	R	-	-	R	-	M	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	R	-	-	-	R	-	-	-	-	M	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	R	-	-	R	R	-	-	M	-	M	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	R	-	-	-	R	-	-	R	-	M	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	R	-	-	R	R	-	-	R	-	M	M	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	R	-	-	M	R	-	-	-	-	M	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	R	-	-	R	R	-	-	M	-	M	M	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	R	-	-	M	R	-	-	-	-	M	M	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	R	-	-	R	R	-	-	M	-	M	M	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	R	-	-	-	R	-	-	R	-	M	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-

4.5.3. Computational Results of Fitness Function 3

Finally, Table 4.8 presents the Pareto optimal solutions of the fitness function 3 for different required reliability range from 0 to 100%. Figure 4.7 presents the Pareto optimal front obtained by the generational and steady state genetic algorithms with fitness function 3. In this case, the Pareto fronts do not exactly coincide with each other as it happened for the first and second fitness functions. Figures 4.8 and 4.9 represent the cost and reliability progress in both genetic algorithms. In this case, the convergence of both algorithms is same but the generational genetic algorithm reduces the total cost better than steady state genetic algorithm does.

Table 4.8. Pareto optimal solutions of fitness function 3 with GAs

Required Reliability	Generational GA		Steady State GA	
	Cost	Reliability	Cost	Reliability
0%	$ 454.85	2.22%	$ 454.85	2.22%
10%	$ 908.70	9.82%	$ 1,253.96	10.00%
20%	$ 1,544.45	20.13%	$ 1,843.41	19.88%
30%	$ 1,971.91	30.02%	$ 3,470.56	29.95%
40%	$ 3,134.55	39.94%	$ 4,407.27	39.98%
50%	$ 4,109.02	50.00%	$ 5,251.48	49.99%
60%	$ 6,381.03	59.95%	$ 7,754.48	59.94%
70%	$ 8,956.37	70.04%	$ 8,903.02	70.02%
80%	$ 14,262.18	79.81%	$ 14,455.02	79.57%
90%	$ 14,286.09	80.25%	$ 15,100.48	80.40%
100%	$ 16,076.14	81.53%	$ 15,103.18	80.67%

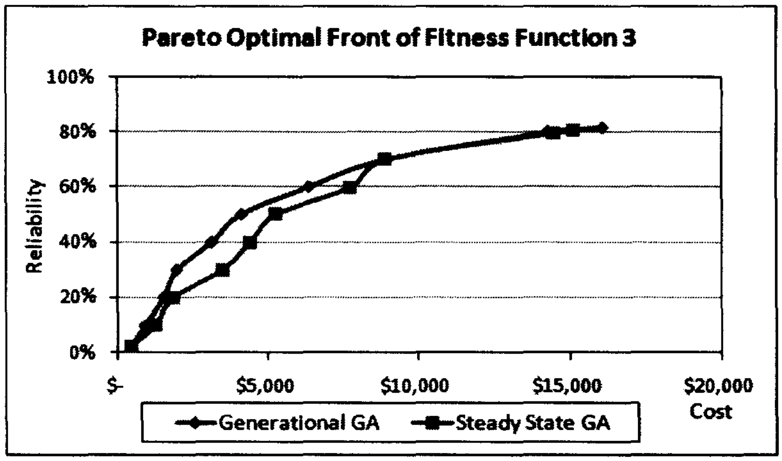

Figure 4.7. Pareto optimal front of fitness function 3 with GAs

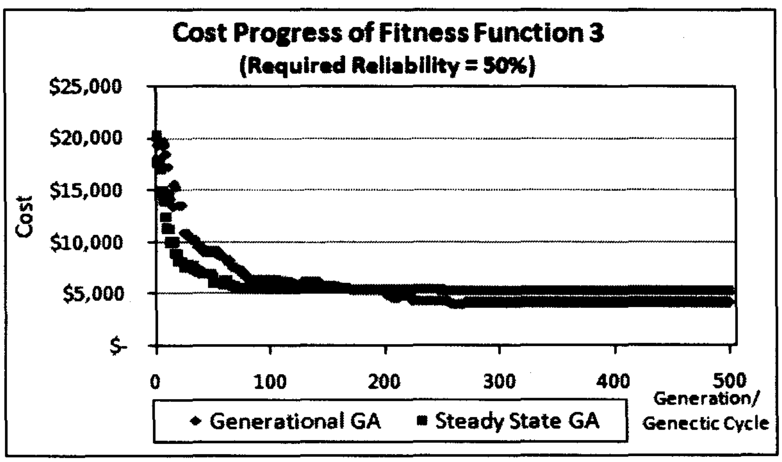

Figure 4.8. Cost progress of fitness function 3 with GAs

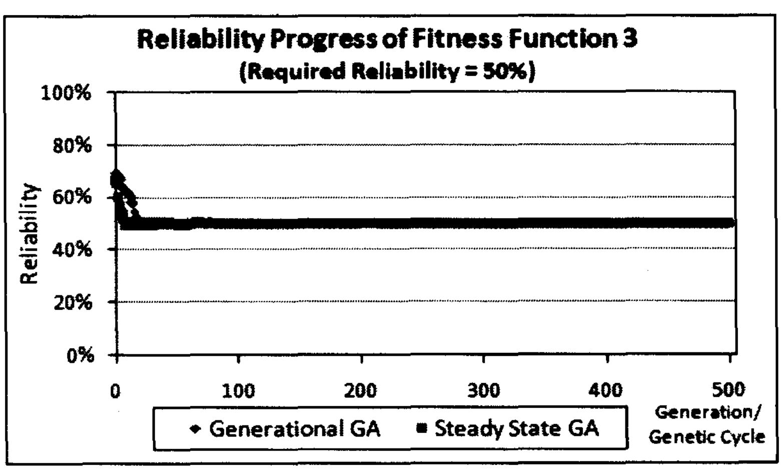

Figure 4.9. Reliability progress of fitness function 3 with GAs

Tables 4.9 and 4.10 show an example of non-dominated preventive maintenance and replacement schedules with fitness function 3 with 50% as the desired reliability. With this level of required reliability, the total cost of the system is \$4109.02 and \$5251.48 achieved by the generational and steady state genetic algorithms respectively. As it can be seen, the structure of both schedules is same as the structure found using previous fitness functions.

Table 4.9. Non-dominated preventive maintenance and replacement schedule Fitness function 3, GGA (Reliability=50% and Cost=\$4109.02)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	-	M	-	-	-	R	-	-	-	-	-	-	-	-	-	-

Table 4.10. Non-dominated preventive maintenance and replacement schedule Fitness function 3, SSGA (Reliability=50% and Cost=\$5251.48)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
2	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
3	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
4	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
5	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
6	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
7	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
8	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
9	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-
10	-	-	R	-	-	R	-	M	-	-	M	-	M	-	R	-	-	-	-	R	-	-	M	M	-	-	-	M	-	-	-	-	-	-	-	-

A comparison between Pareto optimal fronts of the three fitness functions with the genetic algorithms is presented in Figure 4.10. We can conclude that the first fitness function and the third fitness function with generational genetic algorithm produce better Pareto optimal front when compared to the fitness function 2 and the fitness function 3 with steady state genetic algorithm. These Pareto optimal fronts can be used to plan any desired levels of both objective functions. Maintenance engineers and managers can use these curves to design systems reliability in order to meet systems requirements and objectives.

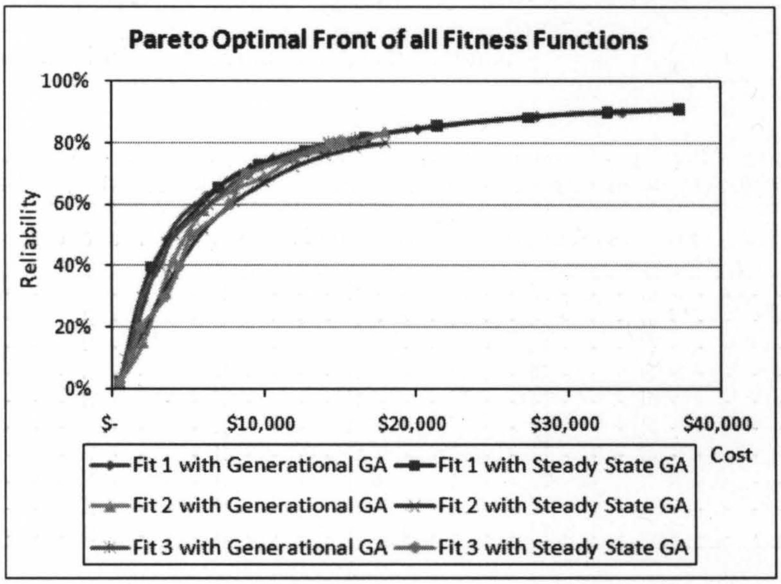

Figure 4.10. Pareto optimal solutions of all fitness functions with GAs

4.6. Simulated Annealing Algorithm

Simulated annealing (SA) algorithm is a general probabilistic method for solving combinatorial optimization problems. It involves random transitions among the solutions of the problem. Unlike iterative progress algorithms, which improve the objective value continuously, the simulated annealing algorithm may encounter some adverse changes in objective value in the course of its progress. Such changes are intended to lead to a global optimal solution instead of a local one. Annealing is a physical process in which a solid is heated up to a high temperature and then allowed to cool gradually. In this process, all of the particles arrange themselves gradually into a low energy level. The ultimate energy level depends on the level of the high temperature and the rate of cooling. The annealing process can be described as a stochastic procedure, such that at each temperature, the solid undergoes a large number of random transitions among states of different energy levels until it attains a thermal equilibrium in which the probability that the solid appears in a state with an energy level E is given by:

$$Pr(X = E) = \frac{1}{Z(t)} e^{(\frac{-E}{K_B T})} \qquad (4.9)$$

Where X denotes the random energy level of the solid, $Z(t)$ is a normalization factor, and K_B is the Boltzmann constant. The above probability distribution is called the *Boltzmann distribution.* As the temperature T decreases, equilibrium probabilities associated with higher energy level states decreases. When the temperature approaches to zero, only the states with the lowest energy level will have nonzero probability. If the cooling is not sufficiently slow, thermal equilibrium will not be attained at any temperature and consequently the solid will finally have a meta-stable condition.

There are several variations of simulated annealing, which arise to different cooling schedules and stopping criteria. The following is a general description of simulated annealing, see Kuo *et al.* (2001).

Begin Simulated Annealing
  k=0
  Select $T_{initial}$ and T_{final} if the termination criterion involves T_{final}
  Randomly produce an initial solution x_0 from S
  Determine the fitness value of the initial solution $f_0 = C(x_0)$
  While a sufficient number of times to ensure a near-equilibrium condition, do

Randomly select a transition $x_k = y$ and compute $\Delta C = C(y) - C(x_k)$. If $\Delta C \le 0$, accept the transition. If $\Delta C > 0$, accept the transition with probability $\Pr_k(\Delta C) = e^{(\frac{-\Delta C}{T_k})}$, and reject it with probability $1 - \Pr_k(\Delta C)$

If the transition is accepted, update $x_k = y$ and $f_k = C(y)$. (To accept or reject the transition with $\Delta C > 0$, First generate a random number p from (0,1). If $p \le \Pr_k(\Delta C)$, accept the transition; otherwise, reject it)

k=k+1. Find T_k from T_{k-1},based on the rule for decreasing the control parameter T

$x_k = x_{k-1}, f_k = f_{k-1}$

End while

End Simulated Annealing

Note that the transition $x_k = y$ is usually selected in such a way that y is in the neighborhood of x_k.

4.7. Implementation of the Simulated Annealing Algorithm

We use the representative data set shown in Table 3.1 and assumed same fixed cost, planning horizon and inflation and interests rates. In addition, we set the simulated annealing parameters to *initial temperature* = 1000000, *final temperature* = 0.01 and *geometric decreasing rate* = 0.98. We develop a computer program in MATLAB R2008a[1] programming environment to construct the simulated annealing algorithm and calculate the fitness functions. It is useful to mention that because of the geometric decreasing rate the number of energy levels, algorithm iterations, is 912 and the computational time is observed as less than 2 seconds. Appendix D presents the MATLAB programs of fitness functions, transition function, and simulated annealing algorithm.

[1] www.mathworks.com

4.7.1. Computational Results of Fitness Function 1

We run the simulated annealing algorithm with fitness function 1 for the set of weights for both objectives functions and achieve Pareto optimal solutions shown in Table 4.11. Figure 4.11 represents the Pareto optimal front of fitness function 1 obtained by the simulated annealing algorithm. Figures 4.12 and 4.13 illustrate cost and reliability progress during the iterations of algorithm. As we can see, the convergence of the algorithm is not too consistent but it can give a near optimal solutions.

Table 4.11. Pareto optimal solutions of fitness function 1 with SA

Weights		Simulated Annealing	
W1	W2	Cost	Reliability
0.0	1.0	$ 37,334.28	91.03%
0.1	0.9	$ 36,632.37	90.10%
0.2	0.8	$ 33,585.74	88.89%
0.3	0.7	$ 26,915.20	84.75%
0.4	0.6	$ 18,569.88	80.22%
0.5	0.5	$ 13,451.70	74.78%
0.6	0.4	$ 9,723.55	68.17%
0.7	0.3	$ 8,841.34	65.32%
0.8	0.2	$ 6,572.84	57.78%
0.9	0.1	$ 4,761.10	46.51%
1.0	0.0	$ 454.85	2.22%

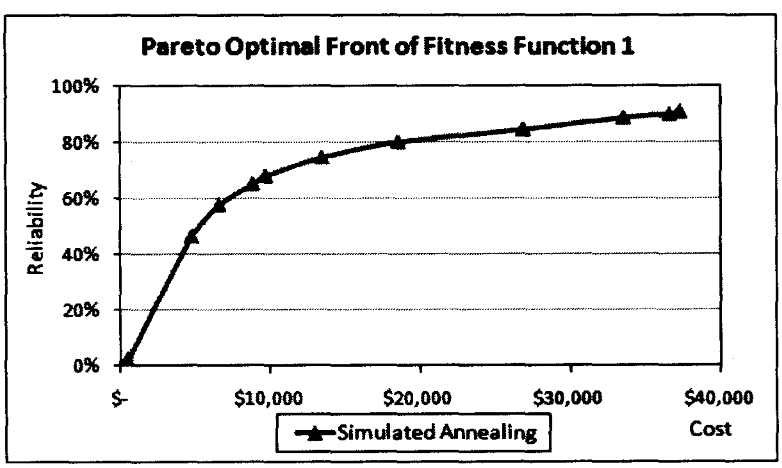

Figure 4.11. Pareto optimal front of fitness function 1 with SA

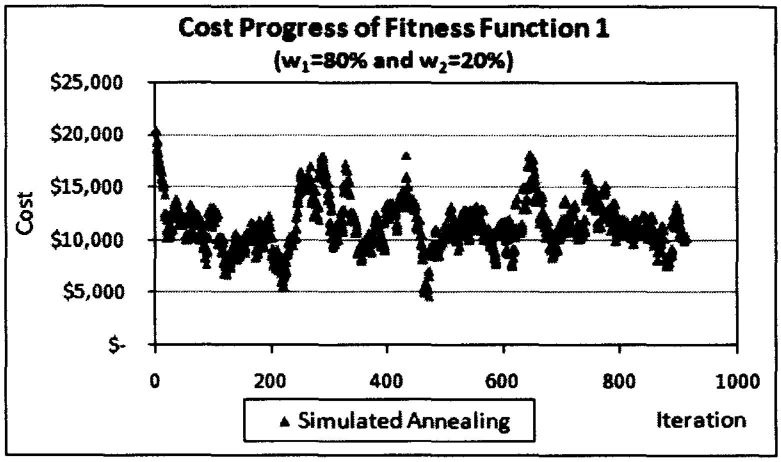

Figure 4.12. Cost progress of fitness function 1 with SA

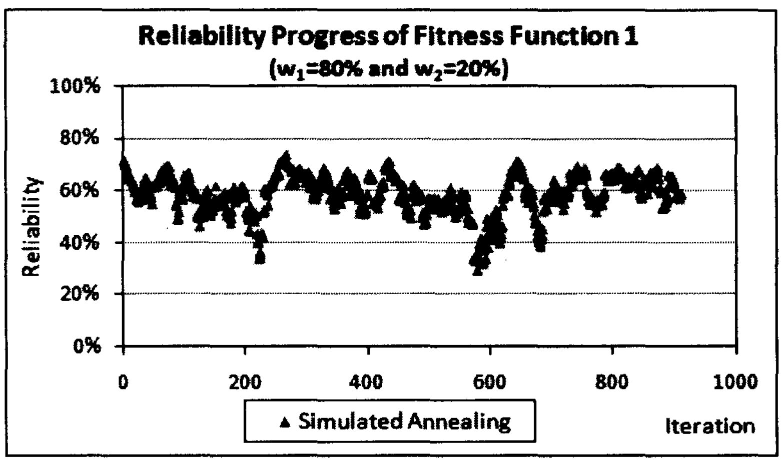

Figure 4.13. Reliability progress of fitness function 1 with SA

Table 4.12 shows a non-dominated preventive maintenance and replacement schedule of fitness function 1 for 0.8 and 0.2 as the weights for cost and reliability objective functions. With these weights, the values of objective functions are \$6572.84 and 57.78%, which are slightly worse than the results achieved by generational and steady state genetic algorithms. It should be mentioned that all maintenance and replacement actions tend to occur in the same month, which reflects the effect of the fixed cost.

Table 4.12. Non-dominated preventive maintenance and replacement schedule Fitness function 1, SA (w_1=80% and w_2=20%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
2	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
3	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
4	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
5	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
6	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
7	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
8	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
9	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-
10	-	-	-	-	R	-	R	-	-	-	M	M	-	M	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	-	R	-	-	-	-

4.7.2. Computational Results of Fitness Function 2

We optimize the model (4.5) with fitness function 2 and by considering different levels of budget in the system and the obtain Pareto optimal solutions presented in Table 4.13. Figure 4.14 represents the Pareto optimal solutions obtained by simulated annealing algorithm with fitness function 2. Figures 4.15 and 4.16 show cost and reliability progress during the iterations of algorithm. It is clear that despite of the convergence of the algorithm with fitness function 1, the convergence in this case is completely consistent over the iterations.

Table 4.13. Pareto optimal solutions of fitness function 2 with SA

Given Budget	Simulated Annealing Cost	Reliability
$ 400.00	$ 454.85	2.22%
$ 2,000.00	$ 1,999.24	18.94%
$ 4,000.00	$ 3,999.99	37.23%
$ 6,000.00	$ 5,999.30	51.62%
$ 8,000.00	$ 7,998.99	60.77%
$ 10,000.00	$ 10,000.64	67.17%
$ 12,000.00	$ 11,999.10	72.39%
$ 14,000.00	$ 13,999.99	76.03%
$ 16,000.00	$ 16,000.23	78.55%
$ 18,000.00	$ 18,000.75	80.11%
$ 20,000.00	$ 19,999.83	81.45%

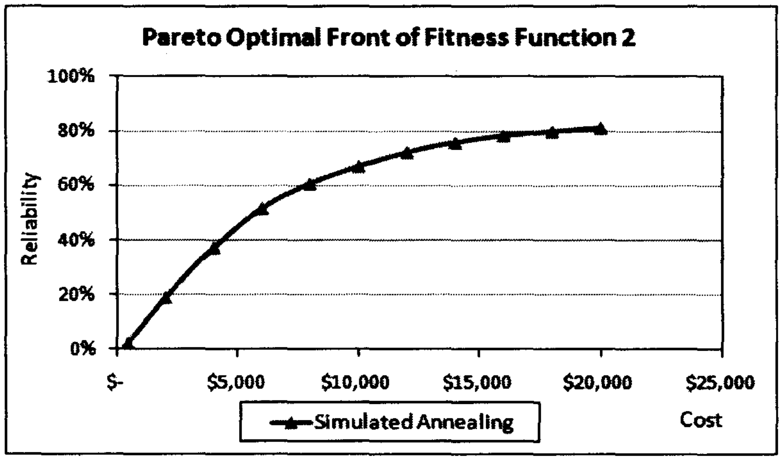

Figure 4.14. Pareto optimal front of fitness function 2 with SA

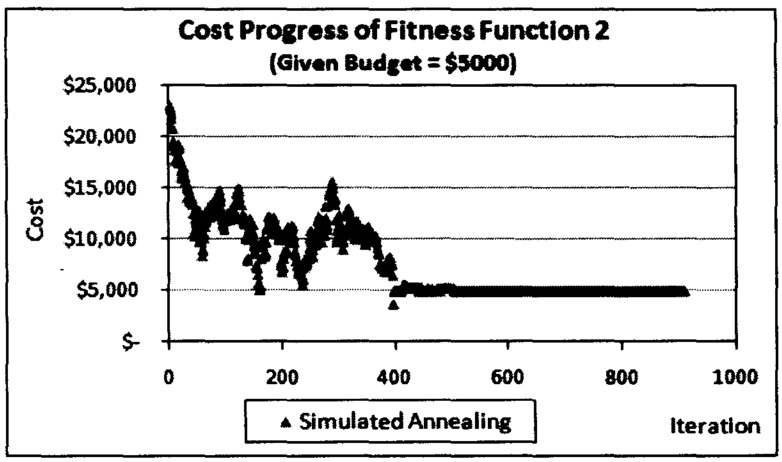

Figure 4.15. Cost progress of fitness function 2 with SA

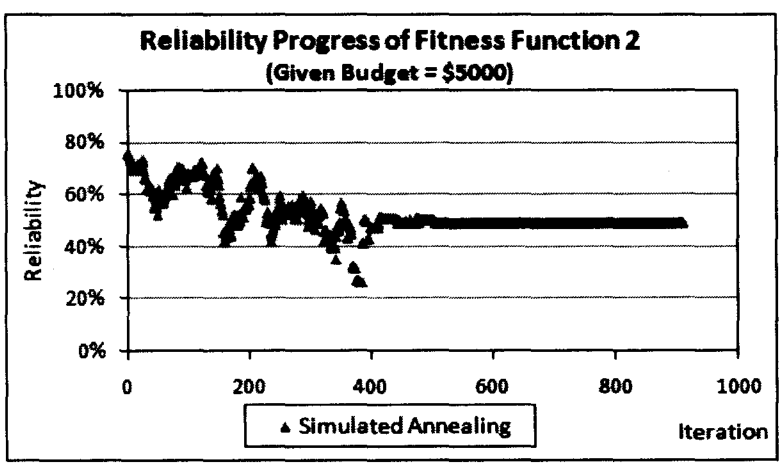

Figure 4.16. Reliability progress of fitness function 2 with SA

Table 4.14 shows a non-dominated preventive maintenance and replacement schedule with fitness function 2 for the given budget equal to $5000. With this amount of budget, the reliability of the system is 48.88% and same as what was mentioned in section 4.7.1 the result is not as good as what is achieved by generational and steady genetic algorithms.

Table 4.14. Non-dominated preventive maintenance and replacement schedule Fitness function 2, SA (Budget=$5000 and Reliability=48.88%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
2	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
3	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
4	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
5	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
6	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
7	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
8	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
9	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-
10	-	-	-	-	-	-	-	-	M	-	-	-	R	-	R	-	-	-	-	-	-	M	-	M	M	-	-	-	-	-	R	-	-	-	-	-

4.7.3. Computational Results of Fitness Function 3

Finally, Table 4.15 presents the Pareto optimal solutions of the model with fitness function 3 for different required reliability values. Figure 4.17 represents the Pareto optimal solutions obtained by simulated annealing algorithm for fitness function 3. Figures 4.17 and 4.18 show cost and reliability progress during the iterations of algorithm and as we can see the convergence of the algorithm with fitness function 3 is very consistent after half of the total iterations.

Table 4.15. Pareto optimal solutions of fitness function 3 with SA

Required Reliability	Simulated Annealing Cost	Simulated Annealing Reliability
0%	$ 454.85	2.22%
10%	$ 1,120.35	9.97%
20%	$ 1,823.81	20.01%
30%	$ 2,356.39	30.00%
40%	$ 3,201.11	40.09%
50%	$ 5,256.75	49.84%
60%	$ 6,523.00	60.05%
70%	$ 9,177.98	70.01%
80%	$ 15,108.03	79.98%
90%	$ 16,249.33	81.11%
100%	$ 18,242.11	84.25%

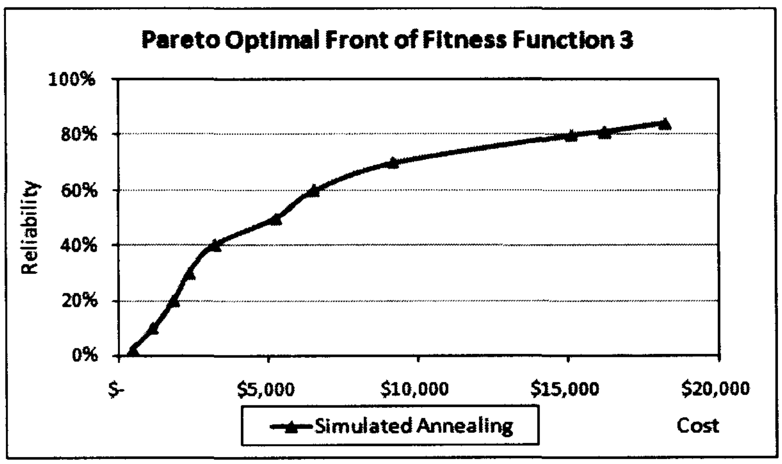

Figure 4.17. Pareto optimal front of fitness function 3 with SA

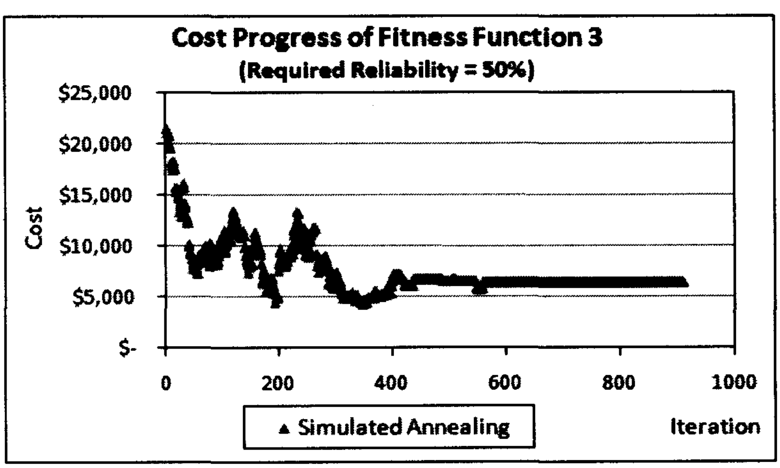

Figure 4.18. Cost progress of fitness function 3 with SA

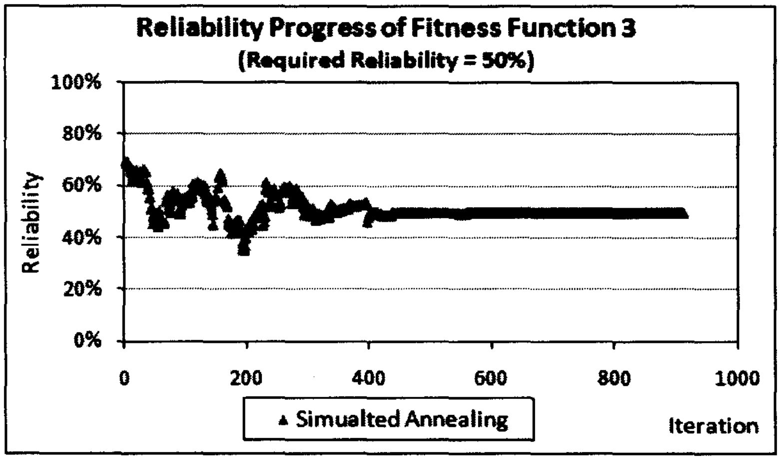

Figure 4.19. Reliability progress of fitness function 3 with SA

Table 4.16 shows a non-dominated preventive maintenance and replacement schedule of fitness function 3 with 50% as the required reliability. With this level of reliability, the total cost of the system is $5256.75, which is almost same as the total cost achieved by steady state genetic algorithm but it is not as good as the total cost obtained by generational genetic algorithm with third fitness function.

Table 4.16. Non-dominated preventive maintenance and replacement schedule Fitness function 3, SA (Reliability=50% and Cost=$5256.75)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
2	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
3	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
4	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
5	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
6	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
7	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
8	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
9	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
10	-	-	R	-	-	-	M	-	M	-	M	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-

An advantage of simulated annealing is its ability to search neighborhoods to find global optimum solution instead of just finding local one. This can be observed in Figures 4.15, 4.16, 4.18, and 4.19 in points that the total cost drops or the reliability rises drastically. A comparison between Pareto optimal fronts of the fitness functions using the simulated annealing algorithm is presented in Figure 4.20. We can observe and conclude that all fitness functions result to the same Pareto optimal solutions but the first fitness function has a lack of convergence consistency in the iterations of the algorithm.

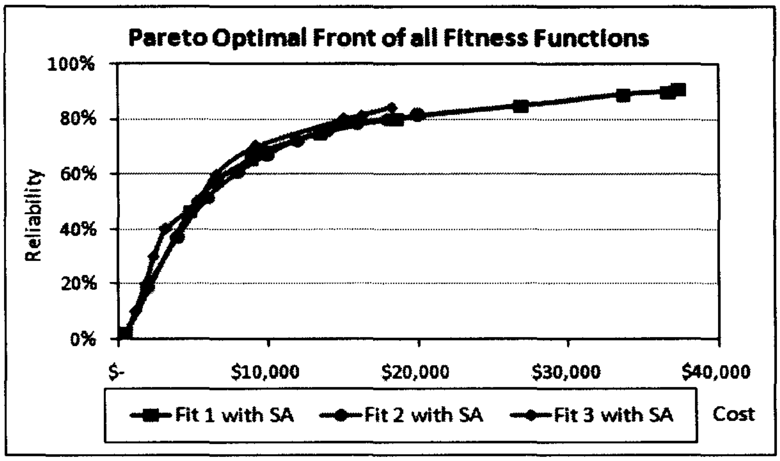

Figure 4.20. Pareto optimal front of all fitness functions with SA

4.8. Chapter Summary

In this chapter, an extension of the optimization models formulated in Chapter 3 was presented by considering engineering economy aspects. A new model multi-objective optimization model was formulated. Generational and steady state genetic algorithms as well as a simulated annealing algorithm were used to optimize the model and new crossover and mutation procedures were developed based on the special structure of the model. In addition, three different fitness functions were developed and utilized to evaluate Pareto optimal solutions. By analyzing the computational time and results of the algorithms, we showed the efficiency and effectiveness of the solution methods. Finally, the convergence of the algorithms in terms of cost and reliability progress was demonstrated and analyzed. In the next chapter, a complete comparison of the exact and metaheuristic algorithms along with the sensitivity analysis of the optimization model parameters will be presented.

CHAPTER 5

SENSITIVITY ANALYSIS AND COMPARISON OF ALGORITHMS

5.1. Introduction

In Chapter 3, we developed two optimization models and solved them via an exact solution approach. We extended the models to consider multiple objectives and applied two types of genetic algorithms along with a simulated annealing to reach non-dominated solutions. This chapter further refines the analysis and includes two main parts. First, we examine the effect of the optimization model parameters on the resulting structure of the optimal preventive maintenance and replacement schedule of multi-component systems. Second, we compare the computational efficiency and accuracy of the metaheuristic methods with the exact method and show the advantages of each.

5.2. Sensitivity Analysis on Parameters

5.2.1. Experimental Design

The optimization models developed in Chapter 3 have two different types of parameters; component reliability characteristics, and costs associated with preventive maintenance and replacement activities. Component reliability parameters include λ_i and β_i, the characteristic life (scale) and the shape parameters of component, and α_i, the improvement factor for each component i. Each component also has three different types of cost, failure cost, maintenance cost, and replacement cost. In addition, the optimization models have constraints on the overall reliability and the total cost, (required reliability and the given budget). Finally, there is a fixed cost charged whenever a component is maintained or replaced in a period.

We design two 2^3 factorial design experiments to find the effect of the optimization model parameters on the structure of the optimal schedule. Based on this consideration, each experiment has three factors, each with two levels. With one replicate in each experiment, there are 8 different trials.

The first experiment, scenario 1, assumes that the reliability parameters of all components are the same, but each component has two levels, low and high, for failure, maintenance, and replacement costs; as shown in Table 5.1. The second experiment, scenario 2, assumes that the failure, maintenance, and replacement costs of all components are the same, but each component has two levels for the reliability parameters; see Table 5.2. We consider each scenario and solve both models with and without the fixed cost. Hence, we achieve four different optimal preventive maintenance and replacement schedules for each scenario.

Table 5.1. Parameters for the scenario 1

Component	λ	β	a	Failure Cost ($)	Maintenance Cost ($)	Replacement Cost ($)
1	0.00025	2.20	0.50	100	100	100
2	0.00025	2.20	0.50	100	100	500
3	0.00025	2.20	0.50	100	500	100
4	0.00025	2.20	0.50	100	500	500
5	0.00025	2.20	0.50	500	100	100
6	0.00025	2.20	0.50	500	100	500
7	0.00025	2.20	0.50	500	500	100
8	0.00025	2.20	0.50	500	500	500

Table 5.2. Parameters for the scenario 2

Component	λ	β	a	Failure Cost ($)	Maintenance Cost ($)	Replacement Cost ($)
1	0.00010	1.80	0.25	100	100	100
2	0.00010	1.80	0.75	100	100	100
3	0.00010	2.50	0.25	100	100	100
4	0.00010	2.50	0.75	100	100	100
5	0.00050	1.80	0.25	100	100	100
6	0.00050	1.80	0.75	100	100	100
7	0.00050	2.50	0.25	100	100	100
8	0.00050	2.50	0.75	100	100	100

5.2.2. Computational Results of the Scenario 1

We utilized LINGO[1] software to solve the models to obtain an optimal solution. We set the required reliability to 50% in the first model and the given budget to $8000 and $18000 for the models without and with the fixed cost respectively in the second model. In addition, we considered the fixed cost equal to $1000 and 36 month as the planning horizon.

Tables 5.3 through 5.6 present optimal schedules for the first scenario for both models. At first glance, the effect of the fixed cost on optimal schedules is clearly evident. As expected, the fixed cost forces maintenance and replacement activities to occur in same periods, as shown in Tables 5.4 and 5.6. In section 3.2.4.4, it was mentioned that an *N*-component model without fixed cost is similar to *N* single-component models in which one could simply find the best sequence of actions for a

[1] http://www.lindo.com

component regardless of the actions taken to other components. Tables 5.3 and 5.5 show such a schedule.

Observing the structure of the optimal schedules, one can see that the failure cost does not noticeably affect structure of the schedule and the frequency of actions. It can be seen that there is no big difference between the schedule of first four components with less failure cost and the last four components with more failure cost.

We find that components 2 and 6 are only maintained, because the maintenance cost for components 2 and 6 are one fifth of their replacement cost, however they have different failure costs, as shown in Table 5.1. We can also see that components 1, 3, 4, 5, 7 and 8 are only replaced, except one maintenance action for component 5 in Tables 5.5 and 5.6 and a maintenance action for component 1 in Table 5.6. In the above components, maintenance cost is greater or equal to replacement cost and it seems that in this case the maintenance and replacement schedule contains replacement actions instead of maintenance actions. Finally, we can observe that components 4 and 8 are replaced less frequently than other components because of their high maintenance and replacement costs.

By reviewing the maintenance and replacement costs presented in Table 5.1 and the structure of the optimal schedules, we can conclude that if all components have the same reliability parameters, structure and frequency of activities in the optimal schedule is affected by just ratio of the maintenance and replacement costs. In addition, we can say that the failure cost does not play a significant role in the structure of maintenance and replacement schedule.

Table 5.3. Scenario 1-Optimal schedule that minimizes total cost without fixed cost (Reliability=50.00% and Cost=$8503.29)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	R	-	-	-	R	-	-	-	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	R	-	-	-
2	-	-	-	-	M	-	-	M	-	M	-	M	-	-	M	-	M	-	-	M	-	-	M	-	M	-	-	-	-	-	-	M	-	-	-	-
3	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-
5	-	-	-	-	-	R	-	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	R	-	-	-	-	-
6	-	-	-	M	-	M	-	-	M	-	M	-	-	-	M	-	M	M	-	-	-	M	-	-	M	-	M	-	-	M	M	-	-	-	-	-
7	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-

Table 5.4. Scenario 1-Optimal schedule that minimizes total cost with fixed cost (Reliability=50.00% and Cost=$18301.00)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	R	-	-	R	-	-	-	R	-	-	-	-	-	-
2	-	-	-	-	-	M	-	-	-	R	-	-	-	-	M	-	-	-	-	M	-	-	M	-	-	M	-	-	-	M	M	-	-	-	-	-
3	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	R	-	-	R	-	-	-	R	-	-	-	-	-	-
4	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-
5	-	-	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	R	-	-	R	-	-	-	R	-	-	-	-	-	-
6	-	-	-	-	-	M	M	-	-	M	-	-	-	-	M	-	-	-	-	R	-	-	M	-	-	M	-	-	-	M	M	-	-	-	-	-
7	-	-	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-
8	-	-	-	-	-	-	R	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-

Table 5.5. Scenario 1-Optimal schedule that maximizes reliability without fixed cost (Budget=$8000 and Reliability=45.46%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	R	-	R	-	-	-	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	R	-	-	R	-	-	-	-
2	-	-	-	-	M	-	-	M	-	-	M	-	-	M	-	M	M	-	-	-	-	M	-	-	M	-	-	M	-	-	-	-	-	-	-	-
3	-	-	-	-	R	-	-	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-
4	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-
5	-	-	-	R	-	-	-	R	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	-	R	-	M	-	-	-	-	R	-	-
6	-	-	-	-	-	M	-	-	M	-	-	-	M	-	M	-	-	-	-	M	-	M	-	-	M	-	-	M	-	-	-	-	-	-	-	-
7	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	R	-	-	R	-	R	-	-	-	-	-	R	-	-	-
8	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-

Table 5.6. Scenario 1-Optimal schedule that maximizes reliability with fixed cost (Budget=$18000 and Reliability=46.90%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	R	-	-	-	M	-	-	-	-	-	R	-	-	-	-
2	-	-	-	-	M	-	-	M	-	-	-	-	R	-	-	-	-	-	-	M	-	M	-	-	-	M	-	-	-	M	-	-	-	-	-	-
3	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	R	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-	-
5	-	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	M	-	R	-	-	-	-	-	R	-	-	-	R	-	-	-	-	-	-
6	-	-	-	-	M	-	-	M	-	-	-	-	M	-	-	-	-	R	-	M	-	M	-	-	-	M	-	-	-	M	-	M	-	-	-	-
7	-	-	-	-	R	-	-	R	-	-	-	-	-	-	-	-	-	R	-	R	-	-	-	-	-	R	-	-	-	R	-	-	-	-	-	-
8	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-

5.2.3. Computational Results of the Scenario 2

In this scenario, we considered 50% as the required reliability, \$4000 and \$12000 as the given budgets, and \$1000 as the fixed cost. LINGO was used to solve the optimization models. Tables 5.7 to 5.10 present the optimal schedules for the second scenario in both models. As in the first scenario, we can see that the fixed cost has an effect on the maintenance and replacement activities occurrence at the same period; see Tables 5.8 and 5.10.

We find that in this scenario all activities are replacement, except four maintenance actions presented in Table 5.10. This indicates that for components with the same failure, maintenance, and replacement costs, and different scale and shape parameters, the value of the improvement factor does not affect structure of the optimal schedule. For example, by comparing the schedule for the first two components, it can be seen that a smaller improvement factor reduces the effective age of the component more than a higher one and thus components with the lower improvement factor are more likely to be maintained; see two maintenance actions of component 7 in Table 5.10.

The scale and shape parameters play the most important role in the configuration of the optimal schedules, especially the shape parameter. For example, consider components 1 and 2 and components 3 and 4. Both pairs have the same scale parameter but the latter have larger value of the shape parameter. This results in more replacement activities for the second pair. The frequency of replacement activities in first four components can be seen in Tables 5.7 to 5.10. On the other hand, the scale parameter has an effect on the structure of the optimal schedules, but not as much as the shape parameter does. For example, compare the pair of components 1 and 2 and the pair of components 5 and 6. Both pairs have the same shape parameter but the second one has a greater scale parameter than the first. Hence, we see more frequent replacement activities in the second pair. Finally, we can say that less reliable components with higher deterioration rate are replaced more frequently than the more reliable components with lower deterioration rate. Compare the frequency of replacements in components 7 and 8 with great scale and shape parameters (less reliable components with high deterioration rate) with components 1 and 2 with small parameters values (more reliable components with low deterioration rate).

Table 5.7. Scenario 2-Optimal schedule that minimizes total cost without fixed cost (Reliability=50.00% and Cost=$3669.26)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	R	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-
4	-	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-
5	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-
7	-	-	-	R	-	-	R	-	-	R	-	-	R	-	-	R	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-
8	-	-	-	-	R	-	-	R	-	-	R	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	R	-	-	-	R	-	-	-	-	-

Table 5.8. Scenario 2-Optimal schedule that minimizes total cost with fixed cost (Reliability=50.00% and Cost=$12668.80)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-
4	-	-	-	-	-	-	-	R	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-
5	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-
7	-	-	R	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-
8	-	-	R	-	-	-	-	R	-	-	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	-	-	R	-	-	-	-	-

Table 5.9. Scenario 2-Optimal schedule that maximizes reliability without fixed cost (Budget=$4000 and Reliability=53.25%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	-
4	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-
7	-	R	-	R	-	-	R	-	-	R	-	R	-	-	-	-	R	-	-	R	-	R	-	-	R	-	-	-	R	-	-	-	R	-	-	-
8	-	-	R	-	-	-	R	-	-	-	R	-	-	-	R	-	-	R	-	-	R	-	-	R	-	-	R	-	-	R	-	-	R	-	-	-

Table 5.10. Scenario 2-Optimal schedule that maximizes reliability with fixed cost (Budget=$12000 and Reliability=48.45%)

Month/ Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	R	-	-	R	-	-	-	-	-	M	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-
4	-	-	-	-	-	-	R	-	-	-	-	-	R	-	R	-	-	-	-	M	-	-	-	-	R	-	-	-	R	-	-	-	-	-	-	-
5	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-
6	-	-	-	-	-	-	R	-	-	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	R	-	-	R	-	-	-	-	-	R	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	M	-	M	-	-	-	-	-
8	-	-	-	R	-	-	R	-	-	-	-	-	R	-	R	-	-	-	-	R	-	-	-	-	R	-	-	-	R	-	R	-	-	-	-	-

5.3. Comparison of Exact and Metaheuristic Algorithms

5.3.1. Experimental Design

The optimization models developed and solved via generalized reduced gradient and branch-and-bound algorithms in LINGO software were single objective models. We considered engineering economy parameters, extended them to consider multiple objectives, and solved using metaheuristic algorithms.

In order to analyze the efficiency and accuracy of the proposed metaheuristic algorithms and compare them with the exact method, we present a comprehensive experiment. We consider just the single objective models without engineering economy parameters and optimized both models with 2 sets of data for series systems with 5 and 10 components and 6, 12, 18, 24, 30, and 36 planning horizons. The reliability and costs associated with components are same as in the original dataset; and for the 5-component system, we use the first 5 components. Finally, for different planning horizons, we assume different required reliability values and budget amounts for each problem. We utilized MATLAB R2008a[1] programming environment to implement the models and optimize them via generational genetic algorithm (GGA), steady state genetic algorithm (SSGA), and simulated annealing (SA). The first 4 columns of Tables 5.11 and 5.12 present the structure of the experiment.

5.3.2. Computational Results

Tables 5.11 and 5.12 show the computational results of the experiment. The results include objective function values, total cost for the first model and reliability for the second model, amount of the reliability and consumed budgetary constraints, the gap of objective function achieved by metaheuristic algorithm in compare with what is achieved by exact method and finally the computational time (CPU time) for each problem and algorithm. We find that the value of objective functions achieved by the generalized reduced gradient and branch-and-bound algorithms is always smaller than values of objective function achieved by metaheuristic algorithms in the first model and vice versa in the second one; as shown in fifth column of Tables 5.11 and 5.12 and Figures 5.1, 5.4, 5.7, and 5.10. The reason is that the metaheuristic algorithms can reach near optimal solutions instead of exact optimal solutions. In addition, we can see that the exact method does not violate the right hand side values of the main constraint, required reliability for the first model and given budget for the second one. In some cases, the metaheuristic algorithms violate the

[1] www.mathworks.com

constraints slightly; compare the values in the sixth column of the tables in some rows.

We also calculate the objective function gaps for metaheuristic algorithms. As seen in the first model the gap varies by about 2% for the generational and steady state genetic algorithms and about 4-6% for the simulated annealing algorithm. The interesting thing is that the gap is almost constant by increasing the problem size in terms of number components and periods as shown in Figures 5.2 and 5.5. We can conclude that the metaheuristic algorithms work well for large-scale problems using Model 1. Such a gap consistency is not observed in the second model and the gap changes too much even for small size problems; see the problems with 5 components in Table 5.12 and also Figure 5.8. However, it can be seen that for large-scale problems, the simulated annealing algorithm works well and its objective function gap varies between 0.1-7%, which is almost constant, see Figure 5.11.

We analyzed the computational time of each algorithm; see the last column of Tables 5.11 and 5.12. We find that the computational time of the exact method goes up exponentially by increasing the size of the problems, especially for the problems with more than 24 periods as the planning horizon as presented in Figures 5.3, 5.6, 5.9, and 5.12. It can be observed that for any problem size, the computational time of all metaheuristic algorithms in both models is completely constant and less than 2 minutes. Based on the analysis of computational results, we can conclude that if it is necessary to solve a preventive maintenance and replacement scheduling optimization model once and use the optimal schedule for a long-term planning horizon, one can use an exact method to optimize it, regardless of how long it takes. On the other hand, if someone wants to solve a large-scale condition-based model day by day, he or she can use metaheuristic algorithms to achieve a near optimal solution and be sure that this solution is good enough to use in other models, such as in simulation models.

Table 5.11. Comparison of exact and metaheuristic algorithms in Model 1

Number of Components	Number of Periods	Required Reliability	Algorithm	OFV Total Cost	Reliability	OFV Gap	Computational Time (minute)
5	6	98%	GRG with BB	$ 4,503.79	98.00%		0
			GGA	$ 4,594.82	98.00%	2.02%	0
			SSGA	$ 4,606.85	98.00%	2.29%	0
			SA	$ 4,768.97	97.95%	5.89%	0
	12	90%	GRG with BB	$ 2,734.17	90.00%		0
			GGA	$ 2,794.43	89.98%	2.20%	0
			SSGA	$ 2,807.44	89.99%	2.68%	0
			SA	$ 2,875.49	90.01%	5.17%	0
	18	80%	GRG with BB	$ 3,047.54	80.00%		2
			GGA	$ 3,128.25	80.01%	2.65%	1
			SSGA	$ 3,133.56	80.01%	2.82%	1
			SA	$ 3,192.17	80.00%	4.75%	0
	24	70%	GRG with BB	$ 4,030.26	70.00%		4
			GGA	$ 4,150.30	70.01%	2.98%	1
			SSGA	$ 4,122.05	70.03%	2.28%	1
			SA	$ 4,226.46	69.50%	4.87%	0
	30	60%	GRG with BB	$ 5,050.93	60.00%		13
			GGA	$ 5,186.53	60.46%	2.68%	1
			SSGA	$ 5,192.92	60.02%	2.81%	1
			SA	$ 5,296.03	60.54%	4.85%	0
	36	50%	GRG with BB	$ 5,470.05	50.00%		33
			GGA	$ 5,605.33	50.06%	2.47%	1
			SSGA	$ 5,619.51	50.19%	2.73%	2
			SA	$ 5,730.54	49.67%	4.76%	0
10	6	97%	GRG with BB	$ 7,390.29	97.00%		0
			GGA	$ 7,545.28	97.00%	2.10%	1
			SSGA	$ 7,582.29	97.00%	2.60%	1
			SA	$ 7,803.29	97.02%	5.59%	1
	12	90%	GRG with BB	$ 9,915.48	90.00%		1
			GGA	$ 10,138.57	89.99%	2.25%	1
			SSGA	$ 10,154.58	90.10%	2.41%	1
			SA	$ 10,535.34	90.02%	6.25%	0
	18	80%	GRG with BB	$ 11,784.30	80.00%		77
			GGA	$ 12,025.88	80.57%	2.05%	1
			SSGA	$ 12,019.87	80.02%	2.00%	1
			SA	$ 12,504.48	79.80%	6.11%	0
	24	70%	GRG with BB	$ 12,305.30	70.00%		91
			GGA	$ 12,561.42	70.00%	2.08%	2
			SSGA	$ 12,573.09	70.06%	2.18%	2
			SA	$ 13,092.47	69.64%	6.40%	0
	30	60%	GRG with BB	$ 12,886.00	60.00%		142
			GGA	$ 13,224.84	60.05%	2.63%	2
			SSGA	$ 13,243.45	59.99%	2.77%	2
			SA	$ 13,737.81	59.93%	6.61%	0
	36	50%	GRG with BB	$ 13,797.10	50.00%		273
			GGA	$ 14,170.91	49.86%	2.71%	2
			SSGA	$ 14,196.45	50.00%	2.89%	2
			SA	$ 14,723.57	49.00%	6.71%	0

Table 5.12. Comparison of exact and metaheuristic algorithms in Model 2

Number of Components	Number of Periods	Given Budget	Algorithm	OFV Reliability	Total Cost	OFV Gap	Computational Time (minute)
5	6	$ 5,000	GRG with BB	98.21%	$ 5,000.00		0
			GGA	97.60%	$ 4,999.88	0.62%	0
			SSGA	97.11%	$ 5,000.05	1.12%	0
			SA	97.09%	$ 4,950.17	1.14%	0
	12	$ 3,000	GRG with BB	90.32%	$ 3,000.00		0
			GGA	85.80%	$ 3,000.01	5.00%	0
			SSGA	86.04%	$ 2,999.98	4.74%	0
			SA	85.81%	$ 2,901.17	4.99%	0
	18	$ 4,000	GRG with BB	81.24%	$ 4,000.00		1
			GGA	76.85%	$ 4,000.06	5.40%	1
			SSGA	76.69%	$ 3,998.88	5.60%	1
			SA	71.36%	$ 3,977.11	12.16%	0
	24	$ 5,000	GRG with BB	73.11%	$ 5,000.00		2
			GGA	61.71%	$ 5,000.04	15.59%	1
			SSGA	64.48%	$ 5,000.18	11.80%	1
			SA	69.37%	$ 5,000.91	5.12%	0
	30	$ 6,000	GRG with BB	64.96%	$ 6,000.00		14
			GGA	58.36%	$ 5,999.87	10.16%	1
			SSGA	56.39%	$ 5,998.63	13.19%	1
			SA	58.31%	$ 6,067.34	10.24%	0
	36	$ 7,000	GRG with BB	55.42%	$ 7,000.00		35
			GGA	48.53%	$ 6,999.26	12.43%	1
			SSGA	48.04%	$ 7,000.01	13.32%	2
			SA	46.96%	$ 6,952.44	15.27%	0
10	6	$ 10,000	GRG with BB	97.53%	$ 10,000.00		0
			GGA	96.32%	$ 9,998.75	1.24%	1
			SSGA	96.46%	$ 9,999.83	1.10%	1
			SA	97.43%	$ 10,020.74	0.10%	0
	12	$ 6,000	GRG with BB	85.06%	$ 6,000.00		4
			GGA	82.81%	$ 6,000.02	2.65%	1
			SSGA	80.20%	$ 6,000.03	5.71%	1
			SA	84.71%	$ 5,890.10	0.41%	0
	18	$ 8,000	GRG with BB	75.64%	$ 8,000.00		5
			GGA	70.79%	$ 8,000.08	6.41%	1
			SSGA	72.24%	$ 8,000.82	4.49%	1
			SA	74.15%	$ 7,986.14	1.97%	0
	24	$ 10,000	GRG with BB	63.49%	$ 10,000.00		13
			GGA	55.62%	$ 10,000.10	12.40%	2
			SSGA	55.38%	$ 9,999.97	12.77%	2
			SA	58.93%	$ 10,060.37	7.18%	0
	30	$ 12,000	GRG with BB	52.15%	$ 12,000.00		24
			GGA	45.68%	$ 12,000.20	12.41%	2
			SSGA	46.90%	$ 12,000.06	10.07%	2
			SA	50.47%	$ 12,196.98	3.22%	0
	36	$ 15,000	GRG with BB	49.91%	$ 15,000.00		92
			GGA	44.98%	$ 15,001.99	9.88%	2
			SSGA	43.86%	$ 15,000.00	12.12%	2
			SA	46.93%	$ 15,158.15	5.97%	0

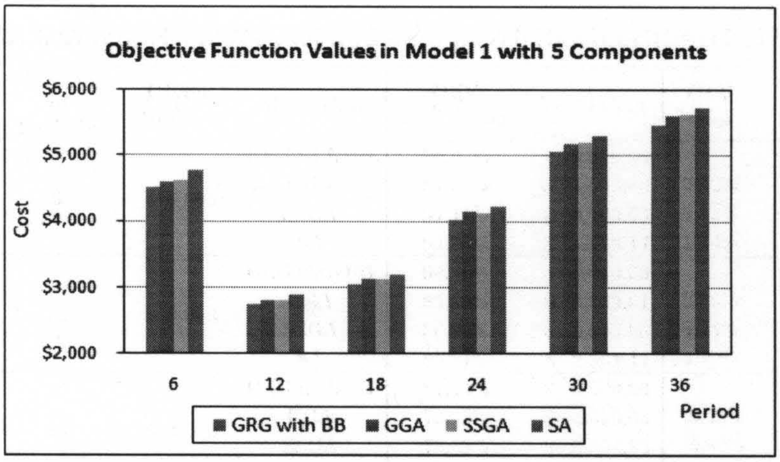

Figure 5.1. Objective function values in Model 1 with 5 components

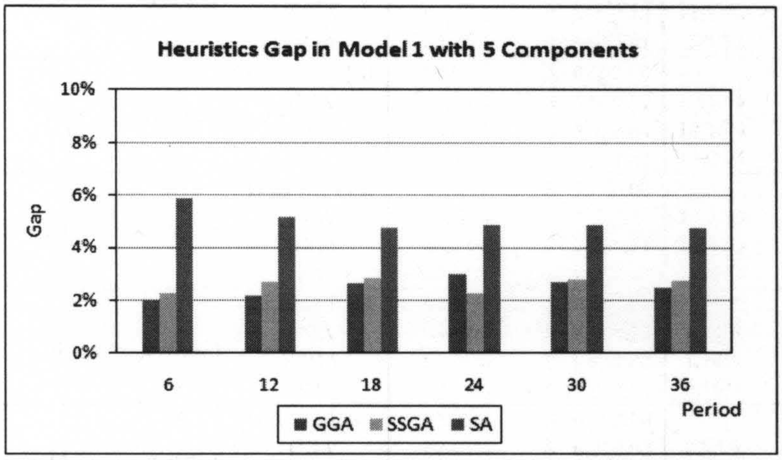

Figure 5.2. Heuristics gap in Model 1 with 5 components

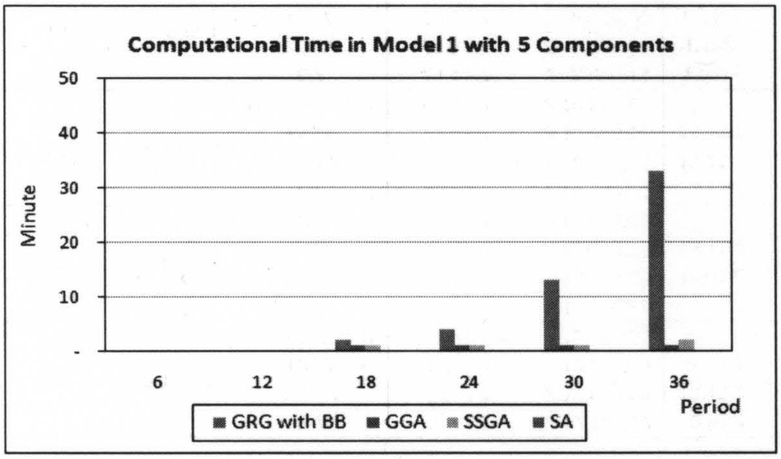

Figure 5.3. Computational time in Model 1 with 5 components

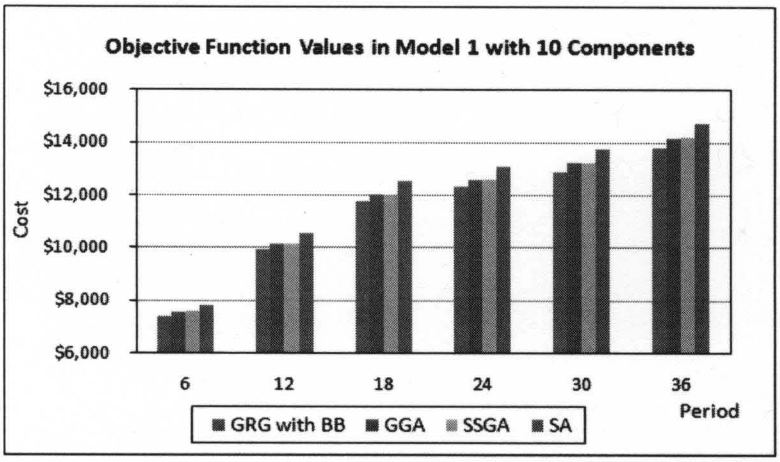

Figure 5.4. Objective function values in Model 1 with 10 components

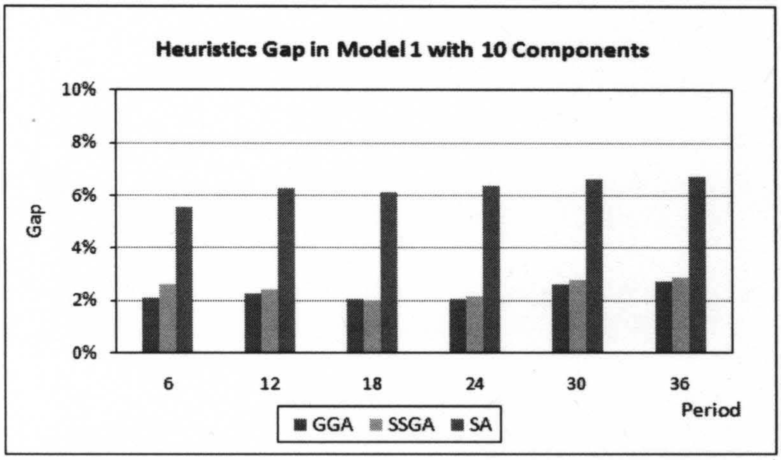

Figure 5.5. Heuristics gap in Model 1 with 10 components

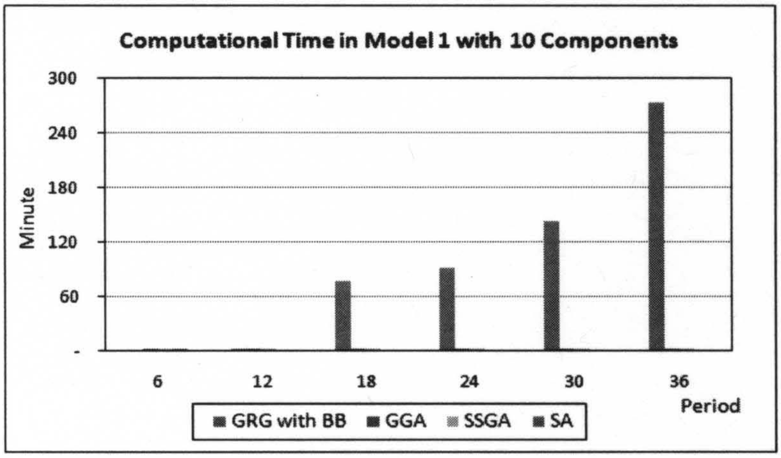

Figure 5.6. Computational time in Model 1 with 10 components

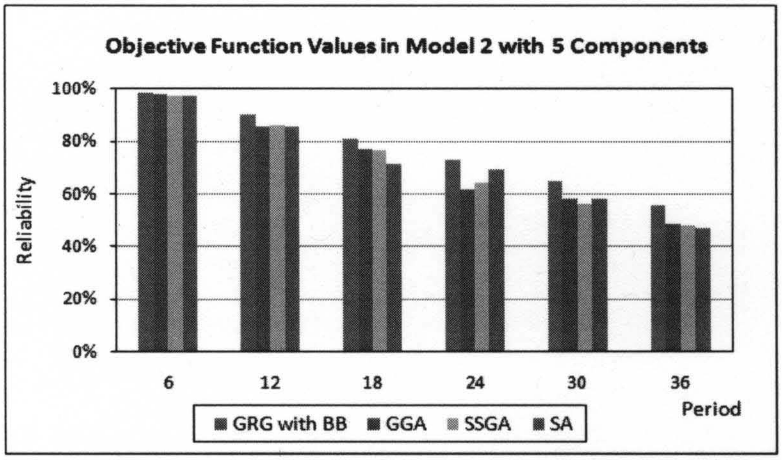

Figure 5.7. Objective function values in Model 2 with 5 components

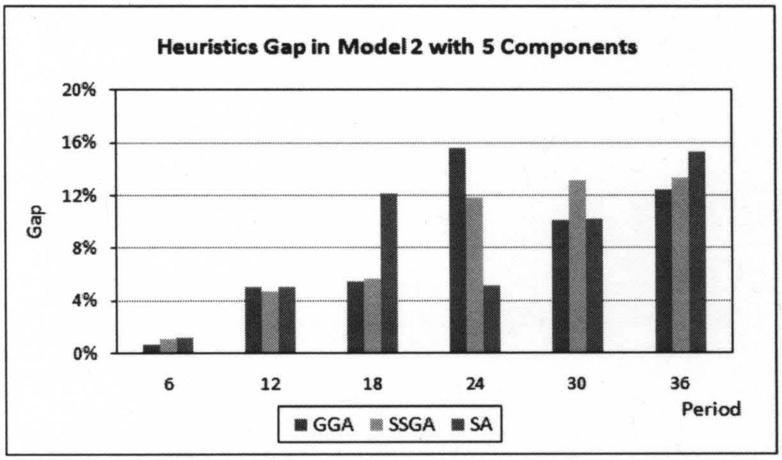

Figure 5.8. Heuristics gap in Model 2 with 5 components

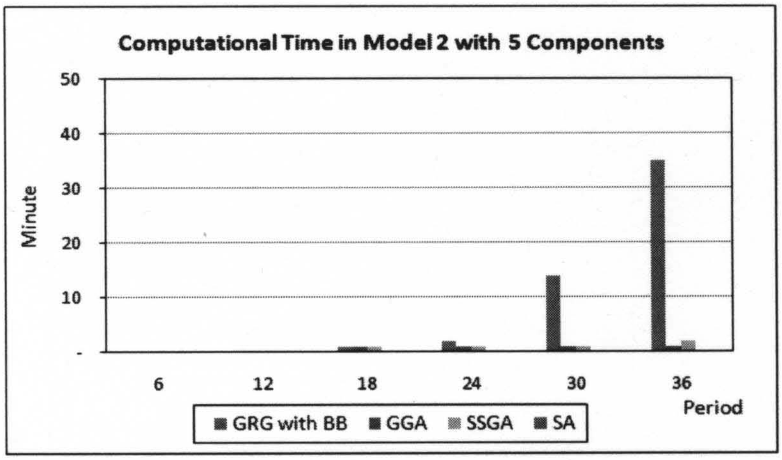

Figure 5.9. Computational time in Model 2 with 5 components

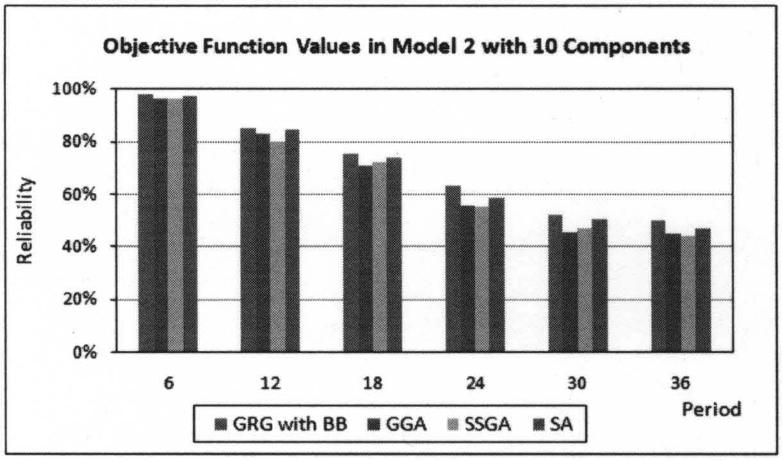

Figure 5.10. Objective function values in Model 2 with 10 components

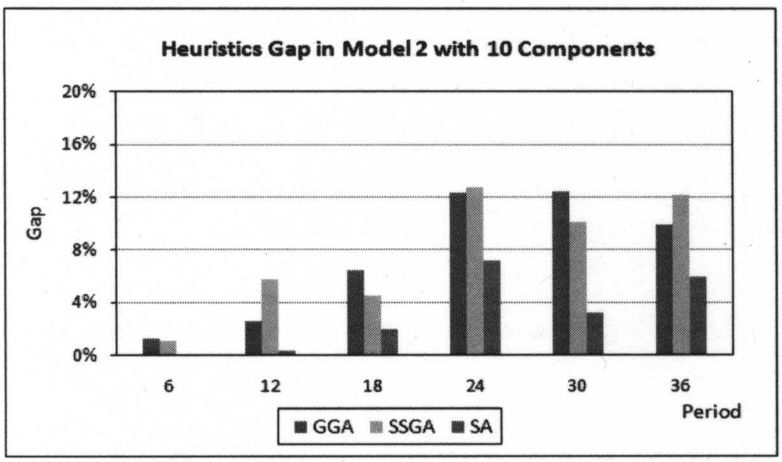

Figure 5.11. Heuristics gap in Model 2 with 10 components

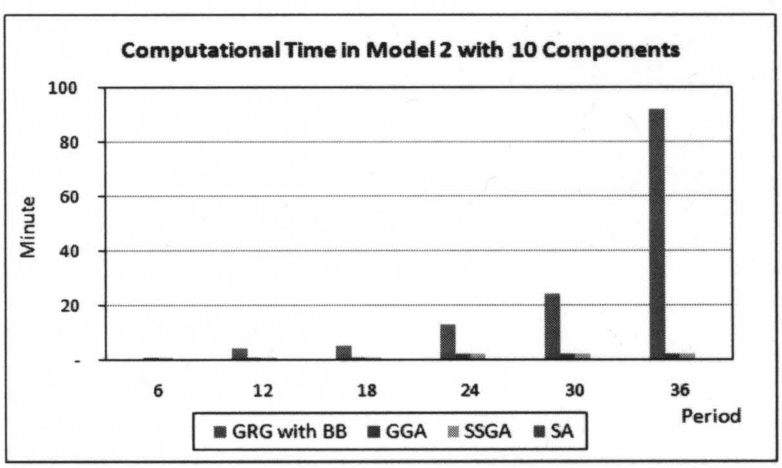

Figure 5.12. Computational time in Model 2 with 10 components

5.4. Chapter Summary

In this chapter, we presented experimental results of a sensitivity analysis on preventive maintenance and replacement scheduling optimization models. These experiments investigate the effect of the parameters on the structure of optimal schedules in multi-component systems. Two factorial design experiments based on the cost associated with maintenance and replacement activities and reliability characteristic parameters were constructed and analyzed. We also designed a comprehensive experiment to analyze and compare the efficiency and accuracy of the exact and metaheuristic algorithms and showed the advantages of each.

CHAPTER 6

IMPROVEMENT FACTOR MODELS

6.1. Introduction

In previous chapters, we developed, extended, and analyzed optimization models to determine an optimal preventive maintenance and replacement schedule in multi-component systems. In this chapter, we prove a closed-form function to show the effectiveness of maintenance actions in long-term planning horizons. As we mentioned in 3.3.1, we review current age reduction and improvement factor models, present a new mathematical function, and apply it into the optimization models. We show the effectiveness of proposed function by comparing its computational results.

6.2. Formulation

In Chapter 3, we defined effective ages of a system at the start and end of each period denoted by X_j and X'_j respectively and presented an equation to relate them to each other by the length of each period T/J as follow:

$$X'_j = X_j + \frac{T}{J} \quad \text{for } j = 1,\ldots,T \tag{6.1}$$

In addition, we assumed the initial age of the system is equal to zero. We also assume that the maintenance activity occurs at the end of the each period and effectively reduces the age of the system at the start of the next period based on an "improvement factor" (*aka* "age reduction factor"). This kind of maintenance that does not change failure characteristic of system but reduces its effective age is known as "minimal repair".

$$X_{j+1} = \alpha \cdot X'_j \quad \text{for } j = 1,\ldots,T \text{ and } (0 \le \alpha \le 1) \tag{6.2}$$

Note that when $\alpha = 0$, the effect of maintenance is to return the system to a state of "good-as-new" and it corresponds to replacement of the system. When $\alpha = 1$, maintenance has no effect, and the system remains in a state of "bad-as-old" which corresponds to "do nothing". Without lose of generality, we can always assume that $0 \le \alpha \le 1$.

Suppose a system is maintained during its service life without any replacement. We can calculate its effective age at the start and end of each period as a function of length of each period, number of maintenance actions, and amount of improvement factor based on the following equations:

$$\begin{cases} X_1 = 0, X'_1 = X_1 + \frac{T}{J} = \frac{T}{J} \\ X_2 = \alpha \cdot X'_1 = \alpha \times \frac{T}{J}, X'_2 = X_2 + \frac{T}{J} = \frac{T}{J}(\alpha + 1) \\ X_3 = \alpha \cdot X'_2 = \frac{T}{J}(\alpha^2 + \alpha), X'_3 = X_3 + \frac{T}{J} = \frac{T}{J}(\alpha^2 + \alpha + 1) \\ X_4 = \alpha \cdot X'_3 = \frac{T}{J}(\alpha^3 + \alpha^2 + \alpha), X'_4 = X_4 + \frac{T}{J} = \frac{T}{J}(\alpha^3 + \alpha^2 + \alpha + 1) \\ \vdots \\ X_j = \alpha \cdot X'_j = \frac{T}{J}(\alpha^{j-1} + \alpha^{j-2} + \alpha^{j-3} + \cdots + \alpha), X'_j = X_j + \frac{T}{J} = \frac{T}{J}(\alpha^{j-1} + \alpha^{j-2} + \alpha^{j-3} + \cdots + \alpha + 1) \end{cases} \tag{6.3}$$

$$X_j = \frac{T}{J}\left(\sum_{r=1}^{j-1} \alpha^r\right) \tag{6.4}$$

$$X'_j = \frac{T}{J}\left(\sum_{r=1}^{j-1} (\alpha^r + 1)\right) \tag{6.5}$$

Now if we assume an unlimited service life for a system with a large number of maintenance actions, we can measure a lower-bound for its effective age by taking a limit of the effective ages when number of maintenance actions goes to infinity. (Note that $0 < \alpha < 1$)

$$\lim_{j\to\infty}(X_j) = \lim_{j\to\infty}\left(\frac{T}{J}\left(\sum_{r=1}^{j-1}\alpha^r\right)\right) = \frac{T}{J}\left(\frac{\alpha}{1-\alpha}\right) \tag{6.6}$$

$$\lim_{j\to\infty}(X'_j) = \lim_{j\to\infty}\left(\frac{T}{J}\left(\sum_{r=1}^{j-1}(\alpha^r+1)\right)\right) = \frac{T}{J}\left(\frac{1}{1-\alpha}\right) \tag{6.7}$$

The equations (6.6) and (6.7) provide a useful perspective to figure out how maintenance actions affect the effective ages of a system over a long-term planning horizon. For example, suppose the length of each period is equal to one month, the planning horizon is long enough and the system is maintained every month with an improvement factor equal to 0.8, which means that each maintenance action reduces the effective age by 20%. Under these assumptions, a lower-bound for the starting and ending effective ages would be close to 4 and 5 months respectively. We can interpret that by performing this kind of maintenance starting and ending effective ages of the system will not be less than 4 and 5 months respectively. These values can be considered as the minimum for starting and ending effective ages of the system.

6.3. Mathematical Model

Many researchers assume a constant improvement factor and develop optimization models to determine an optimal schedule of preventive maintenance actions; see Jayabalan and Chaudhuri (1992), Martorell *et al.* (1996) and Martorell *et al.* (1999).

Some assume a constant improvement factor but variable amount of age reduction, which depends on when maintenance actions are performed; see Dedopoulos and Smeers (1998). Nakagawa (1988) assumes a variable improvement factor as a function of time intervals before any replacement and presents equations (6.8) and (6.9) for hazard rate improvement factor and effective age improvement factor which are also used by Lim and Park (2007).

$$a_k = \frac{2k+1}{k+1} \text{ for } k = 1,\dots,n \tag{6.8}$$

$$b_k = \frac{k}{k+1} \text{ for } k = 1,\dots,n \tag{6.9}$$

Lin *et al* (2001) consider equations (6.10) and (6.11) for the same purpose, which are also used by El-Ferik and Ben-Daya (2006) and Bartholomew-Biggs *et al* (2006).

$$a_k = \frac{6k+1}{5k+1} \text{ for } k = 1,\dots,n \tag{6.10}$$

$$b_k = \frac{k}{2k+1} \text{ for } k = 1,\dots,n \tag{6.11}$$

We present a new improvement factor model as a function of maintenance and replacement costs, and effective age of system at the end of previous period.

$$\alpha_j = \phi(R, M, X'_{j-1}) = \left(\frac{R-M}{R}\right) \cdot \left(\frac{X'_{j-1}}{X'_{j-1}+1}\right), \text{ for } j = 1,\dots,T \tag{6.12}$$

The first term is the constant coefficient based on the ratio of difference of replacement and maintenance costs, which is always between zero and one. It is designed so that if a costly maintenance action is performed on a system, the effective age improves more than when an inexpensive maintenance is performed. That is, more expensive maintenance results in a greater amount of age reduction. For example, overhauling an engine results in more age reduction that does changing the oil. Note that if maintenance cost is equal to the replacement cost, the numerator of the fraction becomes zero, and maintenance action will coincide with replacement action. On the other hand, if the maintenance cost is equal to zero, the ratio becomes one and it means that maintenance does not affect the effective age and it can be considered as do nothing. The second term is a ratio of the effective age at the end of previous period, which is always less than one. The minimum value is obtained whenever the system is replaced at the previous period. It can be seen that the ratio increases by increasing the effective age and the amount of age reduction decreases as the system ages over the planning horizon.

6.4. Computational Results

In order to show the effectiveness of the proposed improvement factor model, we apply it into the optimization models (3.19) and (3.20). We assume a system with $\lambda = 0.00025$ and $\beta = 2.20$ as the characteristic life (scale) and the shape parameters of the system and consider failure, maintenance, and replacement costs equal to \$2500, \$300, \$1500 respectively. In addition, we assume R = 92% as the required reliability for Model 1, B = \$6000 as the given budget for Model 2, and 36 months as the planning horizon.

We consider three improvement factor functions as follows:

$$\alpha_{1,j} = \phi_1(R,M) = \left(\frac{R-M}{R}\right) \tag{6.13}$$

$$\alpha_{2,j} = \phi_2(X'_{j-1}) = \left(\frac{X'_{j-1}}{X'_{j-1}+1}\right), \text{ for } j = 1,\dots,T \tag{6.14}$$

$$\alpha_{3,j} = \phi_3(R,M,X'_{j-1}) = \left(\frac{R-M}{R}\right)\cdot\left(\frac{X'_{j-1}}{X'_{j-1}+1}\right), \text{ for } j = 1,\dots,T \tag{6.15}$$

The first function calculates the improvement factor of system based on the ratio of difference of replacement and maintenance costs, which is constant over the planning horizon. The second function is a simple version of the original model that uses only the ratio of effective age at the end of previous period and the last one is the original proposed model. We employ the improvement factor functions into the single-component version of the optimization models (3.19) and (3.20) developed in Chapter 3. LINGO[1] programs of nonlinear mixed-integer optimization models with different improvement factor functions are presented in Appendix E.

We optimized the models, and obtained optimal solutions. The optimal objective function value for both models with different improvement factor functions is presented in Table 6.1. As we can see that by applying a variable improvement factor, equations (6.14) and (6.15), we can obtain lower optimal value in Model 1, minimizing total cost subject to reliability constraint, and higher optimal value in Model 2, maximizing overall reliability subject to budgetary constraint, than considering constant improvement factor function; equation (6.13). It is clear that variable improvement factor functions have an advantage over constant improvement factor in terms of objective function value in optimal solution.

Table 6.1. Optimal objective function values

Improvement Factor	Model 1		Model 2	
	Total Cost	Reliability	Reliability	Budget
Function 1	$ 8,002.54	92%	89.45%	$ 6,000
Function 2	$ 7,707.74	92%	89.66%	$ 6,000
Function 3	$ 6,506.86	92%	91.17%	$ 6,000

[1] http://www.lindo.com

Tables 6.2 and 6.3 illustrate optimal schedules based on different improvement factor functions in both models. As it could be seen that by using a variable improvement factor the optimal schedules contain more maintenance activities than replacements activities. Especially by applying the third model, the optimal schedule consists of only maintenance actions. We also plot the variation of improvement factor functions over the planning horizon; detailed computational results presented in Appendix F. Figures (6.1) and (6.2) show the variation of improvement factor functions over the planning horizon. It can be seen that the constant coefficient smoothes the second function and reduces its variability. We can state that equation (6.15) which combines maintenance and replacement costs as a constant coefficient along with effective age of the system as an independent variable can model the improvement factor variations very well and results to better optimal solution than the second function.

Finally, we can conclude that the proposed improvement factor model has an advantage over the constant improvement factor and the variable improvement factor function, which uses just the effective age variables without considering maintenance and replacement cost.

Table 6.2. Optimal maintenance and replacement schedules in Model 1

Month/ Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Function 1	-	-	-	-	-	R	-	-	-	R	-	-	-	-	M	-	-	R	-	-	-	-	-	-	R	-	-	-	R	-	-	-	-	-	-	-
Function 2	-	-	-	-	-	R	-	M	-	-	-	-	-	R	-	-	-	-	R	-	M	M	M	-	-	-	-	R	-	M	-	-	-	-	-	-
Function 3	-	-	M	-	M	M	M	M	-	-	-	-	-	-	M	M	M	-	M	M	M	-	-	-	-	-	R	-	-	-	M	-	-	-	-	-

Table 6.3. Optimal maintenance and replacement schedules in Model 2

Month/ Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Function 1	-	-	-	-	M	-	-	-	R	-	-	-	-	-	-	R	-	-	-	-	M	M	-	-	-	R	-	-	-	-	M	-	-	-	-	-
Function 2	-	M	-	-	-	-	-	-	R	-	M	M	M	-	-	-	-	-	-	R	-	-	-	-	-	-	-	R	-	-	-	-	-	-	-	-
Function 3	-	-	-	M	-	M	-	M	M	M	-	M	M	M	-	M	-	M	M	-	M	M	-	M	M	-	M	M	M	M	-	-	-	-	-	-

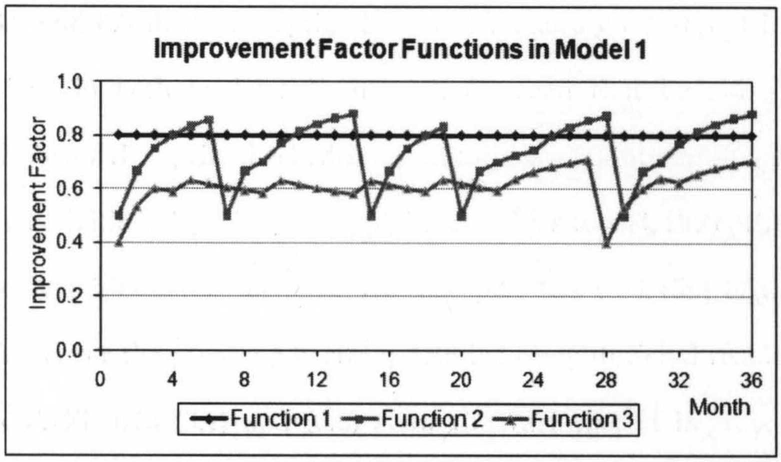

Figure 6.1. Variation of improvement factor functions in Model 1

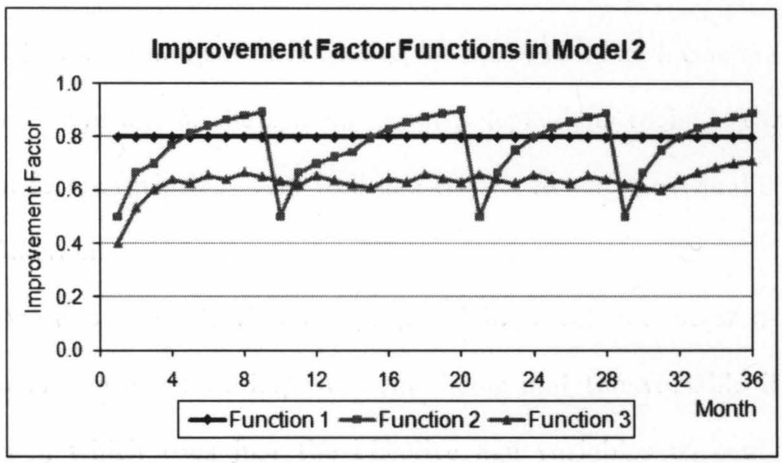

Figure 6.2. Variation of improvement factor functions in Model 2

6.5. Practical Procedure

In most practical situations, it is almost impossible to estimate effect of maintenance activities on service life of systems or even on service life of a single component. In these situations, we suggest using the following procedure.

Suppose we have two new identical repairable and maintainable systems with an increasing rate of occurrence of failure (ROCOF) over a finite planning horizon. We leave the first system to perform its operation until the end of its service life. It is clear that because of increasing failure rate the expected number of failures increase and the overall reliability of the system decreases over the planning horizon. We can fit a non-homogeneous Poisson process (NHPP) to the observed data based on increasing rate of occurrence of failure assumption; where x is the effective age of the system as shown in Figure 6.3.

$$v_1(x) = \lambda_1 \cdot \beta_1 \cdot x^{\beta_1 - 1} \tag{6.16}$$

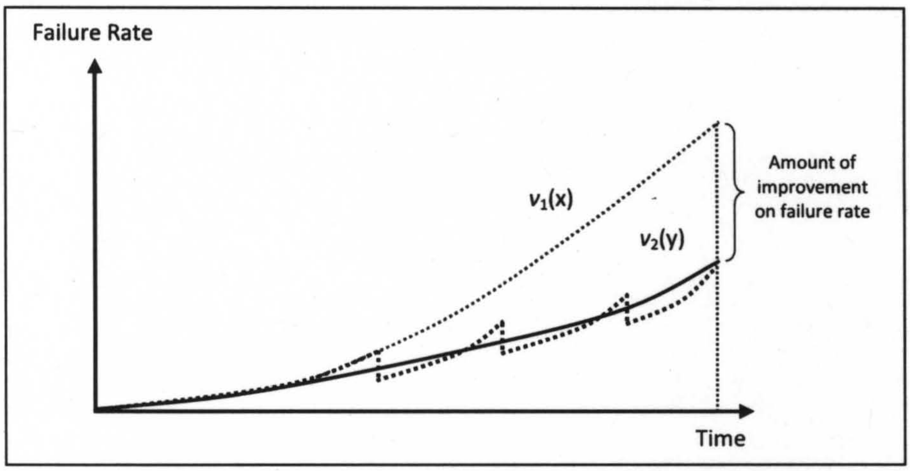

Figure 6.3. Graphical illustration of practical procedure

At the mean time, we perform regular maintenance actions on the second system. After performing each maintenance activity, failure rate of the system decreases to an unknown certain level. At the end of the planning horizon, we can compare the final failure rate of the first system in which no maintenance action was taken with the final failure rate of the second system in which regular maintenance activities were performed. We can also fit an appropriate non-homogeneous Poisson process in which y is the effective age of the system as illustrated in Figure 6.3.

$$v_2(y) = \lambda_2 \cdot \beta_2 \cdot y^{\beta_2 - 1} \tag{6.17}$$

By comparing the of failure rates, we can calculate the effect of maintenance activities and amount of improvement on failure rate of the system based on the ratio of failure rates at the end of the planning horizon; See Figure 6.3.

$$\Delta_T = \frac{v_2(y_T)}{v_1(x_T)} \tag{6.18}$$

Now, we can define amount of improvement factor for the effective age of the systems as follows:

$$\alpha_T = \frac{y_T}{x_T} \tag{6.19}$$

$$\Delta_T = \frac{\lambda_2 \cdot \beta_2 \cdot y_T^{\beta_2 - 1}}{\lambda_1 \cdot \beta_1 \cdot x_T^{\beta_1 - 1}} = \left(\frac{\lambda_2 \cdot \beta_2}{\lambda_1 \cdot \beta_1}\right) \cdot \left(\frac{y_T}{x_T}\right)^{\beta_2 - \beta_1} = \left(\frac{\lambda_2 \cdot \beta_2}{\lambda_1 \cdot \beta_1}\right) \cdot \alpha_T^{\beta_2 - \beta_1} \tag{6.20}$$

$$\alpha_T = \sqrt[\beta_2 - \beta_1]{\left(\frac{\lambda_2 \cdot \beta_2}{\lambda_1 \cdot \beta_1}\right) \cdot \Delta_T} \tag{6.21}$$

Finally, we recommend the equation (6.22) as an estimation of improvement factor for each single maintenance action during the service life of the system.

$$\alpha = \sqrt[\beta_2 - \beta_1]{\left(\frac{\lambda_2 \cdot \beta_2}{\lambda_1 \cdot \beta_1}\right) \cdot \Delta_T} \tag{6.22}$$

6.6. Chapter Summary

In this chapter, we reviewed current improvement factor function applied in maintenance scheduling optimization models. We developed and proved mathematical equations to determine a lower-bound for effective age of maintainable and repairable system in a long-term planning horizon. A new improvement factor model was presented and analyzed by the computational results of optimization models and advantage of it over other models was shown.

CHAPTER 7

CONCLUSION AND FUTURE RESEARCH

7.1. Conclusion of the Research

In this dissertation, we clearly defined a general preventive maintenance and replacement scheduling problem. The aim of solving this problem is to improve the overall reliability and availability of a system and to reduce total cost of its maintenance. We addressed the problem using a multi-objective approach. We reviewed and critiqued the recent literature and mentioned that most studies relied on modeling and analysis of single-component single-objective systems. We defined characteristics of a repairable and maintainable system and developed new optimization models to find optimal preventive maintenance and replacement schedules in multi-component systems. These models also provide a general framework to achieve optimal preventive maintenance and replacement policies and, with modifications, can be used as basic closed-form models for any type of system. Our solution methodology to solve the nonlinear mixed integer programming models allowed us to obtain optimal solutions.

As an extension, we considered engineering economy parameters and constructed a multi-objective optimization model. Due to the including of multiple objectives and its nonlinear structure of the model and the use of integer decision variables, we decided to solve the model using multi-objective metaheuristic algorithms. We applied two types of genetic algorithms and a simulated annealing algorithm, solved the problem, and achieved near optimal solutions to construct the trade-off curves. We also performed sensitivity analysis on parameters of the optimization models and compared computational performance and effectiveness of exact and metaheuristic algorithms for set of problems.

In order to determine and calculate improvement factor parameter used in optimization models, we presented and analyzed a new mathematical function to model age reduction and improvement factor parameter for repairable and maintainable components.

7.2. Direction for Future Research

We considered two main criteria in our models, total cost to be minimized and overall reliability to be maximized. An extension of these models would be considering other criteria such as system availability and demand satisfaction, which make the models more practical but very complicated to solve.

All of our models are classified as NP-hard problems in which there is no polynomial computational time for solving large-scale problems. We recommend applying other heuristic and metaheuristic algorithms to find optimal or near optimal solutions, especially for multi-objective models with more than two objective functions.

We recommend using discrete-event and continuous simulation models and integrating them into our optimization models in order to handle real situations, in which unexpected failures occur between intervals. In this situation, one can re-optimize the models and obtain a new optimal preventive maintenance and replacement schedule for the rest of the planning horizon. This approach combines prescriptive nature of optimization models with descriptive nature of simulation models and develop a complete feedback cycle of modeling in which optimization and simulation models interact with each other.

We also intend to extend our models into specific applications, especially production planning and scheduling which is introduced by some researchers. Because of our proposed modeling approach, in which we define parameters, decision variables, objective functions, and constraints of system, our models can be integrated with production planning and inventory control models.

We recommend using Monte Carlo simulation to model age reduction and improvement factor parameters into the optimization model. Finally, we would like to encourage prospective researchers to develop more advanced procedures to estimate age reduction and improvement factors for practical situations especially in health care applications and medical operations.

REFERENCES

[1] Adzakpa, K.P., Adjallah, K.H., Yalaoui, F., (2004) On-line maintenance job scheduling and assignment to resources in distributed systems by heuristic-based optimization, *Journal of Intelligent Manufacturing*, v 15, n 2, April 2004, p 131-140.

[2] Alardhi, M., Hannam, R.G., Labib, A.W., (2007) Preventive maintenance scheduling for multi-cogeneration plants with production constraints, *Journal of Quality in Maintenance Engineering*, v 13, n 3, 2007, p 276-92.

[3] Alfares, H.K., (2002) Developing optimum preventive maintenance schedules using simulation: A case study, *International Journal of Industrial Engineering: Theory, Applications and Practice*, v 9, n 3, September 2002, p 311-318.

[4] Allaoui, H., Artiba, A., (2004) Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints, *Computers and Industrial Engineering*, v 47, n 4, December 2004, p 431-450.

[5] Ascher, H.E., Feingold, H., (1984) *Repairable Systems Reliability: Modeling, Inference, Misconceptions, and Their Causes*, Published by M. Dekker, 1984.

[6] Bartholomew-Biggs, M., Christianson, B., Zuo, M., (2006) Optimizing preventive maintenance models, *Computational Optimization and Applications*, v 35, n 2, October 2006, p 261-279.

[7] Billinton, R., Pan, J., (2000) Application of Monte Carlo simulation to optimal maintenance scheduling in a parallel-redundant system, *IEEE Proceedings-Generation, Transmission and Distribution*, v 147, n 5, September 2000, p 274-278.

[8] Bottazzi, A., Dubi, A., Gandini, A., Goldfeld, A., Righini, R., Simonot, H., (1992) Improving the preventive maintenance of a bus yard by a Monte Carlo simulation method, *Safety and Reliability '92*, 10-12 June 1992 , Copenhagen, Denmark, p 883-95.

[9] Bris, R., Chatelet, E., Yalaoui, F., (2003) New method to minimize the preventive maintenance cost of series-parallel systems, *Reliability Engineering and System Safety*, v 82, n 3, December 2003, p 247-55.

[10] Budai, G., Huisman, D., Dekker, R., (2006) Scheduling preventive railway maintenance activities, *Journal of the Operational Research Society*, v 57, n 9, September 2006, p 1035-44.

[11] Burton, J.S., Banerjee, A., Sylla, C., (1989) A simulation study of sequencing and maintenance decisions in a dynamic job shop, *Computers and Industrial Engineering*, v 17, 1989, p 447-452.

[12] Canfield, R.V., (1986) Cost optimization of periodic preventive maintenance, *IEEE Transactions on Reliability*, v R-35, n 1, April 1986, p 78-81.

[13] Canto, S.V., (2006) Application of Benders' decomposition to power plant preventive maintenance scheduling, *European Journal of Operational Research*, v 184, Issue 2, January 2008, p 759-777.

[14] Cassady, C.R., Bowden, R.O., Liew, L., Pohl, E.A., (1999) Combining preventive maintenance and statistical process control: a preliminary investigation, *IIE Transactions*, v 32, n 6, June 2000, 471-478.

[15] Cassady, C.R., and Kutanoglu, E., (2005) Integrating preventive maintenance planning and production scheduling for a single machine, *IEEE Transactions on Reliability*, v 54, n 2, June 2005, p 304-309.

[16] Cavory, G., Dupas, R., Goncalves, G., (2001) A genetic approach to the scheduling of preventive maintenance tasks on a single product manufacturing production line, *International Journal of Production Economics*, v 74, n 1-3, December 2001, p 135-46.

[17] Chan, F.T.S., (2001) Simulation analysis of maintenance policies in a flow line production system, *International Journal of Computer Applications in Technology*, v 14, n 1-3, 2001, p 78-86.

[18] Chareonsuk, C., Nagarur, N., Tabucanon, M.T., (1997) A multi-criteria approach to the selection of preventive maintenance intervals, *International Journal of Production Economics*, v 49, n 1, March 1997, p 55-64.

[19] Charles, A.S., Floru, I.R., Azzaro-Pantel, C., Pibouleau, L., Domenech, S., (2003) Optimization of preventive maintenance strategies in a multipurpose batch plant: application to semiconductor manufacturing, *Computers and Chemical Engineering*, v 27, n 4, April 2003, p 449-467.

[20] Che-Hua, L., (2007) Preventive maintenance scheduling via particle swarm optimization method, *International Multi Conference of Engineers and Computer Scientists*, 21-23 March 2007, Kowloon, China, v 1, p 121-126.

[21] Cheng, C.Y., Chen, M.C., (2003) The periodic preventive maintenance policy for deteriorating systems by using improvement factor model, *International Journal Applied Science and Engineering*, v1, n2, 2003, p 114-122.

[22] Cheng, C.Y., Chen, M., Guo, R., (2007) The optimal periodic preventive maintenance policy with degradation rate reduction under reliability limit, *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, 2-4 December 2007, Singapore, Singapore, p 649-653.

[23] Cohon, J.L., (1978) *Multi-objective Programming and Planning*, New York: Academic Press.

[24] Dedopoulos, I.T., Smeers, Y., (1998) Age reduction approach for finite horizon optimization of preventive maintenance for single units subject to random failures, *Computers and Industrial Engineering*, v 34, n 3, July 1998, p 643-654.

[25] Devulapalli, S., Martinez, J.C., De la Garza, J.M., (2002) Evaluation of policies for the maintenance of bridges using discrete-event simulation, *Proceedings of the 2002 Winter Simulation Conference*, 8-11 December 2002, San Diego, CA, USA, p 1759-1764.

[26] Duarte, J.A.C., Craveiro, J.C.T.A., Trigo, T.P., (2006) Optimization of the preventive maintenance plan of a series components system, *International Journal of Pressure Vessels and Piping*, v 83, n 4, April 2006, p 244-248.

[27] Duffuaa, S.O., Ben-Daya, M., Al-Sultan, K.S., Andijani, A.A., (2001) A generic conceptual simulation model for maintenance systems, *Journal of Quality in Maintenance Engineering*, v 7, n 3, 2001, p 207-219.

[28] El-Ferik, S., Ben-Daya, M., (2006) Age-based hybrid model for imperfect preventive maintenance, *IIE Transactions*, v 38, n 4, April 2006, p 365-375.

[29] Fard, N.S., Nukala, S., (2004) Preventive maintenance scheduling for repairable systems, *IIE Annual Conference and Exhibition 2004*, 15-19 May 2004, Houston, TX, USA, p 145-150.

[30] Gembicki, F.W., (1974) *Vector Optimization for Control with Performance and Parameter Sensitivity Indices*, Ph.D. Thesis, Case Western Reserve University., Cleveland, Ohio, USA.

[31] Goel, A.L., Nanda, P.S., D'Sa, N., (1973) A statistical analysis of maintenance system simulation, *Proceedings of the 1973 Annual Reliability and Maintainability Symposium*, 23-25 January 1973 , Philadelphia, PA, USA, p 125-135.

[32] Goldberg, D., (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing, Reading, MA, USA.

[33] Greasley, A., (2000) Using simulation to assess the service reliability of a train maintenance depot, *Quality and Reliability Engineering International*, v 16, n 3, May 2000, p 221-228.

[34] Hagmark, P.E., Virtanen, S., (2007) Simulation and calculation of reliability performance and maintenance costs, *2007 Annual Reliability and Maintainability Symposium*, 22-25 January 2007, Orlando, FL, USA.

[35] Han, B.J., Fan, X.M., Ma, D.Z., (2004) Optimization of preventive maintenance policy of manufacturing equipment based on simulation, *Computer Integrated Manufacturing Systems*, v 10, n 7, July 2004, p 853-857.

[36] Han, B.J., Fan, X.M., Ma, D.Z., Jin, Y., (2003) Optimization of equipment preventive maintenance cycle in production system based on genetic algorithm, *Computer Integrated Manufacturing Systems*, v 9, n 3, March 2003, p 206-209.

[37] Han, B.J., Pan, J., Fan, X.M., Ma, D.Z., (2004) Optimization of preventive maintenance scheduling for production machine of production system in finite time horizon, *Journal of Dong Hua University* (English Edition), v 21, n 1, February 2004, p 112-116.

[38] Holland, J.H., (1975), *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, USA.

[39] Houshyar, A., Jafri, A., Naiksatam, V., (2003) Comparison of different maintenance policies using simulation, *Proceedings of the IASTED International Conference on Modeling, Simulation and Optimization*, 2-4 July 2003, Banff, Alberta, Canada, p 259-265.

[40] Hsu, L.F., (1991) Optimal preventive maintenance policies in a serial production system, *International Journal of Production Research*, v 29, n 12, December 1991, p 2543-2555.

[41] Jayabalan, V., Chaudhuri, D., (1992) Sequential imperfect preventive maintenance policies: a case study, *Microelectronics and Reliability*, v 32, n 9, September 1992, p 1223-1229.

[42] Jayabalan, V., Chaudhuri, D., (1992) Cost optimization of maintenance scheduling for a system with assured reliability, *IEEE Transactions on Reliability*, v 41, n 1, March 1992, p 21-25.

[43] Jaturonnatee, J., Murthy, D.N.P., Boondiskulchok, R., (2006) Optimal preventive maintenance of leased equipment with corrective minimal repairs, *European Journal of Operational Research*, v 174, n 1, October 2006, p 201-215.

[44] Jayakumar, A., Asagarpoor, S., (2004) Maintenance optimization of equipment by linear programming, *International Conference on Probabilistic Methods Applied to Power Systems*, 12-16 September 2004, p 145-149.

[45] Jin, Y.L., Jiang, Z.H., Hou, W.R., (2006) Optimal policy research on reliability-centered preventive maintenance for multi-components equipment, *Journal of Shanghai Jiaotong University*, v 40, n 12, December 2006, p 2051-2056.

[46] Joe, J.S, Levary, R.R., Ferris, M.E., (1997) Planning preventive maintenance for a fleet of police vehicles using simulation, *Journal of Simulation*, v 68, n 2, February 1997, p 93-100.

[47] Kim, H., Nara, K., Gen, M., (1994) A method for maintenance scheduling using GA combined with SA, *Computers and Industrial Engineering*, v 27, September 1994, p 477-480.

[48] Konak, A., Coit, D.W., Smith, A.E., (2006) Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering and System Safety*, v 91, n 9, September 2006, 992-1007.

[49] Kralj, B., Petrovic, R., (1995) A multi-objective optimization approach to thermal generation units maintenance scheduling, *European Journal of Operational Research*, v 84, n 2, July 1995, p 481-93.

[50] Krishnan, B.N.T., (1992) A simulation model for maintenance planning, *Proceedings of the Annual Reliability and Maintainability Symposium*, March 1992, Las Vegas, NV, USA, p 109-118.

[51] Kuo, Y., Chang, Z.A., (2007) Integrated production scheduling and preventive maintenance planning for a single machine under a cumulative damage failure process, *Naval Research Logistics*, v 54, n 6, September 2007, p 602-614.

[52] Kuo, W., Prasad, V.R., Tillman, F.A., Hwang, C.L., (2001) *Optimal Reliability Design: Fundamentals and Applications*, Cambridge University Press, 2001.

[53] Lapa, C.M.F., Pereira, C.M.N.A., De Barros, M.P., (2006) A model for preventive maintenance planning by genetic algorithms based in cost and reliability, *Reliability Engineering and System Safety*, v 91, n 2, February 2006, p 233-240.

[54] Leng, K., Ren, P., Gao, L., (2006) A novel approach to integrated preventive maintenance planning and production scheduling for a single machine using the chaotic particle swarm optimization algorithm, *Proceedings of the 6th World Congress on Intelligent Control and Automation*, 21-23 June 2006, Dalian, China, p 7816-7820.

[55] Leou, R.C., (2003) A new method for unit maintenance scheduling based on genetic algorithm, *2003 IEEE Power Engineering Society General Meeting*, 13-17 July 2003, Toronto, Ontario, Canada, v 1, p 246-251.

[56] Levitin, G., Lisnianski, A., (2000) Optimization of imperfect preventive maintenance for multi-state systems, *Reliability Engineering and System Safety*, v 67, n 2, February 2000, p 193-209.

[57] Levitin, G., Lisnianski, A., (2000) Optimal replacement scheduling in multi-state series-parallel systems, *Quality and Reliability Engineering International*, v 16, n 2, March 2000, p 157-162.

[58] Li, H., Qian, X.S., (2005) Optimizing of preventive maintenance scheduling for cluster tools of semiconductor manufacturing, *Semiconductor Technology*, v 30, n 11, 2005, p 39-42.

[59] Li, W., Zuo, M.J., (2007) Joint optimization of inventory control and maintenance policy, *2007 Annual Reliability and Maintainability Symposium*, 22-25 January 2007, Orlando, FL, USA.

[60] Lim, J.H., Park, D.H., (2007) Optimal periodic preventive maintenance schedules with improvement factors depending on number of preventive maintenances, *Asia-Pacific Journal of Operational Research*, v 24, n 1, February 2007, p 111-124.

[61] Limbourg, P., Kochs, H.D., (2006) Preventive maintenance scheduling by variable dimension evolutionary algorithms, *International Journal of Pressure Vessels and Piping*, v 83, n 4, April 2006, p 262-269.

[62] Lin, D., Zuo, M.J., Yam, R.C.M., (2001) Sequential imperfect preventive maintenance models with two categories of failure modes, *Naval Research Logistics*, v 48, n 2, March 2001, p 172-183.

[63] *Lingo 11 User's Guide*, published by LINDO System Inc, 2008.

[64] Lisnianski, A., Levetin, G., (2003) *Multi-State System Reliability, Assessment, Optimization and Applications*, Published by World Scientific, In series on Quality, Reliability, and Engineering Statistics, NJ, USA.

[65] Malik, M.A.K., (1979) Reliable preventive maintenance scheduling, *AIIE Transactions*, v 11, n 3, September 1979, p 221-228.

[66] Marquez, A.C., Gupta, J.N.D., Ignizio, J.P., (2006) Improving preventive maintenance scheduling in semiconductor fabrication facilities, *Production Planning and Control*, v 17, n 7, October 2006, p 742-754.

[67] Marseguerra, M., Zio, E., Podofillini, L., (2002) Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation, *Reliability Engineering and System Safety*, v 77, n 2, July 2002, p 151-165.

[68] Martin, P., (1988) Optimal preventive maintenance in a health-care facility, *Electrical Construction and Maintenance*, v 87, n 1, January 1988, p 67-71.

[69] Martorell, S., Munoz, A., Serradell, V., (1996) Age-dependent models for evaluating risks & costs of surveillance & maintenance of components, *IEEE Transactions on Reliability*, v 45, n 3, September 1996, p 433-441.

[70] Martorell, S., Sanchez, A., Serradell, V., (1999) Age-dependent reliability model considering effects of maintenance and working conditions, *Reliability Engineering and System Safety*, v 64, n 1, April 1999, p 19-31.

[71] Mathew, J., Rajendran, C., (1993) Scheduling of maintenance activities in a sugar industry using simulation, *Computers in Industry*, v 21, n 3, April 1993, p 331-334.

[72] McClymonds, S.L., Winge, D.E., (1987) Optimization of nuclear plant preventive maintenance, *Transactions of the American Nuclear Society*, v 54, s.1, 1987, p 10-12.

[73] Mohamed-Salah, O., Daoud, A.K., Ali, G., (1999) Simulation model for opportunistic maintenance strategies, *Proceedings of the 1999 7th IEEE International Conference on Emerging Technologies and Factory Automation* (ETFA'99), 18-21 October 1999, Barcelona, Spain, p 703-708.

[74] Nakagawa, T., (1988) Sequential imperfect preventive maintenance policies, *IEEE Transactions on Reliability*, v 37, n 3, August 1988, p 295-298.

[75] Panagiotidou, S., Tagaras, G., (2007) Optimal preventive maintenance for equipment with two quality states and general failure time distributions, *European Journal of Operational Research*, v 180, n 1, July 2007, p 329-353.

[76] Paz, N.M., Leigh, W., Rogers, R.V., (1994) The development of knowledge for maintenance management using simulation, *IEEE Transactions on Systems, Man and Cybernetics*, v 24, n 4, April 1994, p 574-93.

[77] Pongpech, J., Murthy, D.N.P., Boondiskulchock, R., (2006) Maintenance strategies for used equipment under lease, *Journal of Quality in Maintenance Engineering*, v 12, n 1, 2006, 52-67.

[78] Quan, G., Greenwood, G.W., Liu, D., Hu, S., (2007) Searching for multi-objective preventive maintenance schedules: Combining preferences with evolutionary algorithms, *European Journal of Operational Research*, v 177, n 3, March 2007, p 1969-1984.

[79] Rezg, N, Chelbi, A, Xie, X., (2005) Modeling and optimizing a joint inventory control and preventive maintenance strategy for a randomly failing production unit: Analytical and simulation approaches, *International Journal of Computer Integrated Manufacturing*, v 18, n 2-3, March and May 2005, p 225-235.

[80] Rezg, N., Xie, X., Mati, Y., (2004) Joint optimization of preventive maintenance and inventory control a production line using simulation, *International Journal of Production Research*, v 42, n 10, May 2004, p 2029-2046.

[81] Robelin, C.A., Madanat, S.M., (2006) Dynamic programming based maintenance and replacement optimization for bridge decks using history-dependent deterioration models, *Applications of Advanced Technology in Transportation - Proceedings of the Ninth International Conference on Applications of Advanced Technology in Transportation*, 13-16 August 2006, Chicago, IL, USA, p 13-18.

[82] Ruiz, R., Garcia-Diaz, J.C., Maroto, C., (2007) Considering scheduling and preventive maintenance in the flowshop sequencing problem, *Computers and Operations Research*, v 34, n 11, November 2007, p 3314-3330.

[83] Samrout, M., Benouhiba, T., Chatelet, E., Yalaoui, F., (2006) Preventive maintenance optimization using evolutionary hybrid algorithm, 2006 *International Conference on Service Systems and Service Management*, 25-27 October 2006, Troyes, France.

[84] Samrout, M., Yalaoui, F., Chatelet, E., Chebbo, N., (2005) New methods to minimize the preventive maintenance cost of series-parallel systems using ant colony optimization, *Reliability Engineering and System Safety*, v 89, n 3, September 2005, p 346-354.

[85] Savsar, M., (1997) Simulation analysis of maintenance policies in just-in-time production systems, *International Journal of Operations and Production Management*, v 17, n 3, 1997, p 256-266.

[86] Sawhney, R., Martinis, P., Murthi, V., (2004) Simulation based approach for determining maintenance strategies, *International Journal of COMADEM*, v 7, n 3, July 2004, p 32-41.

[87] Schrage, L., (2006) *Optimization Modeling with Lingo*, 6th edition, published by LINDO System Inc, Chicago, USA, 2006.

[88] Shalaby, M.A., Gomaa, A.H., Mohib, A.M., (2004) A genetic algorithm for preventive maintenance scheduling in a multiunit multistate system, *Journal of Engineering and Applied Science*, v 51, n 4, August 2004, p 795-811.

[89] Shirmohammadi, A.H., Zhang, Z.G., Love, E., (2007) A computational model for determining the optimal preventive maintenance policy with random breakdowns and imperfect repairs, *IEEE Transactions on Reliability*, v 56, n 2, June 2007, p 332-339.

[90] Shum, Y.S., Gong, D.C., (2007) The application of genetic algorithm in the development of preventive maintenance analytic model, *International Journal of Advanced Manufacturing Technology*, v 32, n 1-2, 2007, p 169-183.

[91] Sortrakul, N., Nachtmann, H.L., Cassady, C.R., (2005) Genetic algorithms for integrated preventive maintenance planning and production scheduling for a single machine, *Computers in Industry*, v 56, n 2, February 2005, p 161-168.

[92] Suresh, K., Kumarappan, N., (2006) Combined genetic algorithm and simulated annealing for preventive unit maintenance scheduling in power system, *2006 IEEE Power Engineering Society General Meeting*, PES, 18-22 June 2006, Montreal, Quebec, Canada.

[93] Taboada, H.A., Espiritu, J.F., Coit, D.W., (2008) MOMS-GA: A multi-objective multi-state genetic algorithm for system reliability optimization design problems, *IEEE Transactions on Reliability*, v 57, n 1, March 2008, p 182-191.

[94] Tam, A.S.B., Chan, W.M., Price, J.W.H., (2006) Optimal maintenance intervals for multi-component system, *Production Planning and Control*, v 17, n 8. December 2006, p 769-779.

[95] Tan, J.S., Kramer, M.A., (1997) A general framework for preventive maintenance optimization in chemical process operations, *Computers and Chemical Engineering*, v 21, n 12, 1997, p 1451-1469.

[96] Tsai, Y.T., Wang, K.S., Teng, H.Y., (2001) Optimizing preventive maintenance for mechanical components using genetic algorithms, *Reliability Engineering and System Safety*, v 74, n 1, October 2001, p 89-97.

[97] Usher, J.S., Kamal, A.H., Hashmi, S.W., (1998) Cost optimal preventive maintenance and replacement scheduling, *IIE Transactions*, v 30, n 12, December 1998, p 1121-1128.

[98] Verma, A.K, Ramesh, P.G., (2007) Multi-objective initial preventive maintenance scheduling for large engineering plants, *International Journal of Reliability, Quality and Safety Engineering*, v 14, n 3, June 2007, p 241-250.

[99] Wang, Y., Handschin, E., (2000) A new genetic algorithm for preventive unit maintenance scheduling of power systems, *International Journal of Electrical Power and Energy Systems*, v 22, n 5, June 2000, p 343-348.

[100] Westman, J.J., Hanson, F.B., (2000) Manufacturing production scheduling with preventive maintenance in random environments, *Proceedings of the 2000 IEEE International Conference on Control Applications*, 25-27 September 2000, Anchorage, USA, p 582-587.

[101] Westman, J.J., Hanson, F.B., Boukas, E.K., (2001) Optimal production scheduling for manufacturing systems with preventive maintenance in an uncertain environment, *Proceedings of American Control Conference*, 25-27 June 2001 , Arlington, VA, USA, p 1375-1380 vol.2.

[102] Whitley D., (1989), The Genitor algorithm and selective pressure: Rank best allocation of reproductive trials is best proceeding, *Third International Conference on Genetic Algorithms*, p 116-121.

[103] Xi, L.F., Zhou, X.J., Lee, J., (2005) Research on sequential preventive maintenance policy in finite time horizon, *Computer Integrated Manufacturing Systems*, v 11, n 10, October 2005, p 1465-1468.

[104] Yao, X, Fernandez-Gaucherand, E., Fu, M.C., Marcus, S.I., (2004) Optimal preventive maintenance scheduling in semiconductor manufacturing, *IEEE Transactions on Semiconductor Manufacturing*, v 17, n 3, August 2004, p 345-356.

[105] Yao, X., Fu, M., Marcus, S.I., Fernandez-Gaucherand, E., (2001) Optimization of preventive maintenance scheduling for semiconductor manufacturing systems: models and implementation, *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01)*, 5-7 September 2001, Mexico City, Mexico, p 407-411.

[106] Yin, X., Wen, X., Qian, Y., Yang Y., (2007) Modeling and simulation of complex maintenance system dynamics, *Proceedings of the 26th Chinese Control Conference*, 26-31 July 2007, Hunan, China, p 487-491.

[107] Ying, B., Dan, B., Zhang, H., (2005) Optimization model by integrating preventive maintenance planning and production scheduling, *Chinese Journal of Mechanical Engineering*, v 41, n 3, March 2005, p 226-228.

[108] Zhao, Y., Volovoi, V, Waters, M., Mavris, D., (2005) A sequential approach for gas turbine power plant preventive maintenance scheduling, *2005 ASME Power Conference*, 5-7 April 2005, Chicago, IL, USA, p 353-362.

[109] Zhou, B.H., Jiang, S.Y., Wang, S.J., Wu, B., Xi, L.F., (2007) Integrated production and preventive maintenance scheduling algorithm for flow shops, *Journal of Dalian Maritime University*, v 33, n 3, August 2007, p 32-35.

[110] Zhou, X.J., Xi, L.F., Lee, J., (2005) Reliability-based sequential preventive maintenance model, *Journal of Shanghai Jiaotong University*, v 39, n 12, December 2005, p 2044-2047.

APPENDIX A

LINGO PROGRAMS OF NLMIP MODELS

```
! Model 1-Nonlinear mixed integer optimization model that
 minimizes total cost subject to a reliability constraint;

Model:
Data:
    C = 10;
    T = 36;
    L = 1;
Enddata

Sets:
    Component/1..C/: Lambda, Beta, Alpha, Failure_Cost, M_Cost, R_Cost;
    Period/1..T/;
    LinkComPer(Component, Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00022 0.00035 0.00038 0.00034 0.00032 0.00028 0.00015
             0.00012 0.00025 0.00020;
    Beta = 2.20 2.00 2.05 1.90 1.75 2.10 2.25 1.80 1.85 2.15;
    Alpha = 0.62 0.58 0.55 0.50 0.48 0.65 0.75 0.68 0.52 0.67;
    Failure_Cost = 250 240 270 210 220 280 200 225 215 255;
    M_Cost = 35 32 65 42 50 38 45 30 48 55;
    R_Cost = 200 210 245 180 205 235 175 215 210 250;
    Fixed_Cost = 800;
    Given_Reliability = 0.5;
Enddata

! Objective Function, Minimizing the total cost;
    Min = @Sum(LinkComPer(i,j): (Failure_Cost(i) * Lambda(i) *
          ((XP(i,j)^Beta(i)) - (X(i,j)^Beta(i)))) + M_Cost(i) * M(i,j) +
          R_Cost(i) * R(i,j)) + @Sum(Period(j): Fixed_Cost * (1 -
          @Prod(Component(i): (1 - M(i,j) - R(i,j)))));

! Constraints;
    ! Recursive functions;
    @For(Component(i): X(i,1) = 0);
    @For(LinkComPer(i,j): XP(i,j) = X(i,j) + (L));
    @For(LinkComPer(i,j)| j #GE# 2: X(i,j) = ((1-M(i,j-1)) * (1-R(i,j-
        1)) * (XP(i,j-1)) + M(i,j-1) * Alpha (i) * (XP(i,j-1))));

    ! Basic constraints;
    @For(LinkComPer(i,j): M(i,j) + R(i,j) <= 1);
    @For(LinkComPer(i,j): @BIN(M));
    @For(LinkComPer(i,j): @BIN(R));

    ! Reliability constraint for series system of components;
    @Exp(@Sum(LinkComPer(i,j): (-Lambda(i) * ((XP(i,j)^Beta(i)) -
        (X(i,j)^Beta(i)))))) >= Given_Reliability;
End
```

```
    ! Model 2-Nonlinear mixed integer optimization model that
 maximizes overall reliability subject to a budgetary constraint;

Model:
Data:
    C = 10;
    T = 36;
    L = 1;
Enddata

Sets:
    Component/1..C/: Lambda, Beta, Alpha, Failure_Cost, M_Cost, R_Cost;
    Period/1..T/;
    LinkComPer(Component, Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00022 0.00035 0.00038 0.00034 0.00032 0.00028 0.00015
             0.00012 0.00025 0.00020;
    Beta = 2.20 2.00 2.05 1.90 1.75 2.10 2.25 1.80 1.85 2.15;
    Alpha = 0.62 0.58 0.55 0.50 0.48 0.65 0.75 0.68 0.52 0.67;
    Failure_Cost = 250 240 270 210 220 280 200 225 215 255;
    M_Cost = 35 32 65 42 50 38 45 30 48 55;
    R_Cost = 200 210 245 180 205 235 175 215 210 250;
    Fixed_Cost = 800;
    Given_Budget = 15000;
Enddata

! Objective Function, Maximizing the Reliability of series system of
components;
    Max = @Exp(@Sum(LinkComPer(i,j): (-Lambda(i) * ((XP(i,j)^Beta(i)) -
          (X(i,j)^Beta(i)))))) ;

! Constraints;
    ! Recursive functions;
    @For(Component(i): X(i,1) = 0);
    @For(LinkComPer(i,j): XP(i,j) = X(i,j) + (L));
    @For(LinkComPer(i,j)| j #GE# 2: X(i,j) = ((1-M(i,j-1)) * (1-R(i,j-
        1)) * (XP(i,j-1)) + M(i,j-1) * Alpha (i) * (XP(i,j-1))));

    ! Basic constraints;
    @For(LinkComPer(i,j): M(i,j) + R(i,j) <= 1);
    @For(LinkComPer(i,j): @BIN(M));
    @For(LinkComPer(i,j): @BIN(R));

    ! Budget constraint;
    @Sum(LinkComPer(i,j): (Failure_Cost(i) * Lambda(i) *
         ((XP(i,j)^Beta(i)) - (X(i,j)^Beta(i)))) + M_Cost(i) * M(i,j) +
         R_Cost(i) * R(i,j)) + @Sum(Period(j): Fixed_Cost * (1 -
         @Prod(Component(i): (1 - M(i,j) - R(i,j)))))<= Given_Budget ;
End
```

APPENDIX B

COMPUTATIONAL RESULTS OF NLMIP MODELS

Table B.1. Expected number of failures of components in each period in Model 1

(M,C)	1	2	3	4	5	6	7	8	9	10
1	0.00022	0.00035	0.00038	0.00034	0.00032	0.00028	0.00015	0.00012	0.00025	0.00020
2	0.00079	0.00105	0.00119	0.00093	0.00076	0.00092	0.00056	0.00030	0.00065	0.00069
3	0.00146	0.00175	0.00204	0.00147	0.00111	0.00161	0.00106	0.00045	0.00101	0.00123
4	0.00218	0.00245	0.00290	0.00199	0.00143	0.00233	0.00162	0.00059	0.00134	0.00182
5	0.00294	0.00315	0.00378	0.00250	0.00173	0.00308	0.00221	0.00072	0.00166	0.00243
6	0.00022	0.00385	0.00269	0.00174	0.00201	0.00252	0.00284	0.00084	0.00121	0.00305
7	0.00079	0.00035	0.00038	0.00134	0.00140	0.00216	0.00015	0.00073	0.00153	0.00020
8	0.00146	0.00105	0.00119	0.00187	0.00169	0.00290	0.00056	0.00085	0.00185	0.00069
9	0.00218	0.00175	0.00204	0.00238	0.00198	0.00365	0.00106	0.00097	0.00215	0.00123
10	0.00294	0.00245	0.00290	0.00287	0.00225	0.00442	0.00162	0.00109	0.00245	0.00182
11	0.00374	0.00315	0.00378	0.00336	0.00251	0.00521	0.00221	0.00121	0.00274	0.00243
12	0.00022	0.00035	0.00269	0.00034	0.00167	0.00028	0.00015	0.00132	0.00181	0.00020
13	0.00079	0.00105	0.00356	0.00093	0.00195	0.00092	0.00056	0.00143	0.00211	0.00069
14	0.00146	0.00175	0.00444	0.00147	0.00222	0.00161	0.00106	0.00153	0.00241	0.00123
15	0.00218	0.00245	0.00534	0.00199	0.00248	0.00233	0.00162	0.00164	0.00270	0.00182
16	0.00294	0.00315	0.00624	0.00250	0.00273	0.00308	0.00221	0.00174	0.00299	0.00243
17	0.00374	0.00385	0.00714	0.00300	0.00298	0.00384	0.00284	0.00184	0.00327	0.00305
18	0.00022	0.00035	0.00038	0.00034	0.00032	0.00028	0.00015	0.00012	0.00025	0.00370
19	0.00079	0.00105	0.00119	0.00093	0.00076	0.00092	0.00056	0.00030	0.00065	0.00436
20	0.00146	0.00175	0.00204	0.00147	0.00111	0.00161	0.00106	0.00045	0.00101	0.00504
21	0.00136	0.00157	0.00290	0.00120	0.00143	0.00158	0.00162	0.00059	0.00085	0.00020
22	0.00207	0.00227	0.00378	0.00174	0.00173	0.00230	0.00221	0.00072	0.00120	0.00069
23	0.00283	0.00297	0.00467	0.00225	0.00201	0.00304	0.00284	0.00084	0.00152	0.00123
24	0.00363	0.00367	0.00556	0.00275	0.00228	0.00380	0.00350	0.00097	0.00183	0.00182
25	0.00022	0.00035	0.00038	0.00187	0.00154	0.00028	0.00015	0.00081	0.00131	0.00163
26	0.00079	0.00105	0.00119	0.00238	0.00183	0.00092	0.00056	0.00094	0.00163	0.00223
27	0.00146	0.00175	0.00204	0.00287	0.00211	0.00161	0.00106	0.00105	0.00194	0.00285
28	0.00218	0.00245	0.00290	0.00336	0.00237	0.00233	0.00162	0.00117	0.00224	0.00349
29	0.00294	0.00315	0.00378	0.00384	0.00263	0.00308	0.00221	0.00128	0.00253	0.00415
30	0.00374	0.00385	0.00467	0.00432	0.00288	0.00384	0.00284	0.00139	0.00282	0.00482
31	0.00022	0.00035	0.00038	0.00034	0.00187	0.00028	0.00015	0.00112	0.00310	0.00020
32	0.00079	0.00105	0.00119	0.00093	0.00214	0.00092	0.00056	0.00123	0.00338	0.00069
33	0.00146	0.00175	0.00204	0.00147	0.00241	0.00161	0.00106	0.00134	0.00366	0.00123
34	0.00218	0.00245	0.00290	0.00199	0.00266	0.00233	0.00162	0.00145	0.00393	0.00182
35	0.00294	0.00315	0.00378	0.00250	0.00291	0.00308	0.00221	0.00156	0.00420	0.00243
36	0.00374	0.00385	0.00467	0.00300	0.00315	0.00384	0.00284	0.00166	0.00446	0.00305

Table B.2. Reliability of components in each period in Model 1

(M,C)	1	2	3	4	5	6	7	8	9	10	Reliability
1	99.98%	99.97%	99.96%	99.97%	99.97%	99.97%	99.99%	99.99%	99.98%	99.98%	99.74%
2	99.92%	99.90%	99.88%	99.91%	99.92%	99.91%	99.94%	99.97%	99.93%	99.93%	99.22%
3	99.85%	99.83%	99.80%	99.85%	99.89%	99.84%	99.89%	99.96%	99.90%	99.88%	98.69%
4	99.78%	99.76%	99.71%	99.80%	99.86%	99.77%	99.84%	99.94%	99.87%	99.82%	98.15%
5	99.71%	99.69%	99.62%	99.75%	99.83%	99.69%	99.78%	99.93%	99.83%	99.76%	97.61%
6	99.98%	99.62%	99.73%	99.83%	99.80%	99.75%	99.72%	99.92%	99.88%	99.69%	97.92%
7	99.92%	99.97%	99.96%	99.87%	99.86%	99.78%	99.99%	99.93%	99.85%	99.98%	99.10%
8	99.85%	99.90%	99.88%	99.81%	99.83%	99.71%	99.94%	99.91%	99.82%	99.93%	98.60%
9	99.78%	99.83%	99.80%	99.76%	99.80%	99.64%	99.89%	99.90%	99.79%	99.88%	98.08%
10	99.71%	99.76%	99.71%	99.71%	99.78%	99.56%	99.84%	99.89%	99.76%	99.82%	97.55%
11	99.63%	99.69%	99.62%	99.66%	99.75%	99.48%	99.78%	99.88%	99.73%	99.76%	97.01%
12	99.98%	99.97%	99.73%	99.97%	99.83%	99.97%	99.99%	99.87%	99.82%	99.98%	99.10%
13	99.92%	99.90%	99.64%	99.91%	99.81%	99.91%	99.94%	99.86%	99.79%	99.93%	98.61%
14	99.85%	99.83%	99.56%	99.85%	99.78%	99.84%	99.89%	99.85%	99.76%	99.88%	98.10%
15	99.78%	99.76%	99.47%	99.80%	99.75%	99.77%	99.84%	99.84%	99.73%	99.82%	97.58%
16	99.71%	99.69%	99.38%	99.75%	99.73%	99.69%	99.78%	99.83%	99.70%	99.76%	97.04%
17	99.63%	99.62%	99.29%	99.70%	99.70%	99.62%	99.72%	99.82%	99.67%	99.69%	96.51%
18	99.98%	99.97%	99.96%	99.97%	99.97%	99.97%	99.99%	99.99%	99.98%	99.63%	99.39%
19	99.92%	99.90%	99.88%	99.91%	99.92%	99.91%	99.94%	99.97%	99.93%	99.56%	98.85%
20	99.85%	99.83%	99.80%	99.85%	99.89%	99.84%	99.89%	99.96%	99.90%	99.50%	98.31%
21	99.86%	99.84%	99.71%	99.88%	99.86%	99.84%	99.84%	99.94%	99.91%	99.98%	98.68%
22	99.79%	99.77%	99.62%	99.83%	99.83%	99.77%	99.78%	99.93%	99.88%	99.93%	98.15%
23	99.72%	99.70%	99.53%	99.78%	99.80%	99.70%	99.72%	99.92%	99.85%	99.88%	97.61%
24	99.64%	99.63%	99.45%	99.73%	99.77%	99.62%	99.65%	99.90%	99.82%	99.82%	97.06%
25	99.98%	99.97%	99.96%	99.81%	99.85%	99.97%	99.99%	99.92%	99.87%	99.84%	99.15%
26	99.92%	99.90%	99.88%	99.76%	99.82%	99.91%	99.94%	99.91%	99.84%	99.78%	98.66%
27	99.85%	99.83%	99.80%	99.71%	99.79%	99.84%	99.89%	99.89%	99.81%	99.72%	98.14%
28	99.78%	99.76%	99.71%	99.66%	99.76%	99.77%	99.84%	99.88%	99.78%	99.65%	97.62%
29	99.71%	99.69%	99.62%	99.62%	99.74%	99.69%	99.78%	99.87%	99.75%	99.59%	97.08%
30	99.63%	99.62%	99.53%	99.57%	99.71%	99.62%	99.72%	99.86%	99.72%	99.52%	96.54%
31	99.98%	99.97%	99.96%	99.97%	99.81%	99.97%	99.99%	99.89%	99.69%	99.98%	99.20%
32	99.92%	99.90%	99.88%	99.91%	99.79%	99.91%	99.94%	99.88%	99.66%	99.93%	98.72%
33	99.85%	99.83%	99.80%	99.85%	99.76%	99.84%	99.89%	99.87%	99.63%	99.88%	98.21%
34	99.78%	99.76%	99.71%	99.80%	99.73%	99.77%	99.84%	99.85%	99.61%	99.82%	97.69%
35	99.71%	99.69%	99.62%	99.75%	99.71%	99.69%	99.78%	99.84%	99.58%	99.76%	97.17%
36	99.63%	99.62%	99.53%	99.70%	99.69%	99.62%	99.72%	99.83%	99.55%	99.69%	96.63%

Overall Reliability = 50.00%

Table B.3. Cost of components in each period in Model 1

(M,C)	1	2	3	4	5	6	7	8	9	10	Fixed Cost
1	0.06	0.08	0.10	0.07	0.07	0.08	0.03	0.03	0.05	0.05	0.00
2	0.20	0.25	0.32	0.20	0.17	0.26	0.11	0.07	0.14	0.18	0.00
3	0.36	0.42	0.55	0.31	0.24	0.45	0.21	0.10	0.22	0.31	0.00
4	0.54	0.59	0.78	0.42	0.32	0.65	0.32	0.13	0.29	0.46	0.00
5	200.74	0.76	66.02	42.53	0.38	38.86	0.44	0.16	48.36	0.62	800.00
6	0.06	210.92	245.73	42.36	50.44	38.70	175.57	30.19	0.26	250.78	800.00
7	0.20	0.08	0.10	0.28	0.31	0.60	0.03	0.16	0.33	0.05	0.00
8	0.36	0.25	0.32	0.39	0.37	0.81	0.11	0.19	0.40	0.18	0.00
9	0.54	0.42	0.55	0.50	0.44	1.02	0.21	0.22	0.46	0.31	0.00
10	0.74	0.59	0.78	0.60	0.49	1.24	0.32	0.25	0.53	0.46	0.00
11	200.94	210.76	66.02	180.71	50.55	236.46	175.44	0.27	48.59	250.62	800.00
12	0.06	0.08	0.73	0.07	0.37	0.08	0.03	0.30	0.39	0.05	0.00
13	0.20	0.25	0.96	0.20	0.43	0.26	0.11	0.32	0.45	0.18	0.00
14	0.36	0.42	1.20	0.31	0.49	0.45	0.21	0.34	0.52	0.31	0.00
15	0.54	0.59	1.44	0.42	0.55	0.65	0.32	0.37	0.58	0.46	0.00
16	0.74	0.76	1.68	0.53	0.60	0.86	0.44	0.39	0.64	0.62	0.00
17	200.94	210.92	246.93	180.63	205.66	236.07	175.57	215.41	210.70	0.78	800.00
18	0.06	0.08	0.10	0.07	0.07	0.08	0.03	0.03	0.05	0.94	0.00
19	0.20	0.25	0.32	0.20	0.17	0.26	0.11	0.07	0.14	1.11	0.00
20	35.36	32.42	0.55	42.31	0.24	38.45	0.21	0.10	48.22	251.28	800.00
21	0.34	0.38	0.78	0.25	0.32	0.44	0.32	0.13	0.18	0.05	0.00
22	0.52	0.54	1.02	0.36	0.38	0.64	0.44	0.16	0.26	0.18	0.00
23	0.71	0.71	1.26	0.47	0.44	0.85	0.57	0.19	0.33	0.31	0.00
24	200.91	210.88	246.50	42.58	50.50	236.06	175.70	30.22	48.39	55.46	800.00
25	0.06	0.08	0.10	0.39	0.34	0.08	0.03	0.18	0.28	0.42	0.00
26	0.20	0.25	0.32	0.50	0.40	0.26	0.11	0.21	0.35	0.57	0.00
27	0.36	0.42	0.55	0.60	0.46	0.45	0.21	0.24	0.42	0.73	0.00
28	0.54	0.59	0.78	0.71	0.52	0.65	0.32	0.26	0.48	0.89	0.00
29	0.74	0.76	1.02	0.81	0.58	0.86	0.44	0.29	0.54	1.06	0.00
30	200.94	210.92	246.26	180.91	50.63	236.07	175.57	30.31	0.61	251.23	800.00
31	0.06	0.08	0.10	0.07	0.41	0.08	0.03	0.25	0.67	0.05	0.00
32	0.20	0.25	0.32	0.20	0.47	0.26	0.11	0.28	0.73	0.18	0.00
33	0.36	0.42	0.55	0.31	0.53	0.45	0.21	0.30	0.79	0.31	0.00
34	0.54	0.59	0.78	0.42	0.59	0.65	0.32	0.33	0.84	0.46	0.00
35	0.74	0.76	1.02	0.53	0.64	0.86	0.44	0.35	0.90	0.62	0.00
36	0.94	0.92	1.26	0.63	0.69	1.07	0.57	0.37	0.96	0.78	0.00

Total Cost = $13,797.33

Table B.4. Expected number of failures of components in each period in Model 2

(M,C)	1	2	3	4	5	6	7	8	9	10
1	0.00022	0.00035	0.00038	0.00034	0.00032	0.00028	0.00015	0.00012	0.00025	0.00020
2	0.00079	0.00105	0.00119	0.00093	0.00076	0.00092	0.00056	0.00030	0.00065	0.00069
3	0.00022	0.00035	0.00038	0.00034	0.00032	0.00028	0.00015	0.00035	0.00025	0.00020
4	0.00079	0.00105	0.00119	0.00093	0.00076	0.00092	0.00056	0.00050	0.00065	0.00069
5	0.00146	0.00175	0.00204	0.00147	0.00111	0.00161	0.00106	0.00064	0.00101	0.00123
6	0.00218	0.00245	0.00290	0.00199	0.00143	0.00233	0.00162	0.00076	0.00134	0.00182
7	0.00294	0.00315	0.00378	0.00250	0.00173	0.00308	0.00221	0.00089	0.00166	0.00243
8	0.00374	0.00385	0.00467	0.00300	0.00201	0.00384	0.00284	0.00101	0.00197	0.00305
9	0.00458	0.00455	0.00556	0.00348	0.00228	0.00461	0.00350	0.00112	0.00227	0.00370
10	0.00543	0.00525	0.00646	0.00396	0.00254	0.00539	0.00419	0.00124	0.00256	0.00436
11	0.00631	0.00595	0.00737	0.00443	0.00279	0.00619	0.00490	0.00135	0.00285	0.00504
12	0.00022	0.00035	0.00038	0.00034	0.00032	0.00028	0.00015	0.00109	0.00025	0.00020
13	0.00079	0.00105	0.00119	0.00093	0.00076	0.00092	0.00056	0.00120	0.00065	0.00069
14	0.00146	0.00175	0.00204	0.00147	0.00111	0.00161	0.00106	0.00131	0.00101	0.00123
15	0.00218	0.00245	0.00290	0.00199	0.00143	0.00233	0.00162	0.00142	0.00134	0.00182
16	0.00294	0.00315	0.00378	0.00250	0.00173	0.00308	0.00221	0.00153	0.00166	0.00243
17	0.00022	0.00035	0.00038	0.00034	0.00124	0.00028	0.00015	0.00122	0.00025	0.00020
18	0.00079	0.00105	0.00119	0.00093	0.00155	0.00092	0.00056	0.00133	0.00065	0.00069
19	0.00146	0.00175	0.00204	0.00147	0.00184	0.00161	0.00106	0.00144	0.00101	0.00123
20	0.00218	0.00245	0.00290	0.00199	0.00212	0.00233	0.00162	0.00154	0.00134	0.00182
21	0.00180	0.00197	0.00221	0.00147	0.00032	0.00204	0.00221	0.00012	0.00166	0.00243
22	0.00022	0.00035	0.00038	0.00034	0.00055	0.00028	0.00015	0.00030	0.00025	0.00020
23	0.00079	0.00105	0.00119	0.00093	0.00093	0.00092	0.00056	0.00045	0.00065	0.00069
24	0.00146	0.00175	0.00204	0.00147	0.00127	0.00161	0.00106	0.00059	0.00101	0.00123
25	0.00218	0.00245	0.00290	0.00199	0.00158	0.00233	0.00162	0.00072	0.00134	0.00182
26	0.00294	0.00315	0.00378	0.00250	0.00187	0.00308	0.00221	0.00084	0.00166	0.00243
27	0.00374	0.00385	0.00467	0.00300	0.00214	0.00384	0.00284	0.00097	0.00197	0.00305
28	0.00022	0.00035	0.00038	0.00034	0.00147	0.00028	0.00015	0.00012	0.00025	0.00020
29	0.00079	0.00105	0.00119	0.00093	0.00176	0.00092	0.00056	0.00030	0.00065	0.00069
30	0.00146	0.00175	0.00204	0.00147	0.00204	0.00161	0.00106	0.00045	0.00101	0.00123
31	0.00218	0.00245	0.00290	0.00199	0.00231	0.00233	0.00162	0.00059	0.00134	0.00182
32	0.00294	0.00315	0.00378	0.00250	0.00257	0.00308	0.00221	0.00072	0.00166	0.00243
33	0.00374	0.00385	0.00467	0.00300	0.00281	0.00384	0.00284	0.00084	0.00197	0.00305
34	0.00458	0.00455	0.00556	0.00348	0.00306	0.00461	0.00350	0.00097	0.00227	0.00370
35	0.00543	0.00525	0.00646	0.00396	0.00329	0.00539	0.00419	0.00108	0.00256	0.00436
36	0.00631	0.00595	0.00737	0.00443	0.00352	0.00619	0.00490	0.00120	0.00285	0.00504

Table B.5. Reliability of components in each period in Model 2

(M,C)	1	2	3	4	5	6	7	8	9	10	Reliability
1	99.98%	99.97%	99.96%	99.97%	99.97%	99.97%	99.99%	99.99%	99.98%	99.98%	99.74%
2	99.92%	99.90%	99.88%	99.91%	99.92%	99.91%	99.94%	99.97%	99.93%	99.93%	99.22%
3	99.98%	99.97%	99.96%	99.97%	99.97%	99.97%	99.99%	99.96%	99.98%	99.98%	99.72%
4	99.92%	99.90%	99.88%	99.91%	99.92%	99.91%	99.94%	99.95%	99.93%	99.93%	99.20%
5	99.85%	99.83%	99.80%	99.85%	99.89%	99.84%	99.89%	99.94%	99.90%	99.88%	98.67%
6	99.78%	99.76%	99.71%	99.80%	99.86%	99.77%	99.84%	99.92%	99.87%	99.82%	98.13%
7	99.71%	99.69%	99.62%	99.75%	99.83%	99.69%	99.78%	99.91%	99.83%	99.76%	97.59%
8	99.63%	99.62%	99.53%	99.70%	99.80%	99.62%	99.72%	99.90%	99.80%	99.69%	97.05%
9	99.54%	99.55%	99.45%	99.65%	99.77%	99.54%	99.65%	99.89%	99.77%	99.63%	96.50%
10	99.46%	99.48%	99.36%	99.60%	99.75%	99.46%	99.58%	99.88%	99.74%	99.56%	95.95%
11	99.37%	99.41%	99.27%	99.56%	99.72%	99.38%	99.51%	99.87%	99.72%	99.50%	95.39%
12	99.98%	99.97%	99.96%	99.97%	99.97%	99.97%	99.99%	99.89%	99.98%	99.98%	99.64%
13	99.92%	99.90%	99.88%	99.91%	99.92%	99.91%	99.94%	99.88%	99.93%	99.93%	99.13%
14	99.85%	99.83%	99.80%	99.85%	99.89%	99.84%	99.89%	99.87%	99.90%	99.88%	98.60%
15	99.78%	99.76%	99.71%	99.80%	99.86%	99.77%	99.84%	99.86%	99.87%	99.82%	98.07%
16	99.71%	99.69%	99.62%	99.75%	99.83%	99.69%	99.78%	99.85%	99.83%	99.76%	97.53%
17	99.98%	99.97%	99.96%	99.97%	99.88%	99.97%	99.99%	99.88%	99.98%	99.98%	99.54%
18	99.92%	99.90%	99.88%	99.91%	99.84%	99.91%	99.94%	99.87%	99.93%	99.93%	99.04%
19	99.85%	99.83%	99.80%	99.85%	99.82%	99.84%	99.89%	99.86%	99.90%	99.88%	98.52%
20	99.78%	99.76%	99.71%	99.80%	99.79%	99.77%	99.84%	99.85%	99.87%	99.82%	97.99%
21	99.82%	99.80%	99.78%	99.85%	99.97%	99.80%	99.78%	99.99%	99.83%	99.76%	98.39%
22	99.98%	99.97%	99.96%	99.97%	99.95%	99.97%	99.99%	99.97%	99.98%	99.98%	99.70%
23	99.92%	99.90%	99.88%	99.91%	99.91%	99.91%	99.94%	99.96%	99.93%	99.93%	99.19%
24	99.85%	99.83%	99.80%	99.85%	99.87%	99.84%	99.89%	99.94%	99.90%	99.88%	98.66%
25	99.78%	99.76%	99.71%	99.80%	99.84%	99.77%	99.84%	99.93%	99.87%	99.82%	98.12%
26	99.71%	99.69%	99.62%	99.75%	99.81%	99.69%	99.78%	99.92%	99.83%	99.76%	97.58%
27	99.63%	99.62%	99.53%	99.70%	99.79%	99.62%	99.72%	99.90%	99.80%	99.69%	97.04%
28	99.98%	99.97%	99.96%	99.97%	99.85%	99.97%	99.99%	99.99%	99.98%	99.98%	99.63%
29	99.92%	99.90%	99.88%	99.91%	99.82%	99.91%	99.94%	99.97%	99.93%	99.93%	99.12%
30	99.85%	99.83%	99.80%	99.85%	99.80%	99.84%	99.89%	99.96%	99.90%	99.88%	98.60%
31	99.78%	99.76%	99.71%	99.80%	99.77%	99.77%	99.84%	99.94%	99.87%	99.82%	98.07%
32	99.71%	99.69%	99.62%	99.75%	99.74%	99.69%	99.78%	99.93%	99.83%	99.76%	97.53%
33	99.63%	99.62%	99.53%	99.70%	99.72%	99.62%	99.72%	99.92%	99.80%	99.69%	96.98%
34	99.54%	99.55%	99.45%	99.65%	99.69%	99.54%	99.65%	99.90%	99.77%	99.63%	96.44%
35	99.46%	99.48%	99.36%	99.60%	99.67%	99.46%	99.58%	99.89%	99.74%	99.56%	95.89%
36	99.37%	99.41%	99.27%	99.56%	99.65%	99.38%	99.51%	99.88%	99.72%	99.50%	95.34%

Overall Reliability = 49.92%

Table B.6. Cost of components in each period in Model 2

(M,C)	1	2	3	4	5	6	7	8	9	10	Fixed Cost
1	0.06	0.08	0.10	0.07	0.07	0.08	0.03	0.03	0.05	0.05	0.00
2	200.20	210.25	245.32	180.20	205.17	235.26	175.11	30.07	210.14	250.18	800.00
3	0.06	0.08	0.10	0.07	0.07	0.08	0.03	0.08	0.05	0.05	0.00
4	0.20	0.25	0.32	0.20	0.17	0.26	0.11	0.11	0.14	0.18	0.00
5	0.36	0.42	0.55	0.31	0.24	0.45	0.21	0.14	0.22	0.31	0.00
6	0.54	0.59	0.78	0.42	0.32	0.65	0.32	0.17	0.29	0.46	0.00
7	0.74	0.76	1.02	0.53	0.38	0.86	0.44	0.20	0.36	0.62	0.00
8	0.94	0.92	1.26	0.63	0.44	1.07	0.57	0.23	0.42	0.78	0.00
9	1.14	1.09	1.50	0.73	0.50	1.29	0.70	0.25	0.49	0.94	0.00
10	1.36	1.26	1.74	0.83	0.56	1.51	0.84	0.28	0.55	1.11	0.00
11	201.58	211.43	246.99	180.93	205.61	236.73	175.98	30.30	210.61	251.28	800.00
12	0.06	0.08	0.10	0.07	0.07	0.08	0.03	0.24	0.05	0.05	0.00
13	0.20	0.25	0.32	0.20	0.17	0.26	0.11	0.27	0.14	0.18	0.00
14	0.36	0.42	0.55	0.31	0.24	0.45	0.21	0.30	0.22	0.31	0.00
15	0.54	0.59	0.78	0.42	0.32	0.65	0.32	0.32	0.29	0.46	0.00
16	200.74	210.76	246.02	180.53	50.38	235.86	175.44	30.34	210.36	250.62	800.00
17	0.06	0.08	0.10	0.07	0.27	0.08	0.03	0.27	0.05	0.05	0.00
18	0.20	0.25	0.32	0.20	0.34	0.26	0.11	0.30	0.14	0.18	0.00
19	0.36	0.42	0.55	0.31	0.41	0.45	0.21	0.32	0.22	0.31	0.00
20	35.54	32.59	65.78	42.42	205.47	38.65	0.32	215.35	0.29	0.46	800.00
21	200.45	210.47	245.60	180.31	50.07	235.57	175.44	0.03	210.36	250.62	800.00
22	0.06	0.08	0.10	0.07	0.12	0.08	0.03	0.07	0.05	0.05	0.00
23	0.20	0.25	0.32	0.20	0.21	0.26	0.11	0.10	0.14	0.18	0.00
24	0.36	0.42	0.55	0.31	0.28	0.45	0.21	0.13	0.22	0.31	0.00
25	0.54	0.59	0.78	0.42	0.35	0.65	0.32	0.16	0.29	0.46	0.00
26	0.74	0.76	1.02	0.53	0.41	0.86	0.44	0.19	0.36	0.62	0.00
27	200.94	210.92	246.26	180.63	50.47	236.07	175.57	215.22	210.42	250.78	800.00
28	0.06	0.08	0.10	0.07	0.32	0.08	0.03	0.03	0.05	0.05	0.00
29	0.20	0.25	0.32	0.20	0.39	0.26	0.11	0.07	0.14	0.18	0.00
30	0.36	0.42	0.55	0.31	0.45	0.45	0.21	0.10	0.22	0.31	0.00
31	0.54	0.59	0.78	0.42	0.51	0.65	0.32	0.13	0.29	0.46	0.00
32	0.74	0.76	1.02	0.53	0.56	0.86	0.44	0.16	0.36	0.62	0.00
33	0.94	0.92	1.26	0.63	0.62	1.07	0.57	0.19	0.42	0.78	0.00
34	1.14	1.09	1.50	0.73	0.67	1.29	0.70	0.22	0.49	0.94	0.00
35	1.36	1.26	1.74	0.83	0.72	1.51	0.84	0.24	0.55	1.11	0.00
36	1.58	1.43	1.99	0.93	0.77	1.73	0.98	0.27	0.61	1.28	0.00

Total Cost = 14,989.74

APPENDIX C

MATLAB PROGRAMS OF GENETIC ALGORITHMS

```
% Data of the Multi-Objective Optimization Model

% Number of components and periods
N = 10;
T = 36;
J = 36;
L = T/J;

% Specification of the components
% Parameters of the Failure function
Lambda = [0.00022  0.00035  0.00038  0.00034  0.00032  0.00028  0.00015
0.00012  0.00025  0.00020];
Beta = [2.20  2.00   2.05   1.90   1.75   2.10   2.25  1.80   1.85
2.15];
% Improvement factor (Age reduction coefficient)
Alpha = [0.62  0.58  0.55  0.50  0.48  0.65  0.75  0.68  0.52  0.67];
% Failure cost
Failure_Cost = [250  240  270  210  220  280  200  225  215  255];
% Maintenance cost
M_Cost = [35  32  65  42  50  38  45  30  48  55];
% Replacement cost
R_Cost = [200  210  245  180  205  235  175  215  210  250];
% Fixed cost
Fixed_Cost = 800;

% Engineering economics parameters
% Inflation rates of failure cost, maintenance cost, replacement cost
and fixed cost
Inf_Failure = 0.01/12;
Inf_M = 0.015/12;
Inf_R = 0.02/12;
Inf_Fix = 0.01/12;
% Interest rate
Int_Rate = 0.03/12;

% Parameters of the multi-objective optimization model
% Weights of the objective functions in weighted method, W1+W2 = 1
%W1 = 0.0; W2 = 1.0;
%W1 = 0.1; W2 = 0.9;
%W1 = 0.2; W2 = 0.8;
%W1 = 0.3; W2 = 0.7;
%W1 = 0.4; W2 = 0.6;
%W1 = 0.5; W2 = 0.5;
%W1 = 0.6; W2 = 0.4;
%W1 = 0.7; W2 = 0.3;
W1 = 0.8; W2 = 0.2;
%W1 = 0.9; W2 = 0.1;
%W1 = 1.0; W2 = 0.0;
% Design goals for the objective functions in goal attainment method
% Given budget
GB = 5000;
% Required reliability
RR = 0.50;
```

```
% Fitness Functions of the Cost and Reliability Functions
function [cost,reliability,fit1,fit2,fit3] = Fitness(a)

Data;
% This section changes a(1,N*T) to A(N,T)
A = zeros(N,T);
for i = 1:1:N
    for j = 1:1:T
        A(i,j) = a(1,(i-1)*T+j);
    end
end

% This section calculates the x(starting effective age) and xp(ending
effective age)
x = zeros(N,T);
for i = 1:1:N
    for j = 1:1:T-1
        if A(i,j) == 0
           x(i,j+1) = x(i,j)+L;
        elseif A(i,j) == 1
           x(i,j+1) = Alpha(i)*(x(i,j)+L);
        elseif A(i,j) == 2
           x(i,j+1) = 0;
        end
    end
end
xp = x+L;

% This section calculates the cost and reliability functions for series
system of components
cost = 0;
max_cost = 0;
xx = zeros(N,T);
xxp = xx+L;
reliability = 1;
for j = 1:1:T
    counter = 0;
    for i = 1:1:N
        if A(i,j) == 0
           cost = cost+((Failure_Cost(i)*Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i)))*(1+Inf_Failure)^j));
        elseif A(i,j) == 1
           cost = cost+((Failure_Cost(i)*Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i)))*(1+Inf_Failure)^j)+(M_Cost(i)*(1+Inf_M)^j));
        elseif A(i,j) == 2
           cost = cost+((Failure_Cost(i)*Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i)))*(1+Inf_Failure)^j)+(R_Cost(i)*(1+Inf_R)^j));
        end
        if A(i,j) == 1 || A(i,j) == 2
           counter = 1;
        end
        max_cost =
max_cost+((Failure_Cost(i)*Lambda(i)*((xxp(i,j)^Beta(i))-
(xx(i,j)^Beta(i)))*(1+Inf_Failure)^j)+(R_Cost(i)*(1+Inf_R)^j));
         reliability = reliability*exp(-Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i))));
    end
```

```
    if counter == 1
       cost = cost+(Fixed_Cost*(1+Inf_Fix)^j);
    end
    cost = cost*(1+Int_Rate)^(-j);
    max_cost = max_cost+(Fixed_Cost*(1+Inf_Fix)^j);
    max_cost = max_cost+(1+Int_Rate)^(-j);
end

% The fitness functions,
fit1 = W1*(cost/max_cost)+W2*(-reliability);
fit2 = -reliability+(1/max_cost)*abs(GB-cost);
fit3 = (cost/max_cost)+abs(RR-reliability);

% One point Crossover Function
function [offspring] = Onepointcrossover(parent1,parent2)

Data;
crossoverpoint = fix(N*T*rand+1);
offspring =
[parent1(:,1:crossoverpoint),parent2(:,crossoverpoint+1:N*T)];

% Two point Crossover Function
function [offspring] = Twopointcrossover(parent1,parent2)

Data;
crossoverpoint1 = fix(N*T*rand+1);
crossoverpoint2 = fix(N*T*rand+1);
crossoverpoint1 = abs((crossoverpoint1+crossoverpoint2)/2)-
abs((crossoverpoint1-crossoverpoint2)/2);
crossoverpoint2 =
abs((crossoverpoint1+crossoverpoint2)/2)+abs((crossoverpoint1-
crossoverpoint2)/2);
offspring =
[parent1(:,1:crossoverpoint1),parent2(:,crossoverpoint1+1:crossoverpoin
t2),parent1(:,crossoverpoint2+1:N*T)];

% N point Crossover Function
function [offspring] = Npointcrossover(parent1,parent2)

Data;
for i = 1:1:fix(N/2)
    for j = 1:1:T
        offspring(:,(2*(i-1))*T+j) = parent1(:,(2*(i-1))*T+j);
        offspring(:,(2*i-1)*T+j) = parent2(:,(2*i-1)*T+j);
        if mod(N,2) == 1
           offspring(:,(N-1)*T+j) = parent1(:,(N-1)*T+j);
        end
    end
end

% NT point Crossover Function
function [offspring] = NTpointcrossover(parent1,parent2)

Data;
for i = 1:1:(N*T)/2
    offspring(:,2*i-1) = parent1(:,2*i-1);
    offspring(:,2*i) = parent2(:,2*i);
```

```
end

% Two point Inverse Crossover Function
function [offspring] = Ordercrossover(parent1,parent2)

Data;
crossoverpoint1 = fix(N*T*rand+1);
crossoverpoint2 = fix(N*T*rand+1);
crossoverpoint1 = abs((crossoverpoint1+crossoverpoint2)/2)-
abs((crossoverpoint1-crossoverpoint2)/2);
crossoverpoint2 =
abs((crossoverpoint1+crossoverpoint2)/2)+abs((crossoverpoint1-
crossoverpoint2)/2);
for i = 1:1:N*T
    parent1_inv(:,N*T-i+1) = parent1(:,i);
end
offspring =
[parent1_inv(:,1:crossoverpoint1),parent2(:,crossoverpoint1+1:crossover
point2),parent1_inv(:,crossoverpoint2+1:N*T)];

% Mutation Function
function [individual] = Mutation(individual)

Data;
mutation_point = fix(N*T*rand+1);
if individual(:,mutation_point) == 0
   if (rand < 0.5)
      for k = 1:1:N
          if mod(mutation_point,T) == 0
             individual(:,(mod(mutation_point,T)+k*T)) = 1;
          else
             individual(:,(mod(mutation_point,T)+(k-1)*T)) = 1;
          end
      end
   elseif (rand >= 0.5)
      for k = 1:1:N
          if mod(mutation_point,T) == 0
             individual(:,(mod(mutation_point,T)+k*T)) = 2;
          else
             individual(:,(mod(mutation_point,T)+(k-1)*T)) = 2;
          end
      end
   end
elseif individual(:,mutation_point) == 1 ||
individual(:,mutation_point) == 2
   for k = 1:1:N
       if mod(mutation_point,T) == 0
          individual(:,(mod(mutation_point,T)+k*T)) = 0;
       else
          individual(:,(mod(mutation_point,T)+(k-1)*T)) = 0;
       end
   end
end
```

```
% Generational Genetic Algorithm

% Generational genetic algorithm parameters
% Number of generations:  500
% Population size:        2000
% Probability of selection: 0.20
% Probability of crossover: 0.40
% Probability of mutation:  0.40
clear;
generation_number = 500;
population_size = 2000;
p_selection = 0.20;
p_crossover = 0.40;
p_mutation = 0.40;
min = 0;
max = 2;
Data;

% Initial population
a = zeros(1,T*N);
initial_population = zeros(population_size,T*N+5);
for i = 1:1:population_size
    for j = 1:1:T*N
        a(j) = fix((max-min+1)*rand+min);
    end
    [cost,reliability,fit1,fit2,fit3] = Fitness(a);
    initial_population(i,1:N*T) = a ;
    initial_population(i,N*T+1:N*T+5) =
[cost,reliability,fit1,fit2,fit3];
end
population = initial_population;

for g = 1:1:generation_number
    % Selection procedure
    population_sorted = sortrows(population,N*T+5);
    population_selected =
population_sorted(1:fix(p_selection*population_size),:);

    % Crossover procedures
    for i = 1:1:p_crossover*population_size
        parent1 = population(fix((population_size)*rand+1),:);
        parent2 = population(fix((population_size)*rand+1),:);
        if parent1(:,N*T+5) ~= parent2(:,N*T+5)
           % One point crossover
           %offspring = Onepointcrossover(parent1,parent2);

           % Two point crossover
           %offspring = Twopointcrossover(parent1,parent2);

           % N point crossover
           %offspring = Npointcrossover(parent1,parent2);

           % NT point crossover
           offspring = NTpointcrossover(parent1,parent2);
        elseif parent1(:,N*T+5) == parent2(:,N*T+5)
           % Two point inverse crossover
           offspring = Ordercrossover(parent1,parent2);
```

```
        end
        [cost,reliability,fit1,fit2,fit3] = Fitness(offspring);
        population_crossover(i,1:N*T) = offspring;
        population_crossover(i,N*T+1:N*T+5) =
[cost,reliability,fit1,fit2,fit3];
    end

    % Mutation procedure
    for i = 1:1:p_mutation*population_size
        individual = population(fix((population_size)*rand+1),:);
        individual_mutated = Mutation(individual);
        [cost,reliability,fit1,fit2,fit3] =
Fitness(individual_mutated);
        population_mutation(i,1:N*T) = individual_mutated(:,1:N*T);
        population_mutation(i,N*T+1:N*T+5) =
[cost,reliability,fit1,fit2,fit3];
    end

    % This section generates a new population based on selection,
crossover and mutation procedures
    population =
[population_selected;population_crossover;population_mutation];
    % This section sorts the solutions in the current population based
on their fitness value and selects the best one in each generation
    ss = sortrows(population,N*T+5);
    solution_improvement(g,:) = ss(1:1,:);
end

% This section sorts the last population based on its fitness values
and then changes the final solution(1,N*T) to PMR_Schedule(N,T)
last_population = sortrows(population,N*T+5);
final_solution = last_population(1:1,:);
PMR_Schedule = zeros(N,T);
for i = 1:1:N
    for j = 1:1:T
        PMR_Schedule(i,j) = final_solution(1,(i-1)*T+j);
    end
end
```

```
% Steady State Genetic Algorithm

% Steady state genetic algorithm parameters
% Number of generation:    1
% Genetic cycle:         500
% Number of iterations:  100
% Population size:      2000
% Probability of mutation: 0.20
clear;
genetic_cycle = 500;
iteration_number = 100;
population_size = 2000;
p_mutation = 0.20;
min = 0;
max = 2;
Data;

% Initial population
a = zeros(1,T*N);
initial_population = zeros(population_size,T*N+5);
for i = 1:1:population_size
    for j = 1:1:T*N
        a(j) = fix((max-min+1)*rand+min);
    end
    [cost,reliability,fit1,fit2,fit3] = Fitness(a);
    initial_population(i,1:N*T) = a ;
    initial_population(i,N*T+1:N*T+5) =
[cost,reliability,fit1,fit2,fit3];
end
population = initial_population;

for i = 1:1:genetic_cycle
    for j = 1:1:iteration_number
        % Crossover Procedures
        parent1 = population(fix((population_size)*rand+1),:);
        parent2 = population(fix((population_size)*rand+1),:);
        if parent1(:,N*T+5) ~= parent2(:,N*T+5)
           % One point crossover
           %offspring = Onepointcrossover(parent1,parent2);

           % Two point crossover
           %offspring = Twopointcrossover(parent1,parent2);

           % N point crossover
           %offspring = Npointcrossover(parent1,parent2);

           % NT point crossover
           offspring = NTpointcrossover(parent1,parent2);
        elseif parent1(:,N*T+5) == parent2(:,N*T+5)
           % Two point inverse crossover
           offspring = Ordercrossover(parent1,parent2);
        end

        % Mutation procedure
        offspring_mutated = Mutation(offspring);
        [cost,reliability,fit1,fit2,fit3] = Fitness(offspring_mutated);
```

```
        offspring_mutated(:,N*T+1:N*T+5) =
[cost,reliability,fit1,fit2,fit3];
        % This section replaces the new offsprings with the worst
solutions in the population if they are better than the worst solutions
        population_sorted = sortrows(population,N*T+5);
        if offspring_mutated(:,N*T+5) <
population_sorted(population_size,N*T+5)
            population_sorted(population_size,:) = offspring_mutated;
            population_sorted = sortrows(population_sorted);
        end

        population = population_sorted;
    end
    ss = sortrows(population,N*T+5);
    solution_improvement(i,:) = ss(1:1,:);
end

% This section sorts the last population based on its fitness values
and then changes the final solution(1,N*T) to PMR_Schedule(N,T)
last_population = sortrows(population,N*T+5);
final_solution = last_population(1:1,:);
PMR_Schedule = zeros(N,T);
for i = 1:1:N
    for j = 1:1:T
        PMR_Schedule(i,j) = final_solution(1,(i-1)*T+j);
    end
end
```

APPENDIX D

MATLAB PROGRAMS OF SIMULATED ANNEALING

```
% Data of the Multi-Objective Optimization Model

% Number of components and periods
N = 10;
T = 36;
J = 36;
L = T/J;

% Specification of the components
% Parameters of the Failure function
Lambda = [0.00022  0.00035  0.00038  0.00034  0.00032  0.00028  0.00015
0.00012  0.00025  0.00020];
Beta = [2.20  2.00   2.05   1.90   1.75   2.10   2.25  1.80   1.85
2.15];
% Improvement factor (Age reduction coefficient)
Alpha = [0.62  0.58  0.55  0.50  0.48  0.65  0.75  0.68  0.52  0.67];
% Failure cost
Failure_Cost = [250  240  270  210  220  280  200  225  215  255];
% Maintenance cost
M_Cost = [35  32  65  42  50  38  45  30  48  55];
% Replacement cost
R_Cost = [200  210  245  180  205  235  175  215  210  250];
% Fixed cost
Fixed_Cost = 800;

% Engineering economics parameters
% Inflation rates of failure cost, maintenance cost, replacement cost
and fixed cost
Inf_Failure = 0.01/12;
Inf_M = 0.015/12;
Inf_R = 0.02/12;
Inf_Fix = 0.01/12;
% Interest rate
Int_Rate = 0.03/12;

% Parameters of the multi-objective optimization model
% Weights of the objective functions in weighted method, W1+W2 = 1
%W1 = 0.0; W2 = 1.0;
%W1 = 0.1; W2 = 0.9;
%W1 = 0.2; W2 = 0.8;
%W1 = 0.3; W2 = 0.7;
%W1 = 0.4; W2 = 0.6;
%W1 = 0.5; W2 = 0.5;
%W1 = 0.6; W2 = 0.4;
%W1 = 0.7; W2 = 0.3;
W1 = 0.8; W2 = 0.2;
%W1 = 0.9; W2 = 0.1;
%W1 = 1.0; W2 = 0.0;
% Design goals for the objective functions in goal attainment method
% Given budget
GB = 5000;
% Required reliability
RR = 0.50;
```

```
% Fitness Functions of the Cost and Reliability Functions
function [cost,reliability,fit1,fit2,fit3] = Fitness(a)

Data;
% This section changes a(1,N*T) to A(N,T)
A = zeros(N,T);
for i = 1:1:N
    for j = 1:1:T
        A(i,j) = a(1,(i-1)*T+j);
    end
end

% This section calculates the x(starting effective age) and xp(ending
effective age)
x = zeros(N,T);
for i = 1:1:N
    for j = 1:1:T-1
        if A(i,j) == 0
           x(i,j+1) = x(i,j)+L;
        elseif A(i,j) == 1
           x(i,j+1) = Alpha(i)*(x(i,j)+L);
        elseif A(i,j) == 2
           x(i,j+1) = 0;
        end
    end
end
xp = x+L;

% This section calculates the cost and reliability functions for series
system of components
cost = 0;
max_cost = 0;
xx = zeros(N,T);
xxp = xx+L;
reliability = 1;
for j = 1:1:T
    counter = 0;
    for i = 1:1:N
        if A(i,j) == 0
           cost = cost+((Failure_Cost(i)*Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i)))*(1+Inf_Failure)^j));
        elseif A(i,j) == 1
           cost = cost+((Failure_Cost(i)*Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i)))*(1+Inf_Failure)^j)+(M_Cost(i)*(1+Inf_M)^j));
        elseif A(i,j) == 2
           cost = cost+((Failure_Cost(i)*Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i)))*(1+Inf_Failure)^j)+(R_Cost(i)*(1+Inf_R)^j));
        end
        if A(i,j) == 1 || A(i,j) == 2
           counter = 1;
        end
        max_cost =
max_cost+((Failure_Cost(i)*Lambda(i)*((xxp(i,j)^Beta(i))-
(xx(i,j)^Beta(i)))*(1+Inf_Failure)^j)+(R_Cost(i)*(1+Inf_R)^j));
          reliability = reliability*exp(-Lambda(i)*((xp(i,j)^Beta(i))-
(x(i,j)^Beta(i))));
    end
```

```
    if counter == 1
       cost = cost+(Fixed_Cost*(1+Inf_Fix)^j);
    end
    cost = cost*(1+Int_Rate)^(-j);
    max_cost = max_cost+(Fixed_Cost*(1+Inf_Fix)^j);
    max_cost = max_cost+(1+Int_Rate)^(-j);
end

% The fitness functions,
fit1 = W1*(cost/max_cost)+W2*(-reliability);
fit2 = -reliability+(1/max_cost)*abs(GB-cost);
fit3 = (cost/max_cost)+abs(RR-reliability);

% Transition Function
function [x] = Transition(x)

Data;
transition_point = fix(N*T*rand+1);
if x(:,transition_point) == 0
   if (rand < 0.5)
      for k = 1:1:N
          if mod(transition_point,T) == 0
             x(:,(mod(transition_point,T)+k*T)) = 1;
          else
             x(:,(mod(transition_point,T)+(k-1)*T)) = 1;
          end
      end
   elseif (rand >= 0.5)
      for k = 1:1:N
          if mod(transition_point,T) == 0
             x(:,(mod(transition_point,T)+k*T)) = 2;
          else
             x(:,(mod(transition_point,T)+(k-1)*T)) = 2;
          end
      end
   end
elseif x(:,transition_point) == 1 || x(:,transition_point) == 2
   for k = 1:1:N
       if mod(transition_point,T) == 0
          x(:,(mod(transition_point,T)+k*T)) = 0;
       else
          x(:,(mod(transition_point,T)+(k-1)*T)) = 0;
       end
   end
end
```

```
% Simulated Annealing Algorithm

% Simulated annealing algorithm parameters
% Initial temperature: 1000000
% Final temperature:        0.01
% Decreasing rate:          0.98
clear;
t_initial = 1000000;
t_final = 0.01;
t_rate = 0.98;
min = 0;
max = 2;
Data;

% Initial solution
a = zeros(1,T*N);
for j = 1:1:T*N
    a(j) = fix((max-min+1)*rand+min);
end
[cost,reliability,fit1,fit2,fit3] = Fitness(a);
initial_solution(1,1:N*T) = a ;
initial_solution(1,N*T+1:N*T+5) = [cost,reliability,fit1,fit2,fit3];
x = initial_solution;

t_current = t_initial;
i = 1;
while t_final <= t_current

    % Transition procedure
    y = Transition(x);
    [cost,reliability,fit1,fit2,fit3] = Fitness(y);
    y(1,N*T+1:N*T+5) = [cost,reliability,fit1,fit2,fit3];

    % Acceptation procedure
    if y(1,N*T+5) < x(1,N*T+5)
       x = y;
    elseif y(1,N*T+5) >= x(1,N*T+5)
       if rand <= exp(-(y(1,N*T+5)-x(1,N*T+5))/t_current)
          x = y;
       end
    end
    solution_improvement(i,1:N*T+5) = x;
    t_current = t_rate*t_current;
    i = i+1;
end

% This section changes the final solution(1,N*T) to PMR_Schedule(N,T)
ss = sortrows(solution_improvement,N*T+5);
final_solution = ss(1:1,:);
PMR_Schedule = zeros(N,T);
for i = 1:1:N
    for j = 1:1:T
        PMR_Schedule(i,j) = final_solution(1,(i-1)*T+j);
    end
end
```

APPENDIX E

LINGO PROGRAMS OF IMPROVEMENT FACTOR MODELS

```
! Model 1.1-Nonlinear mixed integer optimization model that
minimizes total cost subject to a reliability constraint with
constant improvement factor
based on maintenance and replacement costs;

Model:
Data:
    C = 1;
    T = 36;
    L = 1;
Enddata

Sets:
    Period/1..T/;
    LinkComPer(Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00025;
    Beta = 2.20;
    Failure_Cost = 2500;
    M_Cost = 300;
    R_Cost = 1500;
    Given_Reliability = 0.92;
Enddata

! Objective Function, Minimizing the total cost;
    Min = @Sum(LinkComPer(j): (Failure_Cost * Lambda * ((XP(j)^Beta) -
          (X(j)^Beta))) + M_Cost * M(j) + R_Cost * R(j));

! Constraints;
    ! Recursive functions;
    X(1) = 0;
    @For(LinkComPer(j): XP(j) = X(j) + L);
    @For(LinkComPer(j)| j #GE# 2: X(j) = ((1-M(j-1)) * (1-R(j-1)) *
         (XP(j-1)) + M(j-1) * ((R_Cost-M_Cost)/R_Cost) * (XP(j-1))));

    ! Basic constraints;
    @For(LinkComPer(j): M(j) + R(j) <= 1);
    @For(LinkComPer(j): @BIN(M));
    @For(LinkComPer(j): @BIN(R));

    ! Reliability constraint;
    @Exp(@Sum(LinkComPer(j): (-Lambda * ((XP(j)^Beta) - (X(j)^Beta)))))
         >= Given_Reliability;
End
```

```
! Model 1.2-Nonlinear mixed integer optimization model that
minimizes total cost subject to a reliability constraint with
     variable improvement factor based on effective age;

Model:
Data:
    C = 1;
    T = 36;
    L = 1;
Enddata

Sets:
    Period/1..T/;
    LinkComPer(Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00025;
    Beta = 2.20;
    Failure_Cost = 2500;
    M_Cost = 300;
    R_Cost = 1500;
    Given_Reliability = 0.92;
Enddata

! Objective Function, Minimizing the total cost;
    Min = @Sum(LinkComPer(j): (Failure_Cost * Lambda * ((XP(j)^Beta) -
          (X(j)^Beta))) + M_Cost * M(j) + R_Cost * R(j));

! Constraints;
    ! Recursive functions;
    X(1) = 0;
    @For(LinkComPer(j): XP(j) = X(j) + L);
    @For(LinkComPer(j)| j #GE# 2: X(j) = ((1-M(j-1)) * (1-R(j-1)) *
         (XP(j-1)) + M(j-1) * ((XP(j-1)/(XP(j-1)+1))) * (XP(j-1))));

    ! Basic constraints;
    @For(LinkComPer(j): M(j) + R(j) <= 1);
    @For(LinkComPer(j): @BIN(M));
    @For(LinkComPer(j): @BIN(R));

    ! Reliability constraint;
    @Exp(@Sum(LinkComPer(j): (-Lambda * ((XP(j)^Beta) - (X(j)^Beta)))))
        >= Given_Reliability;
End
```

```
! Model 1.3-Nonlinear mixed integer optimization model that
minimizes total cost subject to a reliability constraint with
variable improvement factor
based on maintenance and replacement costs and effective age;

Model:
Data:
    C = 1;
    T = 36;
    L = 1;
Enddata

Sets:
    Period/1..T/;
    LinkComPer(Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00025;
    Beta = 2.20;
    Failure_Cost = 2500;
    M_Cost = 300;
    R_Cost = 1500;
    Given_Reliability = 0.92;
Enddata

! Objective Function, Minimizing the total cost;
    Min = @Sum(LinkComPer(j): (Failure_Cost * Lambda * ((XP(j)^Beta) -
          (X(j)^Beta))) + M_Cost * M(j) + R_Cost * R(j));

! Constraints;
    ! Recursive functions;
    X(1) = 0;
    @For(LinkComPer(j): XP(j) = X(j) + L);
    @For(LinkComPer(j)| j #GE# 2: X(j) = ((1-M(j-1)) * (1-R(j-1)) *
        (XP(j-1)) + M(j-1) * (((R_Cost-M_Cost)/R_Cost) * (XP(j-
        1)/(XP(j-1)+1))) * (XP(j-1))));

    ! Basic constraints;
    @For(LinkComPer(j): M(j) + R(j) <= 1);
    @For(LinkComPer(j): @BIN(M));
    @For(LinkComPer(j): @BIN(R));

    ! Reliability constraint;
    @Exp(@Sum(LinkComPer(j): (-Lambda * ((XP(j)^Beta) - (X(j)^Beta)))))
        >= Given_Reliability;
End
```

```
! Model 2.1-Nonlinear mixed integer optimization model that
maximizes overall reliability subject to a budgetary constraint
with constant improvement factor
based on maintenance and replacement costs ;

Model:
Data:
    C = 1;
    T = 36;
    L = 1;
Enddata

Sets:
    Period/1..T/:;
    LinkComPer(Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00025;
    Beta = 2.20;
    Failure_Cost = 2500;
    M_Cost = 300;
    R_Cost = 1500;
    Given_Budget = 6000;
Enddata

! Objective Function, Maximizing reliability;
    Max = @Exp(@sum(LinkComPer(j): (-Lambda * ((XP(j)^Beta) -
          (X(j)^Beta)))) ;

! Constraints;
    ! Recursive functions;
    X(1) = 0;
    @For(LinkComPer(j): XP(j) = X(j) + (L));
    @For(LinkComPer(j)| j #GE# 2: X(j) = ((1-M(j-1)) * (1-R(j-1)) *
         (XP(j-1)) + M(j-1) * ((R_Cost-M_Cost)/R_Cost) * (XP(j-1))));

    ! Basic constraints;
    @For(LinkComPer(j): M(j) + R(j) <= 1);
    @For(LinkComPer(j): @BIN(M));
    @For(LinkComPer(j): @BIN(R));

    ! Budget constraint;
    @Sum(LinkComPer(j): (Failure_Cost * Lambda * ((XP(j)^Beta) -
         (X(j)^Beta))) + M_Cost * M(j) + R_Cost * R(j)) <= Given_Budget;
End
```

```
! Model 2.2-Nonlinear mixed integer optimization model that
maximizes overall reliability subject to a budgetary constraint
with variable improvement factor based on effective age;

Model:
Data:
    C = 1;
    T = 36;
    L = 1;
Enddata

Sets:
    Period/1..T/:;
    LinkComPer(Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00025;
    Beta = 2.20;
    Failure_Cost = 2500;
    M_Cost = 300;
    R_Cost = 1500;
    Given_Budget = 6000;
Enddata

! Objective Function, Maximizing the reliability;
    Max = @Exp(@sum(LinkComPer(j): (-Lambda * ((XP(j)^Beta) -
          (X(j)^Beta))))) ;

! Constraints;
    ! Recursive functions;
    X(1) = 0;
    @For(LinkComPer(j): XP(j) = X(j) + (L));
    @For(LinkComPer(j)| j #GE# 2: X(j) = ((1-M(j-1)) * (1-R(j-1)) *
        (XP(j-1)) + M(j-1) * ((XP(j-1)/(XP(j-1)+1))) * (XP(j-1))));

    ! Basic constraints;
    @For(LinkComPer(j): M(j) + R(j) <= 1);
    @For(LinkComPer(j): @BIN(M));
    @For(LinkComPer(j): @BIN(R));

    ! Budget constraint;
    @Sum(LinkComPer(j): (Failure_Cost * Lambda * ((XP(j)^Beta) -
        (X(j)^Beta))) + M_Cost * M(j) + R_Cost * R(j)) <= Given_Budget;
End
```

```
! Model 2.3-Nonlinear mixed integer optimization model that
maximizes overall reliability subject to a budgetary constraint
with variable improvement factor
based on maintenance and replacement costs and effective age;

Model:
Data:
    C = 1;
    T = 36;
    L = 1;
Enddata

Sets:
    Period/1..T/:;
    LinkComPer(Period): X, XP, M, R;
Endsets

Data:
    Lambda = 0.00025;
    Beta = 2.20;
    Failure_Cost = 2500;
    M_Cost = 300;
    R_Cost = 1500;
    Given_Budget = 6000;
Enddata

! Objective Function, Maximizing the reliability;
    Max = @Exp(@sum(LinkComPer(j): (-Lambda * ((XP(j)^Beta) -
          (X(j)^Beta))))) ;

! Constraints;
    ! Recursive functions;
    X(1) = 0;
    @For(LinkComPer(j): XP(j) = X(j) + (L));
    @For(LinkComPer(j)| j #GE# 2: X(j) = ((1-M(j-1)) * (1-R(j-1)) *
         (XP(j-1)) + M(j-1) * (((R_Cost-M_Cost)/R_Cost) * (XP(j-
         1)/(XP(j-1)+1))) * (XP(j-1))));

    ! Basic constraints;
    @For(LinkComPer(j): M(j) + R(j) <= 1);
    @For(LinkComPer(j): @BIN(M));
    @For(LinkComPer(j): @BIN(R));

    ! Budget constraint;
    @Sum(LinkComPer(j): (Failure_Cost * Lambda * ((XP(j)^Beta) -
         (X(j)^Beta))) + M_Cost * M(j) + R_Cost * R(j)) <= Given_Budget;
End
```

APPENDIX F

COMPUTATIONAL RESULTS OF IMPROVEMENT FACTOR MODELS

Table F.1. Variation of improvement factors in each period

	Model 1			Model 2		
(M,F)	Function 1	Function 2	Function 3	Function 1	Function 2	Function 3
1	0.80000	0.50000	0.40000	0.80000	0.50000	0.40000
2	0.80000	0.66667	0.53333	0.80000	0.66667	0.53333
3	0.80000	0.75000	0.60000	0.80000	0.70000	0.60000
4	0.80000	0.80000	0.58947	0.80000	0.76923	0.64000
5	0.80000	0.83333	0.63333	0.80000	0.81250	0.62456
6	0.80000	0.85714	0.61846	0.80000	0.84211	0.65612
7	0.80000	0.50000	0.60520	0.80000	0.86364	0.63974
8	0.80000	0.66667	0.59383	0.80000	0.88000	0.66649
9	0.80000	0.70000	0.58439	0.80000	0.89286	0.64982
10	0.80000	0.76923	0.63016	0.80000	0.50000	0.63374
11	0.80000	0.81250	0.61559	0.80000	0.66667	0.61883
12	0.80000	0.84211	0.60271	0.80000	0.70000	0.65228
13	0.80000	0.86364	0.59173	0.80000	0.72477	0.63607
14	0.80000	0.88000	0.58268	0.80000	0.74415	0.62095
15	0.80000	0.50000	0.62911	0.80000	0.79627	0.60739
16	0.80000	0.66667	0.61464	0.80000	0.83075	0.64477
17	0.80000	0.75000	0.60189	0.80000	0.85525	0.62899
18	0.80000	0.80000	0.59105	0.80000	0.87355	0.65911
19	0.80000	0.83333	0.63432	0.80000	0.88775	0.64262
20	0.80000	0.50000	0.61935	0.80000	0.89908	0.62699
21	0.80000	0.66667	0.60599	0.80000	0.50000	0.65775
22	0.80000	0.70000	0.59449	0.80000	0.66667	0.64132
23	0.80000	0.72477	0.63649	0.80000	0.75000	0.62578
24	0.80000	0.74415	0.66424	0.80000	0.80000	0.65694
25	0.80000	0.79627	0.68394	0.80000	0.83333	0.64053
26	0.80000	0.83075	0.69864	0.80000	0.85714	0.62505
27	0.80000	0.85525	0.71004	0.80000	0.87500	0.65644
28	0.80000	0.87355	0.40000	0.80000	0.88889	0.64006
29	0.80000	0.50000	0.53333	0.80000	0.50000	0.62461
30	0.80000	0.66667	0.60000	0.80000	0.66667	0.61063
31	0.80000	0.70000	0.64000	0.80000	0.75000	0.59843
32	0.80000	0.76923	0.62456	0.80000	0.80000	0.63900
33	0.80000	0.81250	0.65612	0.80000	0.83333	0.66597
34	0.80000	0.84211	0.67805	0.80000	0.85714	0.68520
35	0.80000	0.86364	0.69418	0.80000	0.87500	0.69961
36	0.80000	0.88000	0.70654	0.80000	0.88889	0.71080

CURRICULUM VITAE

Kamran S. Moghaddam

CONTACT INFORMATION

Work Address: Department of Industrial Engineering, J.B. Speed School of Engineering
University of Louisville, Louisville, KY 40292, USA
Work Phone: 502-852-6845
Cell Phone: 502-471-6211
Email: ksmogh01@louisville.edu
kamran.smoghaddam@gmail.com

EDUCATION

Ph.D., Industrial Engineering, May 2010 — 2007 - 2010
University of Louisville, Louisville, KY, USA
Research Focus: *Reliability Engineering, Applied Optimization, Metaheuristic Algorithms*
Dissertation: *Preventive Maintenance and Replacement Scheduling: Models and Algorithms*
Dissertation Advisor: Dr. John S. Usher

M.S., Industrial Engineering, Nov 2003 — 2001 - 2003
Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
Research Focus: *Water Resources Planning, Applied Optimization, Interior-Point Algorithms*
Thesis: *Modeling and Optimization of Karoun-Dez Reservoir System Operation*
Thesis Advisor: Dr. Abbas Seifi

B.S., Applied Mathematics, Feb 2001 — 1996 - 2001
University of Tehran, Tehran, Iran

Diploma, Mathematics and Physics, Jun 1996 — 1989 - 1996
National Organization for Development of Exceptional Talents, Tehran, Iran

CERTIFICATES

Registered Engineer-in-Training in the State of Kentucky — Dec 2009

HONORS AND AWARDS

Outstanding Graduate Student Scholarship Award — Apr 2010
University of Louisville, Louisville, KY, USA

Engineering Exposition Competition Award — Mar 2010
University of Louisville, Louisville, KY, USA

Graduate Fellowship
University of Louisville, Louisville, KY, USA
- *Grosscurth Fellowship* — Jan 2007 - Dec 2008
- *William T. Runner Fellowship* — Jan 2009 - Apr 2010

Nominated for Institute of Industrial Engineers Doctoral Colloquium — May 2009
Industrial Engineering Research Conference, May 2009, Miami, FL, USA

Travel Grant
Graduate Student Council, University of Louisville, Louisville, KY, USA
- *2009 INFORMS Annual Meeting* — Oct 2009
- *2009 Industrial Engineering Research Conference* — May 2009

Department of Industrial Engineering, University of Louisville, Louisville, KY, USA
- *2008 INFORMS Annual Meeting* — Oct 2008
- *2007 INFORMS Annual Meeting* — Nov 2007

American Society for Quality Scholarship Award — Apr 2008
Louisville, KY, USA

Paper Competition Award in Industrial Engineering Conference Nov 2003
Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
15th place in National Entrance Exam of MS in Industrial Engineering Mar 2001
Evaluation of Country's Education Organization, Tehran, Iran

POSITIONS AND EMPLOYMENT

Research and Teaching Assistant Jan 2009 - Apr 2010
Department of Industrial Engineering, University of Louisville, Louisville, KY, USA
Graduate Fellow Jan 2007 - Dec 2008
Department of Industrial Engineering, University of Louisville, Louisville, KY, USA
Project Manager Nov 2005 - Dec 2006
Tehran Comprehensive Transportation and Traffic Studies Company, Tehran, Iran
Industrial Engineer Sep 2002 - Oct 2005
Tehran Comprehensive Transportation and Traffic Studies Company, Tehran, Iran
Industrial Engineer Sep 2001 - Aug 2002
Iran Polymer & Petrochemical Institute, Tehran, Iran

RESEARCH EXPERIENCE

Graduate Fellow and Research Assistant Jan 2007 - Apr 2010
Department of Industrial Engineering, University of Louisville, Louisville, KY, USA
- *Development of optimization models for maintenance and replacement scheduling*
- *Employment of exact and heuristic algorithms to solve optimization models for maintenance and replacement scheduling*
- *Sensitivity analysis and comparison of exact and heuristic algorithms in optimization models for maintenance and replacement scheduling*
- *Development of mathematical models for age reduction and improvement factor for repairable and maintainable systems*
- *Development of chance constrained programming models in a farm management optimization project*
- *Development of intelligent agent-based simulation model in a healthcare management project*

M.S. Student Sep 2001 - Nov 2003
Department of Industrial Engineering, Tehran Polytechnic, Tehran, Iran
- *Development of simulation and optimization of Karoun-Dez reservoir system operation*
- *Development of statistical models to predict seasonal flows in Karoun-Dez reservoir system*

TEACHING EXPERIENCE

Instructor May 2009 - Apr 2010
Department of Industrial Engineering, University of Louisville, Louisville, KY, USA
- *Engineering Economic Analysis*
  Spring 2010: 73 students
  Fall 2009: 48 students, Effectiveness of the instructor: 3.70
  Summer 2009: 74 students, Effectiveness of the instructor: 3.19

Teaching Assistant Jan 2009 - Apr 2009
Department of Industrial Engineering, University of Louisville, Louisville, KY, USA
- *Advanced Production Systems Design*
- *Models for Design and Analysis for Logistics Systems*

Instructor Jan 2004 - Dec 2006
Elmi Karbordi University, Tehran, Iran
- *Calculus I&II*
- *Probability and Statistics I&II*
- *Computer Programming*

Instructor Sep 1997 - Aug 2000
National Organization for Development of Exceptional Talents, Tehran, Iran
- *Introduction to Mathematical Olympiad*
- *Computer Programming*

INDUSTRY EXPERIENCE

Project Manager Nov 2005 - Dec 2006

Tehran Comprehensive Transportation and Traffic Studies Company, Tehran, Iran

- *Engineering economic analysis of 3 transportation and logistics projects*
- *Project management of 4 transportation and logistics projects*
- *Economic feasibility analysis to employ Intelligent Transportation Systems (ITS) in Tehran*

Industrial Engineer Sep 2002 - Oct 2005

Tehran Comprehensive Transportation and Traffic Studies Company, Tehran, Iran

- *Development of computer simulation models for public transportation system*
- *Sensitivity analysis of origin-destination trips (O-D matrices)*
- *Development of mathematical and computer models for trip distributions*
- *Development of statistical models for trip generation and attraction*
- *Development of statistical models for population estimation of the traffic zones in city of Tehran*

Industrial Engineer Sep 2001 - Aug 2002

Iran Polymer & Petrochemical Institute, Tehran, Iran

- *Engineering economic analysis for design of pilot plant emulsion PVC*

REFEREED JOURNAL PUBLICATION

Xu Yang, **Kamran S. Moghaddam**, (2010) Applications of Multi-Objective Programming in Supply Chain Management, *IMA Journal of Management Mathematics* (Under Review).

Kamran S. Moghaddam, John S. Usher, (2009) Optimal Preventive Maintenance and Replacement Schedules with Variable Improvement Factor, *Journal of Quality in Maintenance Engineering* (Accepted).

Kamran S. Moghaddam, John S. Usher, (2009) A New Multi-Objective Optimization Model for Preventive Maintenance and Replacement Scheduling of Multi-Component Systems, *Engineering Optimization* (Accepted).

Kamran S. Moghaddam, John S. Usher, (2009) Sensitivity Analysis and Comparison of Algorithms in Preventive Maintenance and Replacement Scheduling Optimization Models, *Computers and Industrial Engineering* (Under Review).

Kamran S. Moghaddam, John S. Usher, (2009) A Method for Predicting Optimal Preventive Maintenance Policy for a Repairable and Maintainable Series System of Components, *Computers and Industrial Engineering* (Under Review).

Seyed Jamshid Mousavi, **Kamran S. Moghaddam**, Abbas Seifi, (2007) Long-term Planning of Multi-Reservoir Operation, *Amirkabir University of Technology Scientific Journal* (In Persian).

Kamran S. Moghaddam, Abbas Seifi, Seyed Jamshid Mousavi, (2005) A Primal-Dual Algorithm for Optimal Operation of Multi-Reservoir Systems, *Iran Water Resources Research*, v 1, n 3, Fall 2005, p 16-28.

Seyed Jamshid Mousavi, **Kamran S. Moghaddam**, Abbas Seifi, (2004) Application of an Interior-Point Algorithm to the Optimization of a Large-Scale Reservoir System, *Water Resources Management*, v 18, n 6, December 2004, p 519-540.

REFEREED CONFERENCE PROCEEDING

Xu Yang, **Kamran S. Moghaddam**, (2010) Design and Analysis of a Multi-Product Assembly Line. *Proceedings of 2010 Decision Sciences Institute Annual Meeting*, San Diego, CA, USA (Under Review).

Kamran S. Moghaddam, John S. Usher, (2009) Maintenance Scheduling of Multi-Component Systems Using Multi-Objective Simulated Annealing, *Proceedings of the 2009 Industrial Engineering Research Conference*, May 30-June 3, 2009, Miami, FL, USA, p 2189-2194.

WORKING PAPERS

Kamran S. Moghaddam, John S. Usher, (2010) Preventive Maintenance and Replacement Scheduling: A Review paper.

Gail W. DePuy, Lijian Chen, **Kamran S. Moghaddam**, (2010) Chance Constraint Optimization Models in Farm Management.

PRESENTATION AND TALKS

Kamran S. Moghaddam, Preventive Maintenance and Replacement Scheduling: Models and Algorithms, *Engineering Exposition 2010*, March 6^{th}, 2010, University of Louisville, Louisville, KY, USA.

Kamran S. Moghaddam, John S. Usher, Sensitivity Analysis and Comparison of Algorithms in Preventive Maintenance and Replacement Scheduling Optimization Models, *INFORMS Annual Meeting 2009*, October 11-14, 2009, San Diego, CA, USA.

Kamran S. Moghaddam, Preventive Maintenance and Replacement Scheduling: Models and Algorithms, *8th Annual Doctoral Colloquium at the Industrial Engineering Research Conference 2009*, May 31, 2009, Miami, FL, USA.

Kamran S. Moghaddam, John S. Usher, Maintenance Scheduling of Multi-Component Systems Using Multi-Objective Simulated Annealing, *Industrial Engineering Research Conference 2009*, May 30-June 3, 2009, Miami, FL, USA.

Kamran S. Moghaddam, Preventive Maintenance and Replacement Scheduling For Multi-Component Systems, *Engineering Exposition 2009*, March 7^{th}, 2009, University of Louisville, Louisville, KY, USA.

Kamran S. Moghaddam, John S. Usher, Cost Optimal Maintenance and Replacement Scheduling, *INFORMS Annual Meeting 2008*, October 12-15, 2008, Washington D.C., USA.

Kamran S. Moghaddam, A Method for Predicting a Cost Optimal Preventive Maintenance Policy for a Repairable and Maintainable Series System of Components, October 1^{st} 2008, University of Louisville, KY, USA.

Kamran S. Moghaddam, Cost Optimal Maintenance and Replacement Scheduling For Multi-Component Systems, *Engineering Exposition 2008*, March 8^{th} 2008, University of Louisville, Louisville, KY, USA.

Abbas Seifi, **Kamran S. Moghaddam**, Seyed Jamshid Mousavi, Optimal Operation of Karoun-Dez Reservoir System, *The First International Industrial Engineering Conference*, July 13-14, 2004, Tehran, Iran.

Kamran S. Moghaddam, Seyed Jamshid Mousavi, Optimization of Karoun-Dez Reservoir System Operation, *The First National Congress of Civil Engineering*, May 10-12, 2004, Sharif University of Technology, Tehran, Iran.

Kamran S. Moghaddam, Simulation of Karoun-Dez Reservoir System Operation, *Student Conference in Industrial Engineering*, October 2003, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran.

SCIENTIFIC SERVICES

Paper Reviewer Jan 2010 - Present
Journal of Engineering Optimization

Paper Reviewer Feb 2009
2009 Industrial Engineering Research Conference, Miami, FL, USA

NON-SCIENTIFIC SERVICES

Treasurer Apr 2008 - Apr 2009
Iranian Student Organization, University of Louisville, Louisville, KY, USA

Treasurer Mar 2009 - Apr 2010
INFORMS Student Chapter, University of Louisville, Louisville, KY, USA

Representative of Department of Industrial Engineering Aug 2008 - Apr 2010
Graduate Student Council, University of Louisville, Louisville, KY, USA

COMPUTER SKILLS

Mathematical and Optimization Softwares: MATLAB, TOMLAB, LINGO, CPLEX, MPL
Simulation Softwares: ARENA
Statistical Analysis Softwares: MINITAB, SPSS, EVIEWS
Decision and Risk Analysis Softwares: EXPERT CHOICE, DPL
Project Management Softwares: MS-Project
Transportation and Traffic Engineering Softwares: SYNCHRO
Programming Languages: Visual Basic/VBA, Visual C/C++
Database Design Softwares: MS-Access, Visual FoxPro
General Purpose Softwares: MS-Excel, MS-Word, MS-PowerPoint, MS-Visio