DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

JOINT OPTIMIZATION OF SPARE PARTS INVENTORY AND MAINTENANCE POLICIES USING HYBRID GENETIC ALGORITHMS

by

Mehmet Ali ILGIN

July, 2006

İZMİR

JOINT OPTIMIZATION OF SPARE PARTS INVENTORY AND MAINTENANCE POLICIES USING HYBRID GENETIC ALGORITHMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Industrial Engineering, Industrial Engineering Program

by

Mehmet Ali ILGIN

July, 2006
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"JOINT OPTIMIZATION OF SPARE PARTS INVENTORY AND MAINTENANCE POLICIES USING HYBRID GENETIC ALGORITHMS"** completed by **MEHMET ALİ ILGIN** under supervision of **PROF. DR. SEMRA TUNALI** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof.Dr. Semra TUNALI

Supervisor

Asst.Prof.Dr.Latif SALUM

(Jury Member)

Asst.Prof.Dr. M.Evren TOYGAR

(Jury Member)

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude to Prof. Dr. Semra Tunalı for her academic guidance and enthusiastic encouragement throughout the research. Also, I wish to thank my parents who have been always an important source of support and encouragement.

Mehmet Ali ILGIN

JOINT OPTIMIZATION OF SPARE PARTS INVENTORY AND MAINTENANCE POLICIES USING HYBRID GENETIC ALGORITHMS

ABSTRACT

In general, the maintenance and spare parts inventory policies are treated either separately or sequentially in industry. Since the stock level of spare parts is often dependent on the maintenance policies, it is a better practice to deal with these problems simultaneously. In this study, a simulation optimization approach using hybrid genetic algorithms (HGA) has been proposed for the joint optimization of preventive maintenance and spare provisioning policies of a manufacturing system operating in automotive sector. The HGA is formed using the probabilistic acceptance rule of the Simulated Annealing (SA) within the Genetic Algorithm (GA) framework. The cost function is evaluated by integrating the GA with a simulation model of the motor block manufacturing line, which represents the manufacturing system behaviour with its maintenance, and inventory related aspects. Next, to further improve the performance of the GA developed, a set of experiments has been performed to identify appropriate values for the GA parameters (i.e. the size of the population, the crossover probability, and the mutation probability). Finally, various comparative experiments have been carried out to evaluate performance of both the pure GA and HGA.

Key Words: Spare Parts Inventory, Maintenance, Simulation, Genetic Algorithms, Simulated Annealing.

YEDEK PARÇA ENVANTER VE BAKIM POLİTİKALARININ BİRLİKTE OPTİMİZASYONUNDA MELEZ GENETİK ALGORİTMALAR

ÖZ

Genelde endüstride bakım ve yedek parça envanter politikaları birbirlerinden bağımsız veya sıralı olarak değerlendirilir. Ancak yedek parçaların envanter düzeyleri bakım politikalarıyla yakından ilgili olduğundan, bu problemlerin eş zamanlı olarak ele alınması daha doğru bir uygulamadır. Bu çalışmada, bir imalat sisteminin koruyucu bakım ve yedek parça envanter politikalarının birlikte optimizasyonu için melez genetik algoritmaları kullanan bir simulasyon optimizasyonu yaklaşımı önerilmiştir. Melez genetik algoritma, benzetimli tavlama yönteminin olasılıklı kabul kuralının genetik algoritma yapısı içinde kullanılması ile oluşturulmuştur. En iyi bakım ve yedek parça envanter politikalarını belirlemek üzere geliştirilen melez genetik algoritmanın performansını değerlendirmede bir maliyet fonksiyonu önerilmiş ve bu fonksiyona ilişkin hesaplamalar söz konusu imalat sisteminin bakım ve yedek parça envanter özelliklerini detaylı olarak yansıtan bir simulasyon modeli yardımıyla gerçekleştirilmiştir. Ayrıca; önerilen melez genetik algoritmanın performansını daha da iyileştirmek üzere bir dizi deneyler yapılmış ve populasyon büyüklüğü, çaprazlama oranı, mutasyon oranı gibi genetik algoritma parametreleri için en uygun değerler belirlenmiştir. Son olarak da çeşitli deneysel koşullar altında saf ve melez genetik algoritmaların performansları karşılaştırılmıştır.

Anahtar Sözcükler: Yedek Parça Envanteri, Bakım, Simulasyon, Genetik Algoritmalar, Benzetimli Tavlama.

CONTENTS

Page

CHAPTER ONE

INTRODUCTION

The extreme competition in today's global markets forces firms to increase the reliability and availability of their production plants. Increasing the availability of production plants requires the minimization of machines downtime. Significant improvements in the reduction of machines downtime is a direct result of effective maintenance policies. Spare parts availability and its prompt accession has a crucial impact on the success of maintenance policies. That is why determination of optimal spare parts inventory levels is a critical and important problem to be solved by production managers.

Preserving ample sizes of spare part inventories for immediate disposition whenever needed can be a logical solution to the spare parts availability problem. However, this solution entails a high stocking cost. Thus there must be a trade-off between overstock and shortages of spare parts which is an inventory planning problem with a maintenance scheduling aspect. A more cost effective solution of this problem can be obtained by joint, rather than separate or sequential optimization of maintenance and inventory policies. In this way, it is possible to make a trade-off between inventory and maintenance related costs. Many studies dealing with maintenance and inventory policies have been reported in the literature. However relatively little effort has been placed upon their joint optimization, which stimulates us to carry out this study.

The most commonly used approaches in the development of a possible spare provisioning decision model are simulation modelling and mathematical programming. Mathematical programming involves the development of mathematical models based on linear programming, dynamic programming, goal programming etc. However the mathematical model development for spare parts inventory management systems requires the use of some assumptions which damage the realism and reliability of these models.

The use of simulation modeling in spare parts inventory management problem represents a popular alternative to mathematical modeling since simulation has the ability of describing multivariate non-linear relations which can hardly be put in an explicit analytical form. However, simulation modelling is not an optimization technique. It is necessary to integrate the simulation model with an optimization tool.

That is why, in this study, firstly, a detailed simulation model describing the manufacturing system with its spare parts inventory and maintenance policy related aspects was developed. Then, a Genetic Algorithm (GA) was integrated with the model for the joint optimization of spare parts inventory and maintenance policies. Next, to further improve the performance of the GA developed, a set of experiments has been performed to identify appropriate values for the GA parameters (i.e. the size of the population, the crossover probability, and the mutation probability). The Hybrid Genetic Algorithm (HGA) is formed using the probabilistic acceptance rule of the Simulated Annealing (SA) within the GA framework and various experiments have been carried out to evaluate both the pure GA and HGA.

Considering the decreasing profit margins in automotive industry, it is very important to adopt a cost effective maintenance system to be competitive in today's global markets. We hope that, the joint optimization procedure suggested in this study will help to cut down the operational costs and enhance the company's competitiveness in the long run.

Following this introduction, Chapter 2 presents simulation optimization methods, provides brief information on the classical and metaheuristics-based simulation optimization methods, and particularly discusses the GA and SA methods that are employed in this thesis study.

In Chapter 3, information on the maintenance and spare parts inventory management is presented. The structure of a maintenance management system is described and the main types of maintenance policies are discussed. Moreover, this chapter presents distinctive characteristics of spare part inventories and the main

inventory control policies used for spare parts. Relevant literature on the optimization of maintenance and spare parts inventory policies and justification for carrying out this study is presented in Chapter 4. Chapter 5 presents the implementation of the proposed approach for joint optimization of spare parts provisioning and maintenance policies in an automotive company. The concluding remarks and future research directions are presented in Chapter 6.

CHAPTER TWO

SIMULATION OPTIMIZATION

A simulation optimization problem is an optimization problem where the objective function is a response evaluated by the simulation. In the context of simulation optimization, a simulation model can be thought of as a "mechanism that turns input parameters into output performance measures" (Law & Kelton, 1991). In other words, the simulation model is a function (whose explicit form is unknown) that evaluates the merit of a set of specifications, typically represented as a set of values (April et al., 2003). Two major classes of simulation optimization can be distinguished (April et al., 2003; Fu, 2002): Classical Approaches and Metaheuristics.

2.1 Classical Approaches for Simulation Optimization

Fu (2002) identifies 4 classical approaches for optimizing simulations:

- Stochastic approximation (gradient based approaches)
- Sequential response surface methodology
- Random search
- Sample path optimization

Stochastic approximation (StApp) algorithms attempt to mimic the gradient search method used in deterministic optimization. The procedures based on this methodology must estimate the gradient of the objective function in order to determine a search direction (April et al., 2003). The difficulty with StApp is that a large number of iterations of the recursive formula is needed to come up with the optimum (Tekin & Sabuncuoglu, 2004).

Sequential response surface methodology is based on the principle of building metamodels, but it does so in a more localized way. In other words, the metamodels do not attempt to characterize the objective function in the entire solution space but rather concentrate in the local area that the search is currently exploring (April et al., 2003).

Random search algorithms move iteratively from a current single design point to another design point in the neighborhood of the current point. The technique selects points at random from the overall search region (Smith, 1973). Since the search region contains a large number of combinations of p dimensional points, the procedure stops when a specified number of computer runs has been completed (Tekin & Sabuncuoglu, 2004).

Sample path optimization exploits the knowledge and experience developed for deterministic continuous optimization problems. The idea is to optimize a deterministic function that is based on n random variables, where n is the size of the sample path (April et al., 2003). Generally n needs to be large for the approximating optimization problem to be close to the original optimization problem (Andradottir, 1998).

Although classical optimization methods have received a fair amount of attention from the research community, they generally require a considerable amount of technical sophistication on the part of the user. Several of these methods such as design of experiments, gradient methods will be sensitive to local extrema, owing to the exploration strategy they use. Moreover, these methods are not easy to use or to implement in simulation packages. Leading commercial simulation software packages employ metaheuristics as the methodology of choice to provide optimization capabilities to their users. We explore this approach to simulation optimization in the next section.

2.2 Metaheuristic Approach for Simulation Optimization

Metaheuristics, in their original definition, are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space. Over time, these methods have also come to include procedures for overcoming the trap of local optimality in complex solution spaces. These procedures utilize one or more neighborhood structures as a means of defining admissible moves to transition from one solution to another or to build or destroy solutions in constructive and destructive processes (Glover & Kochenberger, 2002).

Various metaheuristics have been suggested for simulation optimization. Such methods include scatter search, genetic algorithms, simulated annealing, tabu search, and neural networks. Although these methods are generally designed for combinatorial optimization in the deterministic context and many not have guaranteed convergence, they have been quite successful when applied to simulation optimization (Olafson & Kim, 2002).

Scatter search is designed to operate on a set of points, called reference points, which constitute good solutions obtained from previous solution efforts. Notably, the basis for defining "good" includes special criteria such as diversity that purposefully go beyond the objective function value. The approach systematically generates combinations of the reference points to create new points, each of which is mapped into an associated feasible point. The combinations are generalized forms of linear combinations, accompanied by processes to adaptively enforce feasibility conditions, including those of discreteness (Glover, 1977). The following principles summarize the foundations of the Scatter Search methodology (Glover et al., 2000):

- Useful information about the form (or location) of optimal solutions is typically contained in a suitably diverse collection of elite solutions.

- When solutions are combined as a strategy for exploiting such information, it is important to provide mechanisms capable of constructing combinations that extrapolate beyond the regions spanned by the solutions considered. Similarly, it is also important to incorporate heuristic processes to map combined solutions into new solutions. The purpose of these combination mechanisms is to incorporate both diversity and quality.

- Taking account of multiple solutions simultaneously, as a foundation for creating combinations, enhances the opportunity to exploit information contained in the union of elite solutions.

Tabu search is a constrained search procedure, where each step consists of solving a secondary optimization problem. At each step, the search procedure omits a subset of the solution space to search. This subset changes as the algorithm proceeds and is usually defined by previously considered solutions, which are called the reigning tabu conditions (Glover & Laguna, 1997).

The main components of Tabu Search algorithm are the Tabu List Restrictions and the Aspiration Level of the solution associated with the recorded moves. Tabu List is managed by recording moves in the order in which they are made. Each time a new element is added to the bottom of a list, the oldest element on the list is dropped from the "top". The Tabu List must be small enough to allow the search to carefully scrutinize the certain parts of the solution space, yet large enough to prevent a return to a previously generated solution. Tabu restrictions are subject to an important exception. When a tabu move has a sufficiently attractive evaluation where it would result in a solution better than any visited so far, then its tabu classification may be overridden. A condition that allows such an override to occur is called an aspiration criterion (Glover et al., 1995).

A neural network, or neural net for short, is a problem-solving method based on a computer model of how neurons are connected in the brain. A neural network consists of layers of processing units called nodes joined by directional links: one

input layer, one output layer, and zero or more hidden layers in between. An initial pattern of input is presented to the input layer of the neural network, and nodes that are stimulated then transmit a signal to the nodes of the next layer to which they are connected. If the sum of all the inputs entering one of these virtual neurons is higher than that neuron's so-called activation threshold, that neuron itself activates, and passes on its own signal to neurons in the next layer. The pattern of activation therefore spreads forward until it reaches the output layer and is then returned as a solution to the presented input. Just as in the nervous system of biological organisms, neural networks learn and fine-tune their performance over time via repeated rounds of adjusting their thresholds until the actual output matches the desired output for any given input. This process can be supervised by a human experimenter or may run automatically using a learning algorithm (Mitchell, 1996, p. 52).

In this study, a simulation optimization approach using hybrid genetic algorithms has been proposed for the joint optimization of preventive maintenance and spare provisioning policies of a manufacturing system operating in automotive sector. Since the hybrid algorithm is formed using the probabilistic acceptance rule of the Simulated Annealing (SA) within the GA framework, the following sections present detailed information on GA and SA.

2.2.1 Genetic Algorithms

GAs search the solution space by building and then evolving a population of solutions. The main advantage of GAs over those based in sampling the neighbourhood of a single solution is that they are capable of exploring a larger area of the solution space with a smaller number of objective function evaluations. A more through discussion of GAs are given in the following section.

2.2.1.1 An Overview of the Genetic Algorithms

GAs are numerical optimization algorithms inspired by both natural selection and natural genetics and are used to search large, non-linear search spaces where expert

knowledge is lacking or difficult to encode and where traditional optimization methods fall short (Goldberg, 1989).

A GA operates on a population of individuals (chromosomes) representing potential solutions to a given problem. Each chromosome is assigned a fitness value according to the result of the fitness (objective) function. The selection mechanism favors individuals of better objective function value to reproduce more often than worse ones when a new population is formed. Recombination allows for the mixing of parental information when this is passed to their descendants, and mutation introduces innovation in the population. Usually, the initial population is randomly initialized and evolution process is stopped after a predefined number of iterations (Azzaro-Pantel et al., 1998). Figure 2.1 (Grupe & Jooste, 2004) shows the general working principle of GAs.

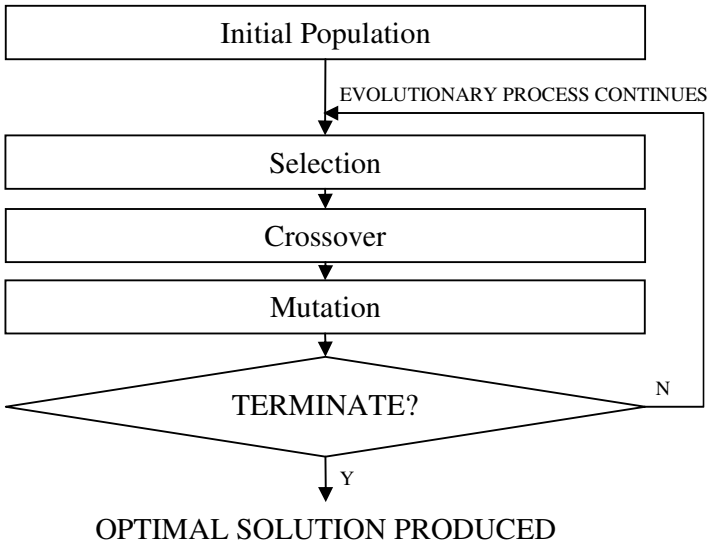

Figure 2.1 A GA illustrated

Because GAs are rooted in both natural genetics and computer science, the terminologies used in GA literature are a mixture of the natural and artificial (Gen & Cheng, 1997). The binary (or other) string can be considered to be a chromosome, and since only individuals with a single string are considered here, this chromosome is also the genotype. The organism, or phenotype, is then the result produced by the

expression of the genotype within the environment. In GAs this will be a particular set of unknown parameters, or an individual solution vector (Coley, 2003). Table 2.1 (Gen & Cheng, 1997) presents the explanations of the terms used in GAs.

Table 2.1 Explanation of genetic algorithm terms

Genetic Algorithms	**Explanation**
Chromosome (string, individual)	Solution (Coding)
Genes (Bits)	Part of the Solution
Locus	Position of Gene
Alleles	Values of Gene
Phenotype	Decoded Solution
Genotype	Encoded Solution

To understand the heuristic substructure of GAs it is important to understand the concepts given below:

- A genetic encoding of solutions to the problem
- A way of creating initial population
- A fitness function rating solutions in terms of their fitness
- Definition and implementation of genetic operators
- Termination criteria

These concepts are discussed in the following sections.

2.2.1.1.1 Encoding. Chromosome encoding depends on the problem to be solved. The format of the representation changes according to the type of the algorithm and the problem (Mitchell, 1996). The original formulation of GAs was based on the *binary encoding*. In binary encoding, each chromosome is a string of bits, 0 or 1. This encoding type gives many possible chromosomes even with a small number of alleles. On the other hand, this encoding is often not natural for many problems and sometimes corrections must be made after crossover and/or mutation. An alternative method for binary encoding is *gray coding*. This is similar to binary encoding except that each successive number only differs by one bit. *Direct value encoding* can be used in problems where some complicated value such as real numbers are used. The use of binary encoding for this type of problems would be difficult. In the value

encoding, every chromosome is a sequence of some values which can be anything connected to the problem, such as (real) numbers, characters or any objects. In *permutation encoding*, every chromosome is a string of numbers that represent a position in a sequence. Permutation encoding is useful for ordering problems. *Tree encoding* is used mainly for evolving programs or expressions, i.e. for genetic programming. In the tree encoding every chromosome is a tree of some objects, such as functions or commands in a programming language.

2.2.1.1.2 Creation of Initial Population. The initial population is usually generated randomly. There are also other alternatives. One of them is to carry out a series of initializations for each individual and then pick the highest performing values. Another alternative is to locate approximate solutions by using other methods (i.e., simulated annealing, tabu search) and to start the algorithm from such points (Coley, 2003). To generate initial population to be used in GAs neural networks are also employed (Reeves, 1995).

2.2.1.1.3 Fitness Function. Each chromosome is evaluated and assigned a fitness value after the creation of an initial population. The fitness function, also called payback function defines a fitness value for every chromosome in the population. On the basis of this value, the selection process decides which of the genomes are chosen for reproduction (Rutishauser, 2002).

The fitness function is a black box for the GA. Internally; this may be achieved by a mathematical function, a simulation model, or a human expert that decides the quality of a chromosome. At the beginning of the iterative search, the fitness function values for the population members are usually randomly distributed and wide spread over the problem domain. As the search evolves, particular values for each gene begin to dominate. The fitness variance decreases as the population converges. This variation in fitness range during the evolutionary process often leads to the problems of premature convergence and slow finishing.

Premature convergence occurs when the genes from a few comparatively fit (not optimal) individuals may rapidly come to dominate the population, causing it to converge on a local maximum. To overcome this problem, the way individuals are selected for reproduction must be modified. One needs to control the number of reproductive opportunities each individual gets so that it is neither too large nor too small. The effect is to compress the range of fitnesses, and prevent any "super-fit" individuals from suddenly taking over.

Slow finishing is the converse problem to premature convergence. After many generations, the population will have largely converged, but may still not have precisely located the global maximum. The average fitness will be high, and there may be little difference between the best and average individuals. Consequently there is an insufficient gradient in the fitness function to push the GA towards the maximum. The same techniques used to combat premature convergence also combat slow finishing. They do this by expanding the effective range of fitnesses in the population. As with premature convergence, fitness scaling can be prone to over compression due to just one "super poor" individual (Beasley et al., 1993).

2.2.1.1.4 Operators. The genes in chromosomes may be manipulated by three main operators:

- Selection
- Crossover
- Mutation

Selection is a process in which chromosomes are copied according to their fitness function value. There are many selection methods for selecting the best chromosome-such as: Roulette Wheel Selection, Boltzman Selection, Tournament Selection, Rank Selection, Steady State Selection and so on.

Selection provides the driving force behind the GA, and the selection pressure is critical in it. At one extreme, the search will terminate prematurely; while at the other

extreme progress will be slower than necessary. Typically, low selection pressure is indicated at the start of the GA search in favor of a wide exploration of the search space, while high selection pressure is recommended at the end in order to exploit the most promising regions of the search space (Gen&Cheng, 1997, p.20).

In roulette wheel selection, the size of each slice corresponds to the fitness of appropriate individual. The algorithm for the roulette wheel selection can be summarized as follows (Coley, 2003, p.24).

- Sum the fitness of all the population members. Call this sum f_{sum}.
- Choose a random number, R_s, between 0 and f_{sum}.
- Add together the fitness of the population members (one at a time) stopping immediately when the sum is greater than R_s. The last individual added is the selected individual and a copy is passed to the next generation.

Tournament selection is implemented by choosing some number of individuals randomly from the population and copying the best individual from this group into the intermediate population, and by repeating it until the mating pool is complete. Tournaments are frequently held only between two individuals. Bigger tournaments are also used with arbitrary group sizes (not too big in comparison with the population size). Tournament selection can be implemented very efficiently because no sorting of the population is required (Da Silva, 2002).

One potential advantage of tournament selection over all other forms is that it only needs a preference ordering between pairs or groups of strings, and it can thus cope with situations where there is no formal objective function at all — in other words, it can deal with a purely subjective objective function. It is also useful in cases where fitness evaluation is expensive; it may be sufficient just to carry out a partial evaluation in order to determine the winner (Reeves & Rowe, 2002, p.35).

Rank Based Selection assigns the individuals' selection probabilities according to the individuals' rank that is based on the fitness function values. There are two main

types of rank based selection. In linear ranking selection, the individuals are sorted according to their fitness values and the last position is assigned to the best individual, while the first position is allocated to the worst one. The selection probability is linearly assigned to the individuals according to their ranks. All individuals get a different selection probability, even when equal fitness values occur. Exponential ranking selection differs from linear ranking selection only in that the probabilities of the ranked individuals are exponentially weighted (Da Silva, 2002).

In steady-state selection, only a few individuals are replaced in each generation: usually a small number of the least fit individuals are replaced by offspring resulting from crossover and mutation of the fittest individuals. Steady-state GAs are often used in evolving rule-based systems (e.g., classifier systems) in which incremental learning (and remembering what has already been learned) is important and in which members of the population collectively (rather than individually) solve the problem at hand (Mitchell, 1996, p.171).

While creating the new population, the best individuals can be lost. To avoid this possibility, elitism is used. Elitism is a method that copies the best chromosome or a few best chromosomes to the new population. For many applications the search speed can be greatly improved by not losing the best or elite member between generations (Coley, 2003).

If during the early stages of a run, one particularly fit individual is produced, fitness proportional selection can allow a large number of copies to rapidly flood the subsequent generations. This can lead to premature convergence (Coley, 2003, p.153). Late in a run, there may still be significant diversity within the population; however, the population average fitness may be close to the population best fitness. If this situation is left alone, average members and best members get nearly the same number of copies in future generations, and survival of the fittest necessary for improvement becomes a random walk among the mediocre (Goldberg, 1989, p.77).

Scaling mechanisms are proposed to mitigate these problems. They include the mapping of raw objective function values to some positive real values. These real values are used to determine the survival probability of each individual. Fitness scaling has a two-fold intention (Gen & Cheng, 1997, p.25):

- To maintain a reasonable differential between relative fitness ratings of chromosomes.
- To prevent a too-rapid takeover by some super chromosomes in order to meet the requirement to limit competition early on, but to simulate it later.

Linear scaling computes the scaled fitness value as $f' = af + b$. where f is the fitness value, f' is the scaled fitness value, and a and b are suitably chosen constants. Here a and b are calculated in each generation to ensure that the maximum value of the scaled fitness value is a small number, say 1.5 or 2.0 times the average fitness value of the population. Then the maximum number of offspring allocated to a string is 1.5 or 2.0. Sometimes the scaled fitness values may become negative for strings that have fitness values less than the average fitness of the population. In such cases, we must recompute a, and b appropriately to avoid negative fitness values (Srivinas & Patnaik, 1994).

Linear scaling works well except when negative fitness calculation prevents its use. To circumvent this scaling problem, population variance information is used (Goldberg, 1989). In this method, which is called as *sigma truncation*, the fitness values of strings are determined as follows:

$$f' = f - (\bar{f} - c\sigma)$$

Where $\bar{f}$ is the average fitness value of the population, σ is the standard deviation of fitness values in the population, and c is a small constant typically ranging from 1 to 3.

Another possibility is power scaling, i.e., $f' = f^k$. In general, the k value is problem dependent and may require adaptation during a run to expand or compress the range of fitness function values. The problem with all fitness scaling schemes is that the degree of compression can be determined by a single extreme individual, degrading the GA performance (Da Silva, 2002).

Crossover is the primary genetic operator that permits new regions in the search space to be explored. Crossover combines the "fittest" chromosomes and passes superior genes to the next generation. It refers to the occasional crossing of two chromosomes in such a way that they exchange equivalent genes with one another.

One-point crossover takes two parents and randomly selects a point where the parents are split, and then the two parts of the parents after the selected point are swapped to make two children. Figure 2.2 shows this operation.

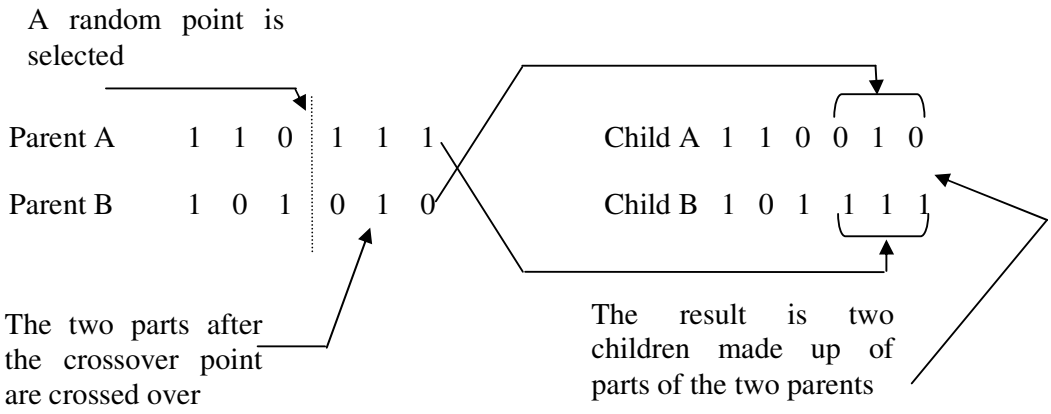

Figure 2.2 One Point Crossover

A more complex way of recombining the genes of a genotype is by using a multiple point crossover technique. The most common multiple point crossover technique is two-point crossover. In two point crossover, two crossover points are selected randomly within a chromosome then the two parent chromosomes between these points are interchanged to produce two new offspring.

Uniform Crossover is a crossover operator that decides (with some probability – known as the mixing ratio) which parent will contribute each of the gene values in the offspring chromosomes. This allows the parent chromosomes to be mixed at the gene level rather than the segment level (as with one and two point crossover). For some problems, this additional flexibility outweighs the disadvantage of destroying building blocks.

Following the creation of the new population, *the mutation* process is carried out in an effort to avoid local minima and to ensure that newly generated populations are not uniform and incapable of further evolution (Holland, 1992). In this process, a random number is generated in the interval [0, 1] and compared with a specified threshold value P_m: if it is less than P_m then mutation is carried out for that gene; otherwise the gene is skipped.

There are many different forms of mutation for different kinds of representation. In the case of binary encoding, mutation is carried out by flipping bits at random, with some small probability (usually in the range [0.001; 0.05]). For real-valued encoding, the mutation operator can be implemented by random replacement, i.e., replace the value with a random one. Another possibility is to add/subtract (or multiply by) a random (e.g., uniformly or Gaussian distributed) amount.

Mutation can also be used as a hill-climbing mechanism. In this case, mutation is done only if it improves the quality of the solution. Such an operator can accelerate the search. But, it might also reduce the diversity in the population and makes the algorithm converge toward some local optima.

Gaussian Mutation is a type of mutation used during genetic optimization. Gaussian mutation uses a bell-curve around the current value to determine a random new value. Under this bell-shaped area, values that are closer to the current value are more likely to be selected than values that are farther away.

2.2.1.1.5 Termination Criterion. Unlike simple neighbourhood search methods that terminate when a local optimum is reached, GAs are stochastic search methods that could in principle run forever. In practice, a termination criterion is needed; common approaches are to set a limit on the number of fitness evaluations or the computer clock time, or to track the population's diversity and stop when this falls below a preset threshold. The meaning of diversity in the latter case is not always obvious, and it could relate either to the genotypes or the phenotypes, or even, conceivably, to the fitnesses, but in any event we need to measure it by statistical means. For example, we could decide to terminate a run if at every locus the proportion of one particular allele rose above 90% (Reeves & Rowe, 2002).

2.2.1.2 Use of Genetic Algorithms in Simulation Optimization

Using simulation in the optimization process includes several specific challenges. Some of these issues are those involved in optimization of any complex and highly nonlinear function. Others are more specifically related to the special nature of simulation modeling (Azadivar, 1999). The major issues to address when comparing simulation optimization problems to generic non-linear programming problems are as follows (Azadivar, 1999; Paul & Chanev, 1998):

- There does not exist an analytical expression of the objective function or the constraints.
- The objective function(s) and constraints are stochastic functions of the deterministic decision variables.
- Performance measures could have many local extrema.
- The parameter space is not continuous. So there is often a need for discrete parameters such as integer, logical or linguistic.
- The search space is not compact. There could be zones of parameter values that are forbidden or impossible for the model.

The above list of features is a direct recommendation for the use of GAs, since they differ from conventional optimization and search procedures in several

fundamental ways (Ding et al., 2003; Gen & Cheng, 1997; Robert & Shahabudeen, 2004):

GAs use only objective function information to guide themselves through the solution space. So, they do not have much mathematical requirements about the optimization problems. The search for solutions will be guided without considering the inner workings of the problem. GAs can handle any kind of objective functions and any kind of constraints (linear or non-linear) defined on discrete, continuous, or mixed search spaces.

One of the most striking difference between GAs and most of the traditional optimization methods is that a GA works with a population of solutions instead of a single solution. Most classical optimization methods generate a deterministic sequence of optimization based on gradient or higher-order derivatives of the objective function. The methods are applied to a single point in the search space. The point is then improved along the deepest descending/ascending direction gradually through iterations. This point-to-point approach takes the danger of falling in local optima. GAs perform a multiple directional search by maintaining a population of potential solutions. The population-to-population approach attempts to make the search escape from local optima.

The other difference is that a GA uses an encoding of control variables, rather than the variables themselves. Encoding discretizes the search space and allows GAs to be applied to discrete and discontinuous problems. The other advantage is that GAs exploit the similarities in string-structures to create an effective search.

In addition to the above differences, GAs use probabilistic transition rules, as opposed to deterministic rules, to guide search. In early GA iterations, this randomness in GA operators makes the search unbiased toward any particular region in the search space. This avoids a hasty wrong decision and affects a directed search later in the optimization process. The use of stochastic transition rules also increases the chance of recovering from a mistake.

Researchers conducted various studies on the application of simulated based GAs for solving optimization problems in the area of scheduling (Fujimoto et al., 1995; Azzaro-Pantel et al., 1998; Lee & Kim, 2001; Breskvar & Kljajic, 2003; Cheu et al., 2004), facility layout (Azadivar & Wang, 2000), assembly line planning (Lee et al., 2000), kanban systems (Köchel & Nielander, 2002), and supplier selection (Ding et al., 2003).

Fujimoto et al. (1995) integrate GAs and simulation to seek the best combinations of dispatching rules in order to obtain an appropriate production schedule under specific performance measures. Based on the results obtained by the simulation, the authors indicate that the hybrid approach using the GA and simulation is more effective in searching for the best rule set for all combinations of dispatching rules.

Azzaro-Pantel et al. (1998) propose a two-staged methodology for solving job shop scheduling problem. The first stage involves the development of a discrete-event simulation model to represent dynamically the production system behaviour. In the second step, GAs are used to solve batch-scheduling problems. The authors apply this approach two case studies corresponding to a big example and a giant one. They report very good solutions in both cases, reducing considerably the search space.

Lee & Kim (2001) propose a method for the integration of process planning and scheduling using simulation based GAs. In this method a simulation module computes performance measures based on process plan combinations and those measures are fed into a GA in order to improve the solution quality until the scheduling objectives are satisfied. Computational experiments show that the proposed method provides improvements in scheduling objectives such as makespan and lateness.

Breskvar & Kljajic (2003) describe an approach to using simulation for multi-criteria scheduling optimization. In this study, a simulation model is used for fitness function computation of the GA as well as for visual representation of the process behaviour of a chosen schedule. They compare manual and simulation based GA

scheduling results. Based on these results, they conclude that the system utilizing GAs and simulation yields from 5 % to 15 % better scheduling within a shorter time compared to manual scheduling.

Cheu et al. (2004) introduce a hybrid GA-simulation methodology for scheduling of pavement maintenance activities involving lane closures, aiming to minimize the network total travel time. They demonstrate the application of this scheduling method through a hypothetical problem and report a 5.1% reduction in network total travel time.

Azadivar & Wang (2000) present an approach for solving facility layout optimization problems for manufacturing systems with dynamic characteristics and qualitative and structural decision variables. Their approach integrates GAs, computer simulation and an automated simulation model generator with a user-friendly interface. The simulation is considered as a function evaluator. The GA systematically searches and generates alternative layout designs according to the decision criterion specified by the user. The simulation model generator then creates and executes simulation models recommended by the GA and returns results to the GA. The test results demonstrate that the proposed approach overcomes the limitations of traditional layout optimization methods and is capable of finding optimal or near-optimal solutions.

Lee et al. (2000) apply GA based simulation optimization to optimize the operations of an assembly flow-line for refrigeration compressors. The line is modelled using simulation and GA is employed to optimize objective functions such as the throughput of the line, machine utilization and tardiness. They also discuss the influence of the size of the population, the crossover probability, the mutation probability and the number of elite chromosomes on the performance of the GA. With the optimized values of process times and speed of incoming conveyor, the authors report significant improvements in throughput and tardiness.

Köchel & Nielander (2002) investigate the problem of the optimal design of multistage systems with Kanban control mechanism. The optimization problem involves a general criterion function and takes the lot sizes as decision variables. In the study, results are reported for three examples that are all based on the same manufacturing system. These results demonstrate the usability of the proposed approach.

Ding et al. (2003) present a simulation optimization approach using GAs to the supplier selection problem. The proposed approach uses discrete event simulation for performance evaluation of a supplier portfolio and GA for optimum portfolio identification based on the performance indices estimated by the simulation model. A real life case study is presented and simulation results are given for the validation of the approach.

Paul & Chanev (1998) apply GAs to the problem of optimising a simplified steelworks simulation model. By taking into consideration four control variables (i.e., number of torpedoes, cranes, steel furnaces and volume of the torpedo) they try to minimize the cost of the proposed solution. They achieve a significant improvement in cost by setting control variables according to the results of the GA.

Pierreval & Tautau (1997) propose a new evolutionary algorithm (EA) to optimize both quantitative and qualitative variables. They focus on the general schema of the EAs given in Muhlebein (1997). The method is applied to a workshop producing plastic yoghurt pots. The near optimal solutions are compared with the results of an exhaustive search. The results indicate that the algorithm achieves reasonably good solutions.

Azadivar & Tompkins (1999) develop a methodology in which simulation models are automatically generated through an object-oriented process and responses are computed by the simulation model for a given set of decision factors. The responses are returned to the GA to be utilized in selection of the next generation of configurations. This method is applied to a manufacturing system where the decision

factors are the types of machines to purchase for each stage, routing for each part type, and layout plan for machines. The authors report that GA outperforms random sampling on three sample problems. They also indicate that the GA consistently achieves a larger fraction of the possible improvement.

Dümmler (1999) considers the problem of sequencing n lots, where each lot can be processed by any of m available cluster tools. The proposed method combines simulation and a GA to generate lot processing sequences. Based on the results of the several sample applications, the authors report that optimal or close-to-optimal sequences can be produced in short time by using the proposed method.

Spieckermann et al. (2000) present a simulation-based optimization approach for the body shop design problem. The approach is based on a combination of metaheuristics, such as GAs and simulated annealing, and simulation models of car body shops. The approach has been evaluated using a standard implementation of a simple GA as well as commercial packages of both metaheuristics. The authors undertake a comprehensive case study at a German car manufacturer to test their approach and report that metaheuristics are able to detect solutions that the manually guided local search procedure has not discovered.

Scheneider et al. (2000) present an approach that integrates human interaction with simulations and GAs for the repair time analysis problem in airbase logistics. The proposed approach consists of two main components. The first component, Solution Explorer (SE) enables analysts to rapidly study the solution space and optimize a set of initial design guesses. The second component, Interactive Analyzer (IA) uses data sets already selected as good solutions for a given system goal and allows the analyst to test the solutions under different and stressful conditions. The authors applied this approach for the repair time analysis of a selected aircraft. Based on the results of this application, they indicate that the overall effectiveness of both components is good.

Marzouk & Moselhi (2002) present a methodology for simulation optimization utilizing GAs and apply it to a newly developed simulation-based system for estimating the time and cost of earthmoving operations. Pilot simulation runs were carried out for all configurations generated by the developed algorithm, and a complete simulation analysis was then performed for the fleet recommended by the genetic algorithm. The numerical example presented by the authors demonstrates the different features of the algorithm and illustrates its capabilities in selecting near-optimum fleets that minimize total project cost.

2.2.2 Simulated Annealing

Simulated Annealing (SA) is a method based on Monte Carlo Simulation, which solves difficult combinatorial optimization problems. The name comes from the analogy to the behaviour of physical systems by melting a substance and lowering its temperature slowly until it reaches freezing point (Magoulas et al., 2002).

In the analogy between a combinatorial optimization problem and the annealing process, the states of the solid represent feasible solutions of the optimization problem, the energies of the states correspond to the values of the objective function computed at those solutions, the minimum energy state corresponds to the optimal solution to the problem and rapid quenching can be viewed as local optimization (Pham&Karaboga, 2000, p. 13).

At each iteration of a SA algorithm applied to a discrete optimization problem, two solutions generated by the objective function (the current solution and a newly selected solution) are compared. Improving solutions are always accepted, while a fraction of non-improving (inferior) solutions are accepted with a probability

$$p = \exp (-\delta f / T)$$

Where δf is the increase in f and T is a control parameter, which by analogy with the original application is known as the system *"temperature*" irrespective of the objective function involved.

The implementation of the basic SA algorithm is straightforward. The following figure (Busetti, 2000, p.2) shows its structure:

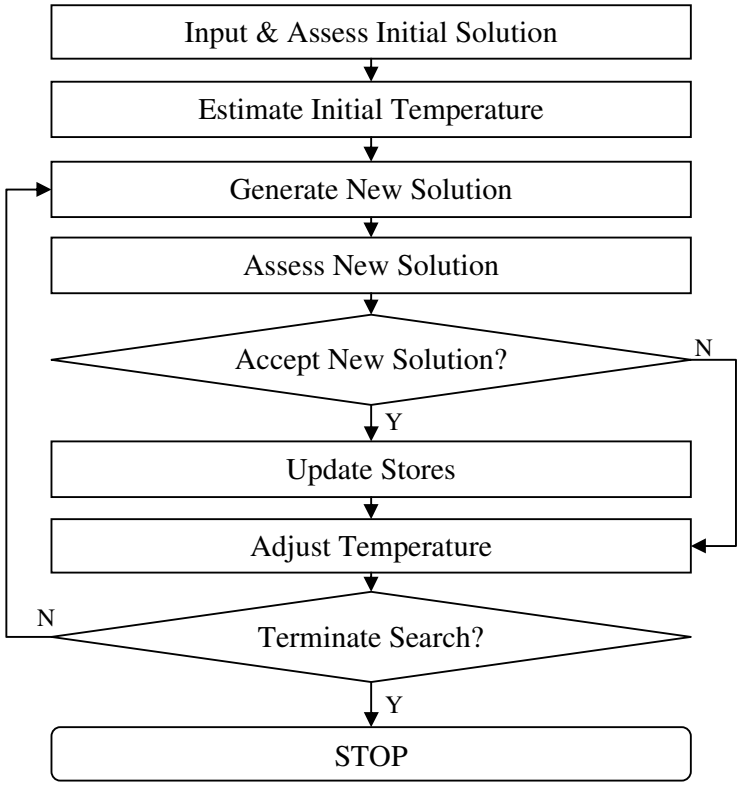

Figure 2.3 Structure of the simulated annealing algorithm

To apply SA to a problem, it is necessary to deal with the following issues:

- A representation of possible solutions
- A generator of random changes in solutions
- A means of evaluating the problem functions and
- An annealing schedule - an initial temperature and rules for lowering it as the search progresses

2.2.2.1 Solution Representation and Generation

When attempting to solve an optimization problem using the SA algorithm, the most obvious representation of the control variables is usually appropriate. However, the way in which new solutions are generated may need some thought. The solution generator should introduce small random changes, and allow all possible solutions to be reached (Busetti, 2000, p.6).

2.2.2.2 Solution Evaluation

Presented with a solution to a problem, there must be some way of measuring the quality of the solution. In defining this cost function we obviously need to ensure that it represents the problem we are trying to solve. It is also important that the cost function can be calculated as efficiently as possible, as it will be calculated at every iteration of the algorithm. If possible, the cost function should also be designed so that it can lead the search. One way of achieving this is to avoid cost functions where many states return the same value (Kendall, 2000).

The SA algorithm does not require or deduce derivative information; it merely needs to be supplied with an objective function for each trial solution it generates. Thus, the evaluation of the problem functions is essentially a `black box' operation as far as the optimization algorithm is concerned (Busetti, 2000).

2.2.2.3 Cooling Schedule

The cooling schedule of a SA algorithm consists of four components.

- Initial Temperature
- Final Temperature
- Temperature Decreasing Scheme
- Iterations at each temperature

We will consider these further below:

2.2.2.3.1 Initial Temperature. The process must start with a high initial temperature so that most if not all moves can be accepted – i.e. the initial temperature must be 'high'. In practice this may require some knowledge of the magnitude of neighbouring solutions; in the absence of such knowledge, one may choose what appears to be a large value, and run the algorithm for a short time and observe the acceptance rate. If the rate is 'suitably high', this value of T may be used to start the process. What is meant by a 'suitably high' acceptance rate will vary from one situation to another, but in many cases an acceptance rate of between 40% and 60% seems to give good results. More sophisticated methods are possible, but not often necessary (Rayward-Smith et al., 1996, p.9).

2.2.2.3.2 Final Temperature. It is usual to let the temperature decrease until it reaches zero. However, this can make the algorithm run for a lot longer, especially when a geometric cooling schedule is being used. In practice, it is not necessary to let the temperature reach zero because as it approaches zero the chances of accepting a worse move are almost the same as the temperature being equal to zero (Kendall, 2000).

To some extent, the determination of final temperature is problem dependent, and as in the case of selecting an initial temperature, may involve some monitoring of the ratio of acceptances (Rayward-Smith et al., 1996).

2.2.2.3.3 Temperature Decreasing Scheme. The way in which the temperature is decremented is critical to the success of the algorithm. Theory states that enough iterations at each temperature should be carried out so that the system stabilizes at that temperature. Unfortunately, theory also states that the number of iterations at each temperature to achieve this might be exponential to the problem size.

The most common temperature decrement rule is: $T_{k+1} = \alpha T_k$. Where α is a constant close to, but smaller than 1. This *exponential cooling scheme (ECS)* was first proposed with $\alpha = 0.95$. Typical values lie between 0.8 and 0.99.

In *linear cooling scheme (LCS),* T is reduced every L trials: $T_{k+1} = T_k - \Delta T$. The reductions achieved using the two schemes have been found to be comparable, and the final value of f is, in general, improved with slower cooling rates, at the expense of greater computational effort. The algorithm performance depends more on the *cooling rate* $\Delta T/L$ than on the individual values of ΔT and L. Obviously, care must be taken to avoid negative temperatures when using the LCS.

2.2.2.4 Use of Simulated Annealing in Simulation Optimization

SA has shown successful applications in a wide range of combinatorial optimization problems, and this fact has motivated researchers to use SA in simulation optimization.

Some of the researchers aimed at outlining and improving the general structure of simulation optimization based on SA. Haddock & Mittenthal (1992) use a heuristic cooling function in the SA based simulation optimization of a hypothetical system. Based on the experimental results, they indicate that a lower final temperature, a slower rate of temperature decrease, and large number of iterations performed at each temperature result in better solutions.

Jones & White (2004) explore an approach to global simulation optimization which combines StApp and SA. SA directs a search of the response surface efficiently, using a conservative number of simulation replications to approximate the local gradient of a probabilistic loss function. StApp adds a random component to the SA search, needed to escape local optima and forestall premature termination. They compare the performance of the proposed approach with the commercial package OptQuest.

Alkhamis & Ahmed (2004) develop a variant of SA for solving discrete stochastic optimization problems where the objective function is stochastic and can be evaluated only through Monte Carlo simulations. In the proposed variant of SA, the Metropolis criterion depends on whether the objective function values indicate statistically significant difference at each iteration. The differences between objective function values are considered to be statistically significant based on confidence intervals associated with these values. Unlike the original SA, the proposed method uses a constant temperature.

A number of studies applying SA-based simulation optimization has been noted in the literature. Brady & McGarvey (1998) integrate four heuristic optimization techniques namely SA, tabu search, GA, a frequency-based heuristic and a simulation model. The goal was to optimize the operating performance of a pharmeutical manufacturing laboratory in which a small set of operators service a larger set of testing machines. Barretto et al. (1999) apply a variant of the LinearMove and Exchange Move (LEO) optimization algorithm (Barretto et al., 1998) based on SA to a steelworks simulation model. Cave et al. (2002) present a SA based simulation optimization of a real scheduling problem in industry. They investigate the practicality of using SA to produce high-quality schedules. The experimental results of the optimization study were compared against average data collected during the operation of the system. This comparison shows that SA produces quality results with a low degree of variance.

2.2.3 Hybrid Genetic Algorithms

A hybrid GA combines the power of the GA with the speed of a local optimizer. The GA excels at gravitating the global minimum. However it is not especially fast at finding the minimum when in a locally quadratic region (Haupt & Haupt, 2004). There are many other, more efficient, traditional algorithms for climbing the last few steps to the global optimum. This implies that using a GA to locate the hills and a traditional technique to climb them might be very powerful optimization technique. (Coley, 2003).

The basic idea of hybrid GA is to divide the optimization task into two complementary parts. The coarse, global optimization is done by the GA while local refinement is done by the conventional method (e.g. gradient-based, hill climbing, greedy algorithm, simulated annealing, etc.). A number of variants is reasonable (Bodenhofer, 2003):

1. The GA performs coarse search first. After the GA is completed, local refinement is done.

2. The local method is integrated in the GA. For instance, every K generations, the population is mixed with a locally optimal individual.

3. Both methods run in parallel: All individuals are continuously used as initial values for the local method. The locally optimized individuals are re-implanted into the current generation.

One of the most common forms of hybrid GAs is to incorporate local optimization as an add-on extra to the simple GA loop of recombination and selection. With the hybrid approach, local optimization is applied to each newly generated offspring to move it to a local optimum before injecting it into population. GAs are used to perform global exploration among a population, while heuristic methods are used to perform local exploration among a population. (Gen &Cheng, 1997, p.31).

2.2.3.1 Hybridizing Genetic Algorithms and Simulated Annealing

GAs and SA are both independently valid approaches toward problem solving with certain strengths and weaknesses. While GA can begin with a population of solutions in parallel, it suffers from poor convergence. SA, by contrast, has better convergence properties, but it cannot easily exploit parallelism (Wang et al., 2005).

In order to retain the strengths of GA and SA, hybrid GA/SA blends both approaches into a single approach. GA/SA is naturally parallel by exploiting the

population-based model and recombination operators of GA. At the same time, GA/SA employs the temperature gradient property of SA by using a local acceptance policy based on the fitness of a new solution compared to its parent, and a probability based on a global temperature gradient (Shroff et al., 2002).

The structure of a GA hybridated by SA is as follows (Popa et al., 2002):

begin
Generate randomly the initial population chromosomes and establish the initial temperature T_0 ;
repeat
-calculate the fitness of chromosomes in current iteration;
repeat
-apply selection operator;
-apply crossover;
-apply mutation;
-calculate the fitness of chromosomes;
-the new chromosomes are accepted or not accepted in the new population;
until end of the number of new chromosomes
-update the population
-the temperature is decreased
until end of the number of iterations
end

The probability of acceptance is

$$Prob(\Delta E)=\exp(-\Delta E/T)$$

Where ΔE is the amount of deterioration between the new and old solutions and T is the temperature level at which the new solution is generated.

Acceptance probability will be low when the temperature is low. Some valuable chromosomes will be replaced during the entire period of evolution, but this change is greatly reduced towards the end of the process. In this way, sufficient diversity of chromosomes can be maintained and premature convergence can be eliminated.

CHAPTER THREE

MAINTENANCE MANAGEMENT & SPARE PART INVENTORIES

3.1 An Overview of Maintenance Management

Johnson (2002) defines maintenance management as "the recurring day-to-day, preventive or scheduled work required to preserve or restore facilities, systems and equipment to continually meet or perform according to their designed functions". According to Shenoy & Bhadury (1999) maintenance management can be defined as "a set of activities, or tasks, that are related to preserving equipment in a specified operating condition, or restoring failed equipment to a normal operating condition". The set of tasks or activities that constitute maintenance management ranges from simple cleaning operations and lubrication to performing condition monitoring, and planning and scheduling maintenance resources.

Maintenance activities should be managed properly for the company's success and for cost control. As companies become more automated, they increasingly rely on equipment to produce a greater percentage of their output. The cost of idle time gets higher as equipment becomes more specific and expensive. Also, more highly trained workers are needed, and the cost of managing spare parts is higher. So, to establish a competitive edge and to provide good customer service, companies should establish an effective maintenance management system.

3.1.1 Functions of Maintenance Management

As shown in Figure 3.1 (Shenoy & Bhadury, 1999), modern maintenance management involves the following functions (Shenoy & Bhadury, 1999):

- Maintenance planning
- Organizing maintenance resources, including staffing/recruiting
- Directing execution of maintenance plan
- Controlling the performance of maintenance activities

- Defining processes for performing maintenance
- Budgeting

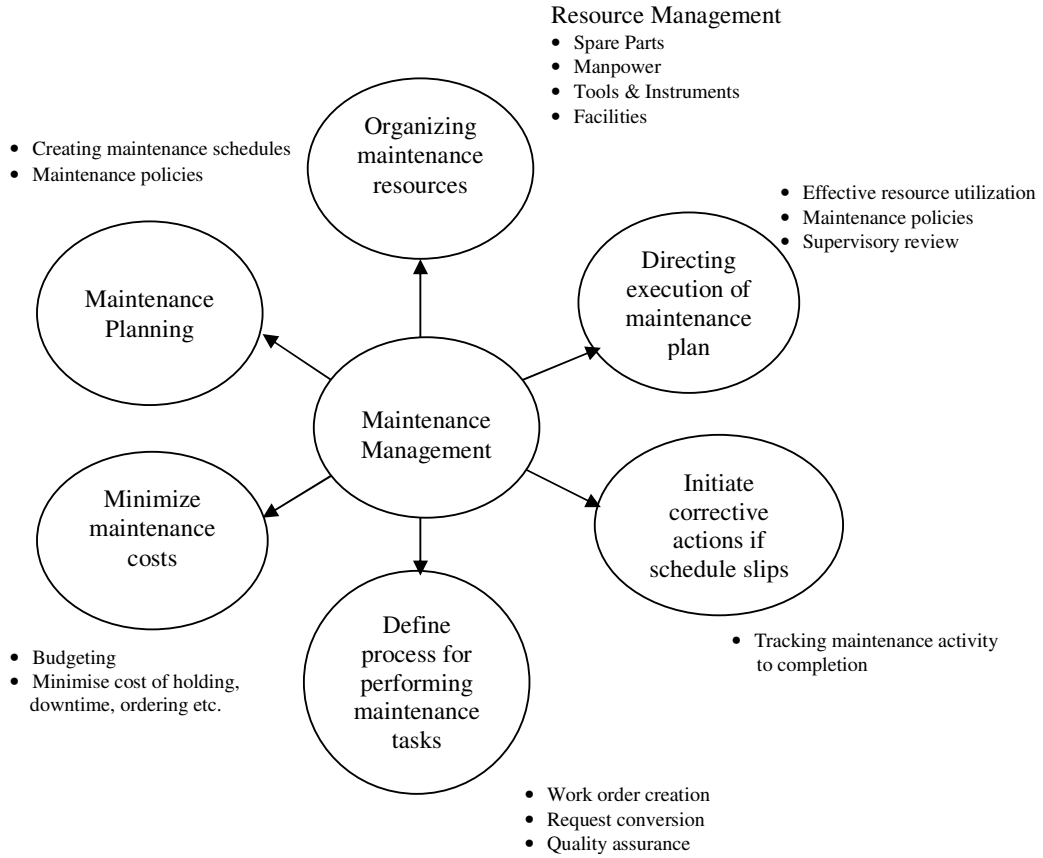

Figure 3.1 Functions of maintenance management

Planning and scheduling are vital for a successful maintenance program. The vast majority of maintenance work needs to be planned and scheduled so that the quality and cost-effectiveness of the operations is assured. Only emergency repairs can be carried out without advance planning and scheduling. In fact, these repairs also must be planned as they are taking place, operation by operation (Niebel, 1994).

Execution of maintenance activities as planned depends on the availability of required resources in the right quantity and at the right time. If these resources are unavailable, maintenance activities can not be performed as planned. This will result in degradation of equipment performance and can also result in its failure.

Once the required resources are available, the maintenance activity can be initiated. The maintenance department should ensure that the equipment is restored to its normal working condition as quickly as possible. This way not only is the downtime cost kept to the minimum but also the resources are utilised effectively. The maintenance work should be tracked to completion. After the completion of the maintenance activity, a review by the manager or the maintenance supervisor would be essential to ensure and authorise that the maintenance work has been carried out properly.

Other common tasks carried out by maintenance management comprise generating reports on equipment, work and costs. It also includes activities related to collection and analysis of maintenance data and reporting to top management.

3.1.2 Objectives of Maintenance Management

Effective maintenance management ensures the productivity of a company by influencing the percentage of time that its equipment can operate. Maintenance also influences the return on investment, since the economic lifetime and salvage value of equipment are affected by maintenance. The objectives of maintenance management include (Dilworth, 1992; Gaither & Frazier, 2002):

- Reduction of the frequency and severity of interruptions to production caused by machine malfunctions.
- Efficient use of maintenance personnel and equipment.
- Preserving the company's investment and prolonging the life of assets to increase the time over which investments provide service.
- Providing a safe working environment for workers.
- Improving product quality by keeping equipment in proper adjustment.

3.1.3 Maintenance Management Approaches

Four general approaches to maintenance management can be identified, namely breakdown, corrective, preventive and predictive maintenance.

3.1.3.1 Breakdown Maintenance

The logic of breakdown or run-to-failure maintenance is simple and straightforward. “When a machine breaks down, fix it”. This is a reactive maintenance management approach that waits for machine or equipment failure before any maintenance action is taken.

Because no attempt is made to anticipate maintenance requirements, a plant that uses true run-to-failure maintenance management must be able to react to all possible failures within the plant. This reactive method of management forces the maintenance department to maintain extensive spare part inventories or at least all major components for all critical equipment in the plant. The alternative is to rely on the equipment vendors that can provide immediate delivery of all required spare parts (Mobley, 2002, p. 3).

3.1.3.2 Corrective Maintenance

Repair is done after initiation of failure, leading to degraded performance. Usually condition monitoring or inspections reveal such degradation. The actual repair may be done before or after functional failure, based on the evaluation of consequences of failure, but the key difference from breakdown maintenance is this – the functional failure is known before it occurs, so there is an opportunity to schedule the repair.

A typical corrective maintenance process is given in Figure 3.2 (Honkanen, 2004). It begins with a failure or possible failure identification. The failure is then diagnosed. Diagnosing is an activity that may require several participants from specialists to production personnel. After the failure is diagnosed, repair is planned and materials are ordered. After that, the repair is scheduled, and the work is ordered. When the materials are available the machine is repaired according to the schedule (Honkanen, 2004, p. 27).

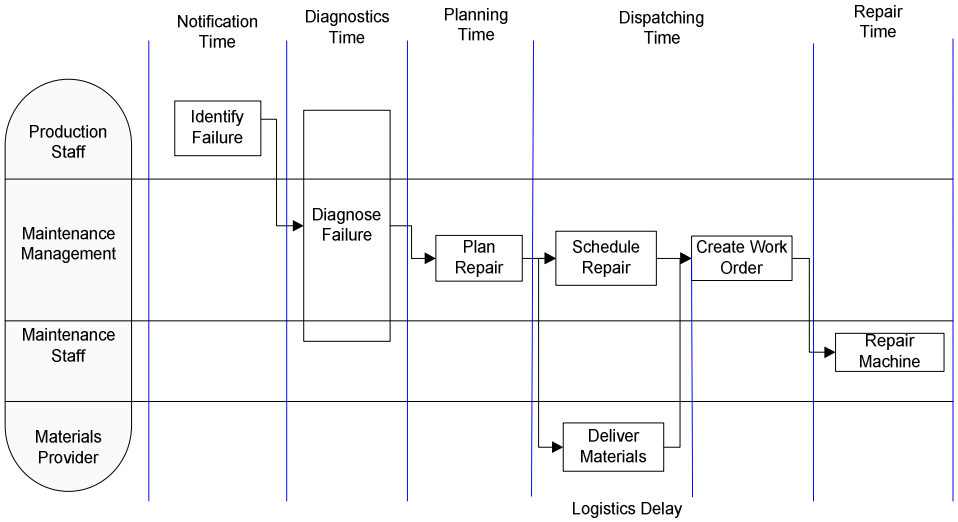

Figure 3.2 The Corrective maintenance process

3.1.3.3 Preventive Maintenance

Preventive maintenance (PM) consists of maintenance activities performed before equipment breaks down, with the intent of keeping it operating acceptably and reducing the likelihood of breakdown. The main purpose of PM is to extend equipment lifetime, or at least the mean time to the next failure whose repair may be costly. Furthermore, it is expected that effective PM policies can reduce the frequency of service interruptions and the many undesirable consequences of such interruptions.

A typical PM process is shown in Figure 3.3 (Honkanen, 2004). As opposed to the corrective maintenance process, it does not include the diagnostics stage. In practice, this is not always true. When using condition monitoring there may be signs of future failures, which makes it possible to predict a failure. The symptoms may be so clear that there is no need for diagnostics, but if the symptoms are unknown or contradictory there may be a similar diagnostics stage as in corrective maintenance. Failure prediction marks the beginning of a process as well as the preventive event that is created according to machine operating time or other usage measurement. If the process is well designed and pre-planned, the waiting time from the preventive

event to the beginning of maintenance execution should be much shorter than in the corrective process (Honkanen, 2004, pp. 27-28).

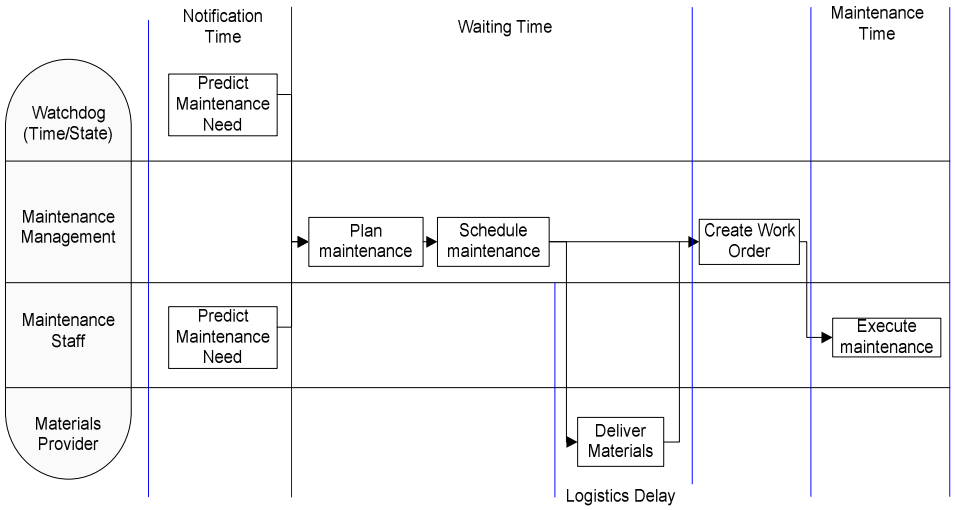

Figure 3.3 The preventive maintenance process

Compared to corrective maintenance, the system loss in PM is lower. There are three main reasons for this. First, PM can be arranged at convenient times when the system completes a production cycle or during shift changes. Such arrangement will usually result in less loss than corrective maintenance. Second, the chances of damaged products are significantly less. Third, it does not require emergency maintenance service which is generally very expensive (Peng, 1998).

PM clearly impacts on component and system reliability: if too little is done, this may result in an excessive number of costly failures and poor system performance and, therefore, reliability is degraded; done often, reliability may improve but the cost of maintenance will sharply increase. In a cost-effective scheme, the two expenditures must be balanced. Figure 3.4 describes this trade-off.

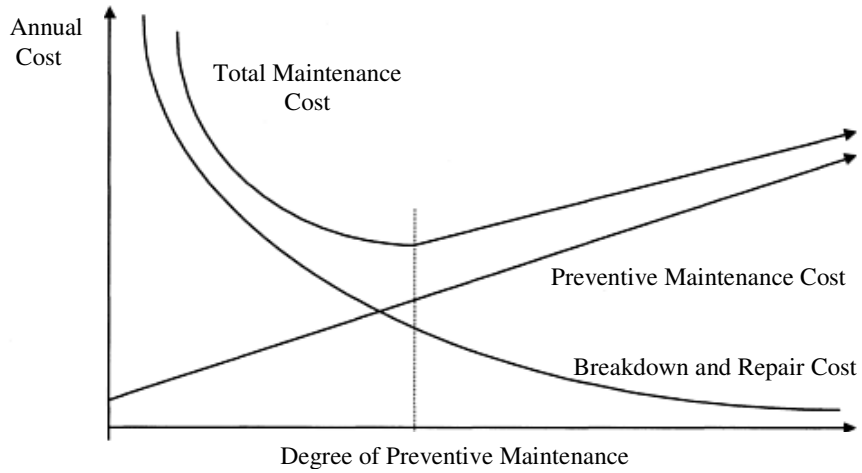

Figure 3.4 Level of maintenance

The actual implementation of PM varies greatly. Some programs are extremely limited and consist of only lubrication and minor adjustments. Comprehensive PM programs schedule repairs, lubrication, adjustments, and machine rebuilds for all critical plant machinery. The most common PM types are explained briefly in the following sections.

3.1.3.3.1 On-Condition. Repair is based on the result of inspections or condition-monitoring activities that are themselves scheduled on calendar time to discover if failure has already commenced. Vibration monitoring and on-stream inspections are typical examples of on-condition tasks. Monitoring of some parameters may be continuous, with the use of dedicated instrumentation. All on-condition maintenance is corrective in nature.

3.1.3.3.2 Condition Monitoring. Statistics and probability theory are the basis for condition monitoring maintenance. Trend detection through data analysis often rewards the analyst with insight into the causes of failure and preventive actions that will help avoid future failures. For example, stadium lights burn out within a narrow range of time. If 10 percent of the lights have burned out, it may be accurately assumed that the rest will fail soon and should, most effectively, be replaced as a group rather than individually (Patton, 1983, p.4).

3.1.3.3.3 Scheduled. Repair is done based on age (calendar time, number of cycles, number of starts or similar measures of age as appropriate). This strategy is applicable when the age at failure is predictable (i.e., the failure distribution curve is peaky). Fouling, corrosion, fatigue and wear related failures typically exhibit such distributions.

Scheduled, fixed-interval PM tasks should generally be used only if there is an opportunity for reducing failures that cannot be detected in advance, or if dictated by production requirements. The distinction should be drawn between fixed-interval maintenance and fixed-interval inspection that may detect a threshold condition and initiate condition monitoring tasks. Examples of fixed-interval tasks include 3,000-mile oil changes and 48,000-mile spark plug changes on a car, whether it needs the changes or not. This approach may be wasteful because all equipment and operating environments are not alike. What is right for one situation may not be right for another (Mobley, 2002, p.415).

3.1.3.4 Predictive Maintenance

Predictive maintenance is a condition-driven PM program. Instead of relying on industrial or in-plant average-life statistics to schedule maintenance activities, predictive maintenance uses direct monitoring of the mechanical condition, system efficiency, and other indicators to determine the actual mean-time-to-failure or loss of efficiency for each machine-train and system in the plant (Mobley, 2002, p.5).

Predictive maintenance is more feasible today because of technology that is available for equipment surveillance and diagnosis of problems while the machines are still running. The condition of a machine can be monitored by several means. Critical monitor points on equipment are identified. Sensors may be installed, or periodic readings may be taken with portable units to measure the temperature or vibration. Vibration sensors and ultrasonic sensors are used to feed data into a computer for analysis. Trends away from normal vibration patterns, which were recorded when the machine was working properly, are analyzed to determine where a

problem is developing and when it will become serious. Infrared imaging can detect areas that are unusually warm – another indication of a trouble spot (Dilworth, 1992, p.604).

3.2 Maintenance Spare Parts

"Spare parts" refer to the parts required for keeping owned equipment in healthy operating condition by meeting repair and replacement needs imposed by breakdown, preventive and predictive maintenance. The spare part management function is critical from an operational perspective especially in asset intensive industries such as refineries, chemical plants, paper mills, etc as well as organizations owning and operating costly assets such as airlines, logistics companies, etc (Kumar, 2003).

It must be noted that the spare parts inventories differ from other manufacturing inventories in several ways. First of all, the function of spare parts inventories is different from work-in-process (WIP) and finished product inventories. WIP inventories are used to smooth the production flow. Finished product inventories exist to protect against irregularities in lead time, differences in quality levels, and differences in machine production rates, labour troubles, scheduling problems, differences between capacity and demand and other well-known production characteristics. However, the function of spare parts is to assist the maintenance staff in keeping the equipment in operating condition.

Second, the policies that govern the spare parts inventories are different from those, which govern WIP and final product inventories. WIP and finished product inventories can be adjusted by altering production rates and schedules. But, the spare parts inventory levels are largely a function of usage and maintenance of equipment.

In order to have a greater understanding on spare parts inventories, the conditions that make them different from WIP or finished product inventories need to be described in more detail. These conditions are as follows (Kennedy et al., 2002);

- Maintenance policies, rather than customer usage dictate the need for spare parts inventories.
- Reliability information is generally not available to the degree needed for the prediction of failure times, particularly in the case of new equipment.
- Since part failures are often dependent, the dependence relation is needed to analyze the failures.
- Demands for spare parts are sometimes met through cannibalism of other spare parts or units.
- The shortage costs of a spare part generally include quality as well as lost production, and these costs are difficult to quantify.
- Obsolescence may be a problem as the machines for which the spare parts were designed become obsolete and are replaced.
- Components of equipment are more likely to be stocked than complete units if the major unit of equipment is expensive.

3.2.1 Types of Maintenance Spares

Maintenance has many different types of spare parts that need to be tracked through the inventory function. Wireman (2004) classifies them into eight categories:

- Bin Stock Spare Parts-Free Issue
- Bin Stock Spare Parts-Controlled Issue
- Critical or Insurance Spare Parts
- Rebuildable Spare Parts
- Consumables
- Tools and Equipment
- Residual or Surplus Parts
- Scrap or Useless Spare Parts

Bin stock spare parts are materials that have little individual value with high volume usage. They are usually stocked in an open issue area. The most common inventory control policy used for these items is the two-bin method.

Critical or insurance spare parts are those items that may not have much usage, but they must be kept in stock in case they are needed. Since the cost of these items is usually high on a per unit base, the inventory control policy of these items should be determined by taking a balance between the cost of lost production and inventory related costs.

Rebuildable spare parts include items that the repair cost is less than the cost to rebuild it. Depending on the size of the organization, the spare may be repaired by maintenance technicians, departmental shop personnel, or sent outside the company to a repair shop. These items are also generally high dollar spares and must be kept in good environmental conditions. Their usage, similar to the critical spares must be closely monitored and tracked. Lost spares of this type can result in considerable financial loss.

Consumables are items that are used up or thrown away after a time period. These items might include flashlight batteries, soap, oils, greases, etc. Their usage is tracked and charged to a work order number or accounting code. Historical records may be studied and charted to determine the correct levels of stock to carry for each item. If problems develop with the stock level, the inventory level can be adjusted on a periodic basis.

Some companies keep *tools and equipment* in the stores location or in a tool crib and issue them like inventory items. Unlike other spare part types, the tools are brought back when the job is finished.

When maintenance involves construction work, there are generally *surplus or residual materials* left over. These materials are usually stocked in maintenance stores. If the parts are not going to be used again in short term (1-6 months) they should be returned to the vendor for credit. If they are going to be used, or are critical spares, they should be assigned a stock number and properly stored.

3.2.2 Maintenance Spare Parts Inventory Policies

In any maintenance inventory system, the availability of spare parts when they are needed by the maintenance department has an immense importance. The objectives of effective spare parts inventory control are (Niebel, 1994):

- To relate stock and stores quantities to demand, thus avoiding both overstocking or under-stocking
- To avoid losses due to spoilage, pilferage, and obsolescence
- To obtain the best turnover rate on all items by considering both the costs of acquisition and possession

3.2.2.1 ABC Classification System

To best classify inventory and acquire the control needed in the least costly manner, Pareto's law is usually applied. This law emphasizes the fact that the significant items in a group usually constitute only a small portion of the total number of items in the group. Thus, the major proportion of the total inventory value will usually be comprised of as little as 10% of the items controlled. In order to determine the amount and type of control to establish on all the items inventoried, ABC analysis classifies all maintenance stores and stocks into three categories.

"A Items" would represent only between 10 and 15% of the total items yet their monetary value would be between 70 and 85% of the total investment in inventory. A items are high dollar, "insurance" type items that must be in stock. Orders for these items should be placed based on economic ordering quantity and strict control of the inventory should be maintained. Buffer stocks should be kept minimum to keep investment low.

"B Items" represent perhaps 20 to 30% of the items but about 25% of the total investment. Order quantities of B items will usually be larger than Class A items since the cost of possession of these items will be less.

"C items" represent maybe 60 to 70% of the items and about 10% of the investment. The procedure here will be to maintain a buffer stock to accommodate a reasonable period such as 10 weeks. Then periodically, perhaps every six months, reordering can take place (Niebel, 1994).

3.2.2.2 Two-Bin Inventory Control

In the two-bin system, each material has two bins that physically hold the material in a warehouse. As the material is used, material is withdrawn from a large bin until the large bin is empty. At the bottom of the large bin there is a preprinted requisition for another order of the material. This replenishment requisition is sent out, and in the meantime materials are used out of the small bin, which holds just enough material to last until the next inventory replenishment. When the inventory is replenished, a requisition is placed in the bottom of the large bin, both bins are filled, and the cycle is repeated (Gaither & Fraizer, 2002, p.359).

This system is useful for inexpensive items that cost more to count and monitor than it costs simply to use some approximate reorder level (Dilworth, 1993). On the less costly and frequently used Class B items and the vast majority of Class C items, control can be maintained using two-bin inventory control (Niebel, 1994).

3.2.2.3 Reorder Point/EOQ

A reorder point/economic order quantity (ROP/EOQ) system requires that for every item stocked in inventory where the EOQ formula is used, a predetermination is made of the minimum and maximum levels required. Also, it is assumed that lead-time is fairly constant. Assumptions for using the ROP/EOQ method are as follows:

- Item cost does not vary
- Order size does not vary
- Lead time is constant and known
- Storage costs are linear

The EOQ system is most effectively used with a central storehouse that must supply materials to a number of smaller storehouses. The central warehouse must have the capacity to store excess material until needed by the secondary storehouses. A typical reorder point calculation is shown below:

Reorder Point = Average Demand Rate x Lead Time + Safety Stock

(For 95% assurance, a safety factor of 1.65 is used; for 99% assurance, a safety factor of 1.96 is used.)

3.2.2.4 Min/Max (s, S) System

In this type of system, maximum and minimum numbers of units to be stocked are determined plus the amount of safety stock required until the next order is filled. Whenever the inventory on hand reaches this minimum stocking level or reorder point, an order is placed for the number of items necessary to reach the maximum stocking level. The advantage of a min/max system is that different minimum and maximum levels can be set for each class of items or for individual items if necessary. The major difference between the min/max and EOQ systems is that the size of each order can be varied based on need. The EOQ system assumes a stable and independent demand.

The min/max system is used for items where material demand (or usage) is constantly changing. It is based on the maximum and minimum amounts of material that the user (not a system calculation) determines.

CHAPTER FOUR

LITERATURE REVIEW

Since the aim of this study is the joint optimization of maintenance and spare parts inventory policies, in this section, a review of previous studies on management of maintenance and spare parts inventory policies is presented.

In the literature, the most commonly used approaches to develop a possible spare provisioning decision model are simulation and mathematical programming. Mathematical programming concerns the development of mathematical models based on linear programming, dynamic programming, goal programming etc. Multi-echelon technique for recoverable item control model of Sherbrooke (1968) is the first application of mathematical programming in spare parts inventory management problem. Following this study, several researchers studied different aspects of the spare parts management problem. The reader can refer to Kennedy et al. (2002) for an overview of these studies. It is noted that all these studies focus on the use of simplified plants' or systems' models whose predictions may be of questionable realism and reliability.

Another approach, which is commonly used to solve spare parts inventory management problem in industrial world is simulation modeling (Kabir & Al-Olayan, 1996; Sarker & Haque, 2000). The main advantage of simulation modeling over mathematical modeling is its ability to describe multivariate non-linear relations, which can hardly be put in an explicit analytical form. However, simulation modeling is not an optimization technique. If the objective is to develop optimal spare parts inventory policies using simulation, then it is necessary to integrate the simulation model with an optimization technique. In simulation optimization, one or more discrete event simulation models replace the analytical objective function and constraints. The decision variables are the conditions the simulation is run under, and the performance measure becomes one (or a function of several) of the responses generated by a simulation model (Azadivar & Tompkins, 1999). The classical methods used with simulation are response surface methodology (Gharbi & Kene,

2000), design of experiments (Chien et al., 1997), and stochastic approximation (Rossetti & Clark, 1998).

Kabir & Al-Olayan (1996) proposed a jointly optimized age based replacement and ordering policy using simulation. They employed a 5-factor second order rotatory design to select ranges for the replacement interval, stocking level and replenishment level over which the total cost of replacement is minimized.

Sarker & Haque (2000) extended Kabir & Al-Olayan (1996) by considering replacement durations of the operating units with spare parts. Since their aim is the joint optimization of block replacement and spare provisioning policy, the simulation model of the manufacturing system is developed to include both the maintenance and inventory related functions. The authors compare jointly optimized policies with separately optimized policies in terms of cost effectiveness. They also analyze the effect of cost and statistical parameters on the cost effectiveness of jointly and separately optimized policies. For all scenarios studied, the jointly optimized policy yields better and cost effective solutions than the combination of the separately optimized policies.

In both of these studies, the authors specified all experimental design points prior to experimentation process. In other words, they did not integrate the simulation model with any guided search method to decide on which factor levels to run in the next experiment so that the danger of falling in local optima can be avoided.

In recent years, metaheuristics such as Genetic Algorithms (GAs), Simulated Annealing (Haddock & Mittenthal, 1992), and Tabu Search (Yang et al., 2004) have been extensively used along with simulation to enhance the efficiency of the search procedure. Among these guided search methods, simulation optimization via GAs is a quite active research area. There are successful applications of GA based simulation optimization in scheduling (Azzaro-Pantel et al.,1998; Fujimoto et al., 1995; Lee & Kim, 2001), facility layout (Azadivar & Wang, 2000), assembly line planning (Lee et al., 2000), supply chain management (Ding et al., 2003), kanban

systems (Köchel & Nielander, 2002), maintenance policy selection (Azadivar & Shu, 1998; Marsequerra et al., 2002; Robert & Shahabudeen, 2004), and spare parts inventory management (Marsequerra et al., 2001; Marsequerra et al., 2005) . This study particularly deals with maintenance and spare parts inventory policy optimization using GAs.

Azadivar & Shu (1998) investigated the performance of five different maintenance policies based on the desired service level of a production system. Since the system performance depends on a combination of qualitative and policy variables (the choice of the maintenance policy) as well as a set of quantitative variables (allowable buffer spaces), a simulation optimization procedure based on GAs was developed and applied to four different problems ranging from a very simple to a very complex system. The authors also compare the performance of GA to random search and reported that GA performed relatively better than the random search and its superiority became rather remarkable as the problem size increased.

Marsequerra et al. (2002) applied GA based simulation optimization to determine the optimal on-condition maintenance strategy in terms of the thresholds of components beyond which maintenance has to be performed. The problem was framed as a multi-objective search aiming at simultaneously optimizing two typical objectives of interest, profit and availability. The approach has been applied to a very simple system for which analytic solution was feasible. The results obtained analytically were compared to those obtained by the GA approach and confirmed the good performance of the methodology implemented.

Robert & Shahabudeen (2004) evaluated multiple corrective maintenance policies to suggest the best policy for a Reactor-Regenerator system of a fluid catalytic cracking unit (FCCU). They employed Analysis of Variance to determine the best GA parameters (i.e., population size, number of generations, mutation rate, and crossover rate). Based on these optimized parameter values, the best corrective maintenance policy to achieve maximum system availability with minimum total maintenance cost was then determined.

In these three studies, the optimal maintenance policies were developed under the assumption that the required spare parts will be immediately available. During this literature survey we also noted another group of studies which solely focused on the development of spare parts inventory policy by ignoring the effect of maintenance policies.

Marsequerra et al. (2001) proposed a GA based simulation optimization approach for the determination of spare parts inventory levels required by a multi component system. They considered the net profit achievable during a given mission time as objective function and used simulation to determine the objective function values of various alternative spare part allocation schemes. The proposed approach was verified on a simple system.

In a following study, the authors (Marsequerra et al., 2005) extended their previous work (Marsequerra et al., 2001) to a multi-objective optimization problem involving maximization of the net profit of the system and minimization of the total volume of the spare parts. The comparison of two alternative solutions with respect to these objectives was achieved through the use of the concepts of the Pareto optimality and dominance. The authors gave a good example of GA based simulation optimization in spare parts inventory management, but they did not take into consideration some practical aspects such as age related failure processes and maintenance-driven spare demands.

The influence of maintenance policies on the spare provisioning policy cannot be ignored, since the need for spare parts is directly dictated by the maintenance policies. Considering the fact that the PM is scheduled, the demand for spare parts is predictable. For a machine breakdown, which requires unplanned repair, the stock-outs of spare parts cause the production to stop with significant costs. We noted only one study (Shum & Gong, 2005) which explicitly considers both maintenance and spare parts inventory management using GA. Shum & Gong (2005) proposed a GA for the joint optimization of maintenance and spare part purchasing policies. The maintenance policy proposed in this study included both frequency of PM and

maintenance workforce level. The authors utilized an analytical objective function to evaluate the performance of alternative policies under some simplified assumptions. Namely, they ignored the replacement times of spare parts, the probabilistic nature of spare part demand, and shortage and emergency ordering costs of spare parts.

Noting only one study for joint optimization of maintenance and spare parts inventory policies using GA, we can state that this area needs further attention. So, to fill the perceived gap in this area, we not only dealt with these problems simultaneously by proposing a GA, but we also employed a detailed simulation model of the manufacturing line as a fitness function evaluator. A simulation based fitness function evaluator enables us to capture all dynamic and stochastic aspects of the system such as age related failure processes, maintenance driven spare demand, spare part shortages, and emergency orders.

Joint optimization of maintenance and spare parts inventory policies usually leads to complex optimization problems where the criterion function possesses no analytically trustable form and owns many local optima. In such cases the estimation of the values of the criterion function by simulating the corresponding system and the search for an optimal solution by GAs has been proved to be a powerful approach. However, GAs suffer from poor convergence properties. SA, on the other hand, has better convergence properties, but it cannot easily exploit parallelism. So, by hybridizing GAs and SA, the strengths of both algorithms can be retained.

The purpose of this study is to demonstrate how simulation optimization based on hybrid GAs can be used to achieve the joint optimization of maintenance and spare parts inventory policies. In order to be competitive in todays global markets, firms must adopt a cost effective spare parts inventory management system since the shortages of spare parts when they are needed by the maintenance department often result in costly plant unavailabilities. We hope that, the joint optimization procedure suggested in this study will be effective in enhancing the company’s competitiviness in the long run by cutting down the operational costs.

CHAPTER FIVE
JOINT OPTIMIZATION OF SPARE PARTS INVENTORY AND MAINTENANCE POLICIES FOR AN AUTOMOTIVE COMPANY

The objective of our work is to suggest a Hybrid Genetic Algorithm (HGA) for joint optimization of spare part provisioning and maintenance policies of an automotive factory by minimizing the associated cost. The cost function is evaluated by integrating the GA with a simulation model of the motor block manufacturing line, which represents the manufacturing system behaviour with its maintenance, and inventory related aspects. Next, to further improve the performance of the GA developed, a set of experiments has been performed to identify appropriate values for the GA parameters (i.e. the size of the population, the crossover probability, and the mutation probability). The HGA is formed using the probabilistic acceptance rule of the Simulated Annealing (SA) within the GA framework and various experiments have been carried out to evaluate both the pure GA and HGA.

5.1 Problem Statement

This study particularly focuses on operations of motor block manufacturing line in an automotive factory and suggests an integrated approach to develop optimal policies for maintenance and spare parts inventory management.

The manufacturing process in motor block line begins with the arrival of block castings from the foundry and various operations like milling, drilling are carried out. The arrival rate of block castings is constant and is equal to 16 castings per day. At the end of the process, the completed motor blocks are sent to the motor assembly storage area. Information about these operations and the precedence relations can be found in Appendix (see Table A1 and Figure A1).

The operation of the motor block line is affected by the efficiency of Preventive Maintenance (PM) and Breakdown Maintenance (BM) (i.e. occurring due to unexpected machine breakdowns) activities. The efficiency of these maintenance

activities, in turn, depends directly on the availability of spare parts. The spare parts, which are associated with the critical machines (see Table 5.1) are provisioned according to continuous review inventory system. To control the reorder and maximum inventory levels for these spare parts, the company just relies on the intuition and experience of the maintenance personnel. Table 5.2 and Table 5.3 list the spare parts inventory levels and PM intervals currently practiced in the company, respectively. During the field studies in the production floor, it has been observed that, from time to time, this practice leads to stock-out incidences and at some other times it creates overstocking of spare parts. This is a typical inventory management problem whose optimum solution requires minimizing the cost of shortage, holding and ordering.

Table 5.1 Machines subject to PM and associated spare parts

Machine Code	Spare Part Codes
M01	SP01, SP02, SP03
M03	SP02, SP03, SPR04, SP05, SP06
M07	SP05, SP06, SP07, SP08, SP09, SP10, SP11, SP12
M08	SP08, SP13
M09	SP14, SP15, SP16
M12	SP17, SP18

Table 5.2 Current reorder and maximum stock levels for spare parts

Spare Part Code	s	S	Spare Part Code	s	S	Spare Part Code	s	S
SP01	2	5	SP07	1	3	SP13	2	6
SP02	1	2	SP08	3	6	SP14	1	2
SP03	1	2	SP09	2	4	SP15	2	4
SP04	2	4	SP10	3	5	SP16	2	5
SP05	3	5	SP11	3	5	SP17	2	5
SP06	1	2	SP12	1	2	SP18	1	2

Table 5.3 Current PM intervals for the machines

Machines	M01	M03	M07	M08	M09	M12
PM Intervals	1350	1450	1350	1350	1450	1350

The demand for the spare parts originates from two sources: 1. Preventive Maintenance 2. Breakdown Maintenance. While the first source of demand is deterministic (i.e., PM is carried out in pre-specified time intervals), the second

source is subject to randomness. Hence, overall, the aggregate demand for spare parts is stochastic. Therefore, an optimal spare part inventory management system should consider this stochastic behavior of the demand for spare parts.

The development of mathematical models based on linear programming, dynamic programming, goal programming etc. for such systems requires the use of simplified plants' or systems' models whose predictions may be of questionable realism and reliability. The use of simulation modeling in spare parts management problem as an alternative to mathematical modelling represents a popular approach in industrial world. The main advantage of simulation modeling over mathematical modeling is its ability to describe multivariate non-linear relations which can hardly be put in an explicit analytical form. However, simulation modelling is not an optimization technique. If the objective is to develop optimal spare parts inventory policies using simulation, then it is necessary to integrate the simulation model with an optimization technique.

In this study, we suggest a methodology, which aims at jointly optimizing the spare parts provisioning and also maintenance policies of an automotive company. Particularly, we suggest a Hybrid Genetic Algorithm, which is formed using the probabilistic acceptance rule of Simulated Annealing (SA) within the GA framework. It must be noted that before the GA is integrated with the SA, the control parameters of the GA (i.e., the size of the population, the crossover probability, and the mutation probability) are optimized. The objective function, which involves various cost components, is evaluated by integrating the GA with a simulation model of motor block line. This simulation model is developed in detail to realistically represent the manufacturing system behaviour with its inventory and maintenance related aspects. Lastly, various set of experiments have been carried out to evaluate the performance of both pure GA and also hybrid GA.

5.2 Proposed Hybrid Approach

The primary objective of this study is to develop a procedure that will be effective and efficient in the search for the optimum levels of PM intervals for the machines

and the optimum inventory levels for the critical spare parts. This section presents the proposed GA/SA algorithm for joint optimization of maintenance and spare parts inventory policies. The motivation behind integrating the GA with the SA is to exploit the power of GA to work on a solution in a global sense while allowing SA to locally optimize each individual solution.

During the search process, the performance of alternative inventory management and maintenance policies is evaluated using the following Total Annual Cost (TAC) function:

$$TAC = C_H + C_R + C_E + C_S + C_P \quad (1)$$

$C_H = \sum_{j=1}^{N_{sp}} H_j$ is the cost associated with holding N_{sp} number of spare parts. H_j being the cost of managing spare part j throughout a year.

$C_R = \sum_{j=1}^{N_r} R_j$ is the cost associated with N_r number of regular orders given throughout a year. R_j being the cost of placing regular order j.

$C_E = \sum_{j=1}^{N_e} E_j$ is the cost associated with N_e number of emergency orders given throughout a year. E_j being the cost of placing emergency order j.

$C_S = \sum_{j=1}^{N_{out}} D_j \cdot S$ is the cost associated with N_{out} number of stockouts of critical spare parts. D_j is the duration of stockout j. S is the cost of stockout per unit of time.

In this study S is assumed to be equal to the gross revenue per minute (Westerkamp, 1998).

$C_P = \sum_{j=1}^{N_P} T_j \cdot P$ is the cost associated with N_P number of PM instances. T_j is the duration of PM activity j. P is the cost of PM per unit of time.

The various terms of the objective function (1) lend themselves to an analytical formulation only provided that some simplifying assumptions are made. If a more realistic modeling of the system is required, the only feasible approach for the evaluation of the objective function (1) is the discrete event simulation modelling. That is why, in the proposed approach, the fitness of each possible solution is evaluated by a detailed simulation model of the motor block line. Based on the fitness results generated by this simulation model, the GA creates new sets of solutions. So, there is a two-way communication between the GA and the simulation model.

Figure 5.1 shows the control structure of the proposed approach. First, an initial population is generated randomly and genetic operators are applied to these solutions iteratively. The resulting candidate solutions are inserted into the population pool in a controlled manner, by taking into account of their total annual cost value and the stage reached in the evaluation process. The probabilistic acceptance approach of the simple SA is incorporated into the GA to decide whether a candidate solution should be included in the population pool. This is expressed by

$$\text{Prob } (\Delta E)=\exp (-\Delta E/T), \qquad (2)$$

Where ΔE is the increase in the total annual cost value of the new solution and T is the temperature, which defines the stage reached in the process. The temperature is initially fixed to a large value and is reduced gradually according to a cooling schedule as the algorithm progresses. As the temperature is reduced, the probability of accepting non-improved candidate solutions during the GA/SA process is reduced.

At the beginning of the search process candidate solutions are accepted with a high probability. In the latter stages however, the GA/SA approach is constrained to a local search space due to the reduction in the probability of accepting non-improved solutions.

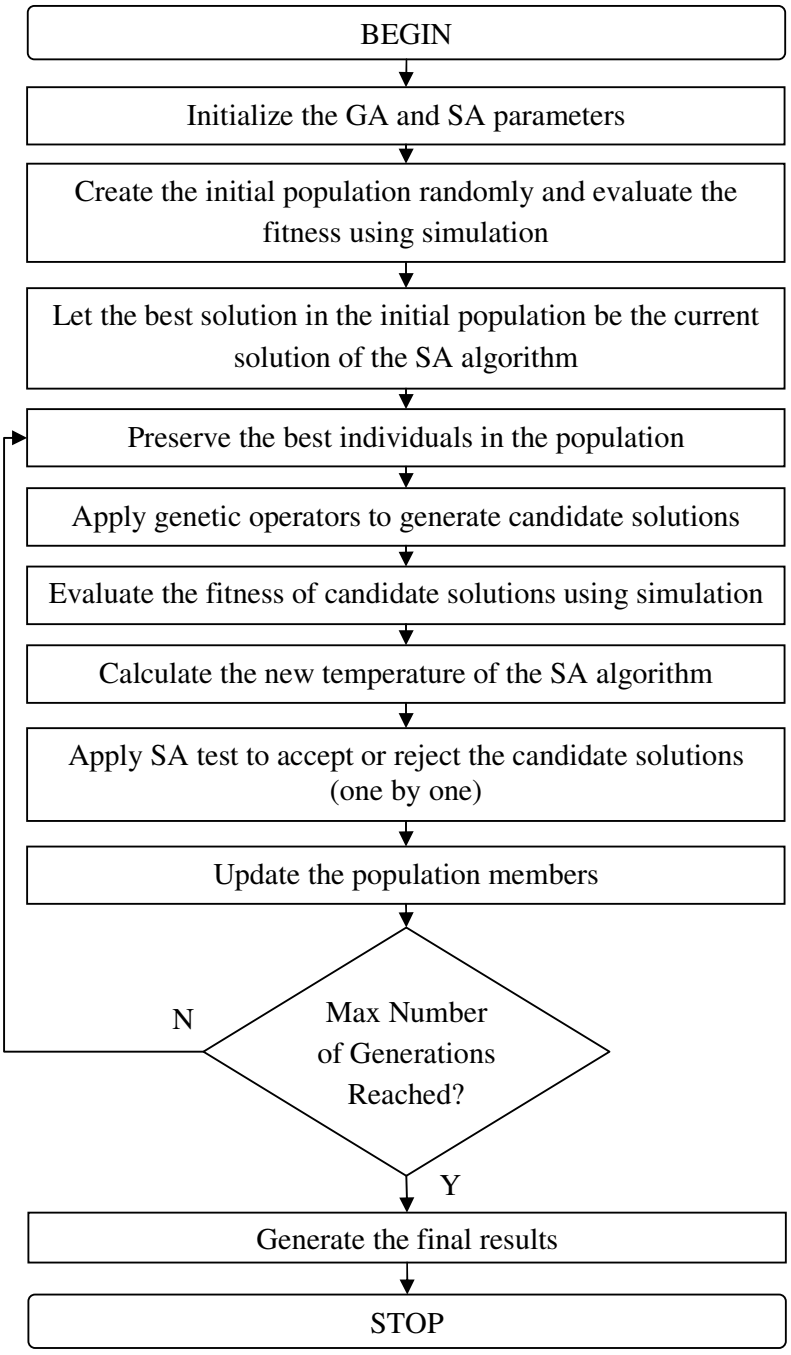

Figure 5.1 Flow Chart of HGA

5.2.1 Design of the Genetic Algorithm

The development of a GA to solve a particular problem involves two types of decisions. The first decision concerns the way in which the problem is to be modelled to fit into the GA framework and includes the definition of the range of feasible solutions, the form of the fitness function and the way in which individuals are to be represented as chromosomes. The second decision involves the parameters of the GA itself and includes the proportion of population to be produced as a result of reproduction, crossover and mutation, selection procedure, population size, number of generations, and a number of other decisions concerning variants of the basic algorithm.

In the following sections, firstly, the design details of the GA are presented. Then the results of experiments carried out for the optimization of the GA parameters are discussed.

5.2.1.1 Chromosome Representation

While designing the GA, at first, the reorder and maximum stock levels for critical spare parts and the PM intervals of the machines were coded into chromosomes so as to perform the genetic operation. So, each chromosome represents a possible configuration of the reorder, maximum stock levels of critical spare parts and the PM intervals for the machines. An example chromosome structure is given in Figure 5.2. In this figure, s_j, S_j, T_k represent the reorder, maximum stock levels of the spare parts, and the PM intervals for the machines, respectively.

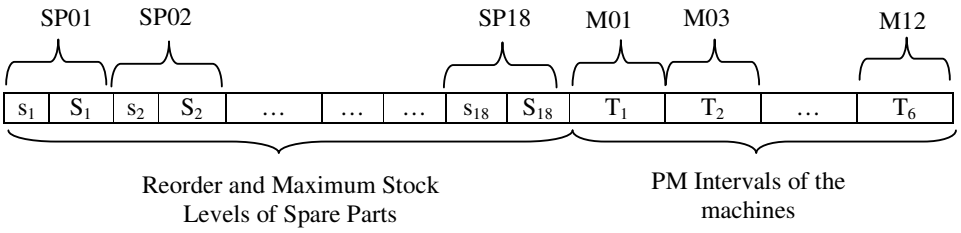

Figure 5.2 Structure of a chromosome

5.2.1.2 Genetic Operators

5.2.1.2.1 Selection. The proposed GA performs tournament selection. In tournament selection, the tournament size has been taken as two. In other words, two individuals are chosen at random from the population. The fittest of two individuals is selected to be a parent. The other is returned to the population and can be selected again.

Associated with the selection step is the "elitism" strategy, where the best two chromosomes (as determined from their fitness evaluations) are placed directly into the next generation. This guarantees the preservation of the best chromosomes at each generation. Note that the two elitist chromosomes in the original population are also eligible for selection and subsequent recombination.

5.2.1.2.2 Crossover. For each pair of selected parents, the crossover operation is applied to generate a new pair of offspring. The proposed GA performs two-point crossover (Cheu et al., 2004; Ding et al., 2003), and the crossover points are selected randomly. Two parent chromosomes between these points are then interchanged to produce two new offsprings. The process of crossover operation is demonstrated in Figure 5.3.

5.2.1.2.3 Mutation. Since real-valued encoding is used in the GA, the mutation operator, which is applied to each gene, is implemented by random replacement (Marzouk & Moselhi, 2002). So, if a gene is to be mutated, a new inventory level or PM interval is randomly picked and assigned to the gene (see Figure 5.4).

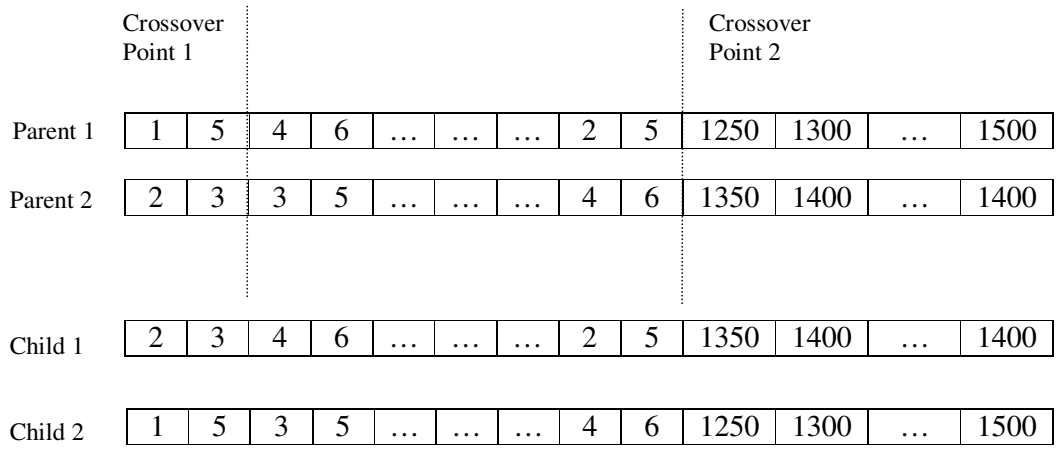

Figure 5.3 Two-point cross-over

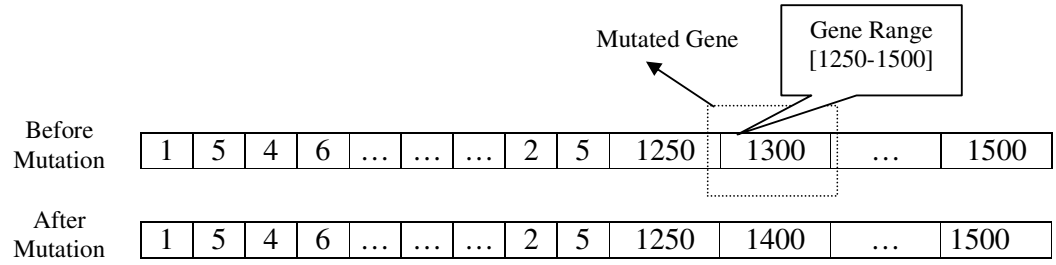

Figure 5.4 Mutation

5.2.1.3 Fitness Evaluation

The primary objective in this study is to develop a procedure for joint optimization of spare parts inventory and maintenance policies. The fitness of the solutions generated during the search process is evaluated using the total annual cost function given in section 5.2. This cost function is embedded to the simulation model of the motor block line, which represents the spare parts inventory management and maintenance policies in detail. The control logic, validation and verification of the simulation model is given in the following sections.

5.2.1.3.1 The Control Logic of Simulation Model. The simulation model was developed in a modular approach using Arena 3.0. The first module includes the operation of the motor block manufacturing line. The second module incorporates preventive maintenance, breakdown maintenance activities and the spare parts

demand arising from these activities. The issues related to inventory control, ordering and emergency ordering of the spare parts are included in the third module. Arena Input Analyzer was utilized to analyze spare part lead time and maintenance data which were gathered from the Purchasing and Maintenance departments of the firm, respectively. Based on this analysis Mean Time Between Failures (MTBF), Mean Time to Repair (MTTR), and PM durations are all found to follow Weibull distribution and order lead times are found to follow triangular distribution. Hence, in modeling all stochastic input data, we referred to the results of this analysis. Besides, we assumed that:

- A PM action is performed when the machine is free of the product. So, there are no interruptions due to PM.
- Failures are detected instantaneously and the actions of maintenance are carried out in a perfect way, i.e. the machine becomes as good as new.
- Once a maintenance action begins on a machine, it is completed if all the required spare parts are present.
- There are enough maintenance personnel to carry out the required maintenance activities.

As mentioned earlier, the proposed HGA approach aims at finding optimal levels of reorder and maximum inventory for eighteen spare parts and also PM intervals for six critical machines. Before applying the method suggested, we carried out a number of simulation runs to identify an operability region for each decision variable, i.e. reorder and maximum inventory levels of spare parts and PM intervals of the machines. The figures 5.5 and 5.6 show the estimates of the total annual cost function under different values of maximum inventory level and reorder level kept for spare part 1, respectively. Due to the limited space, here, we presented the results of these simulation runs for only spare part 1. The region of experiment determined for other spare parts is summarized in Table 5.4. It must be noted that, for each machine, the PM Interval range was accepted to change from 1250 to 1500 hours.

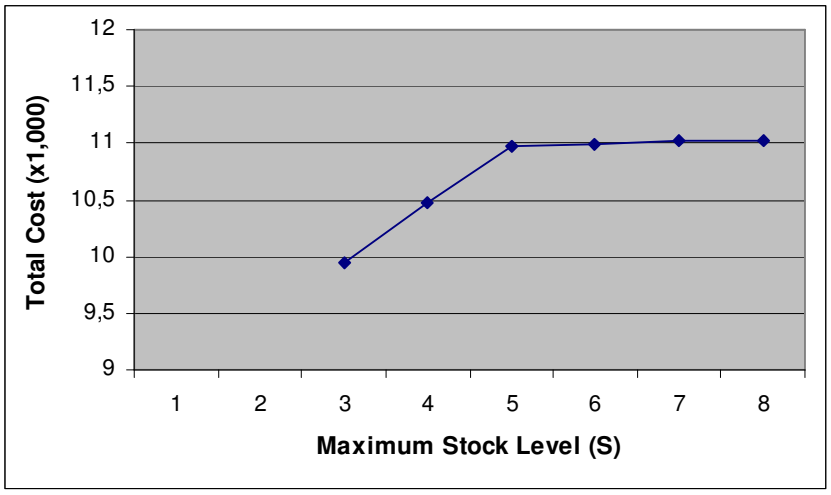

Figure 5.5 Total annual cost under the different levels of the maximum stock level for spare part 1

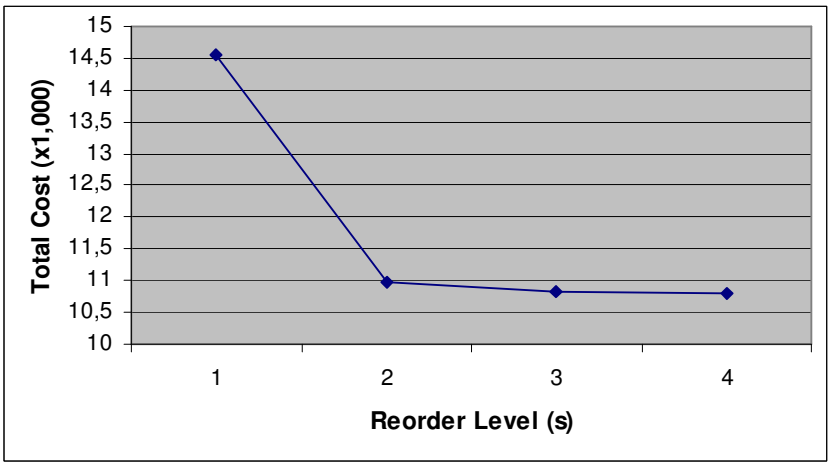

Figure 5.6 Total annual cost under the different levels of the reorder level for spare part 1

Based on these ranges, the whole search space has a volume of $4.46*10^{30}$ solutions. The search for a globally optimum solution in such a large search space is very difficult. This necessitates the use of search heuristics such as genetic algorithms since traditional, local search methods require a large computational time to search for quality solutions.

Table 5.4 Ranges for reorder, maximum stock levels of critical spare parts

Spare Part Code	s	S	Spare Part Code	s	S	Spare Part Code	s	S
SP01	1-2	3-5	SP07	1-2	2-4	SP13	1-2	3-5
SP02	2-4	5-8	SP08	1-2	3-5	SP14	1-3	5-8
SP03	1-3	5-8	SP09	1-2	3-6	SP15	1-2	3-5
SP04	1-2	3-5	SP10	1-2	3-5	SP16	1-3	4-6
SP05	1-3	4-6	SP11	1-3	4-8	SP17	1-2	3-4
SP06	2-4	5-8	SP12	1-3	5-7	SP18	1-3	4-6

We developed the simulation model in detail to realistically reflect the issues arising in case of BM or PM. The need for a BM arises due to a machine breakdown whereas PM is carried out in pre-specified time intervals. Following the record of a need for a maintenance activity of any type, it is checked whether the maintenance activity requires operating unit replacement (i.e., based on the probabilities given in Table 5.5). The flow chart describing the control logic of the simulation model can be seen in Figure 5.7.

Table 5.5 Replacement probabilities

Machine Code	Maintenance Type	Probability of Operating Unit Replacement
M01	BM	0.60
	PM	0.52
M03	BM	0.73
	PM	0.59
M07	BM	0.60
	PM	0.65
M08	BM	0.50
	PM	0.50
M09	BM	0.53
	PM	0.52
M12	BM	0.49
	PM	0.62

If the maintenance activity requires the replacement, the types of the spare parts required are determined by the simulation model based on the probabilities given in Table 5.6. As seen in this table, for each machine, one or more replacement types have been defined. Each replacement type of a machine requires the use of a

different spare part or a combination of spare parts. The demand for a spare part under each replacement type is determined using the probabilities given in Table 5.7.

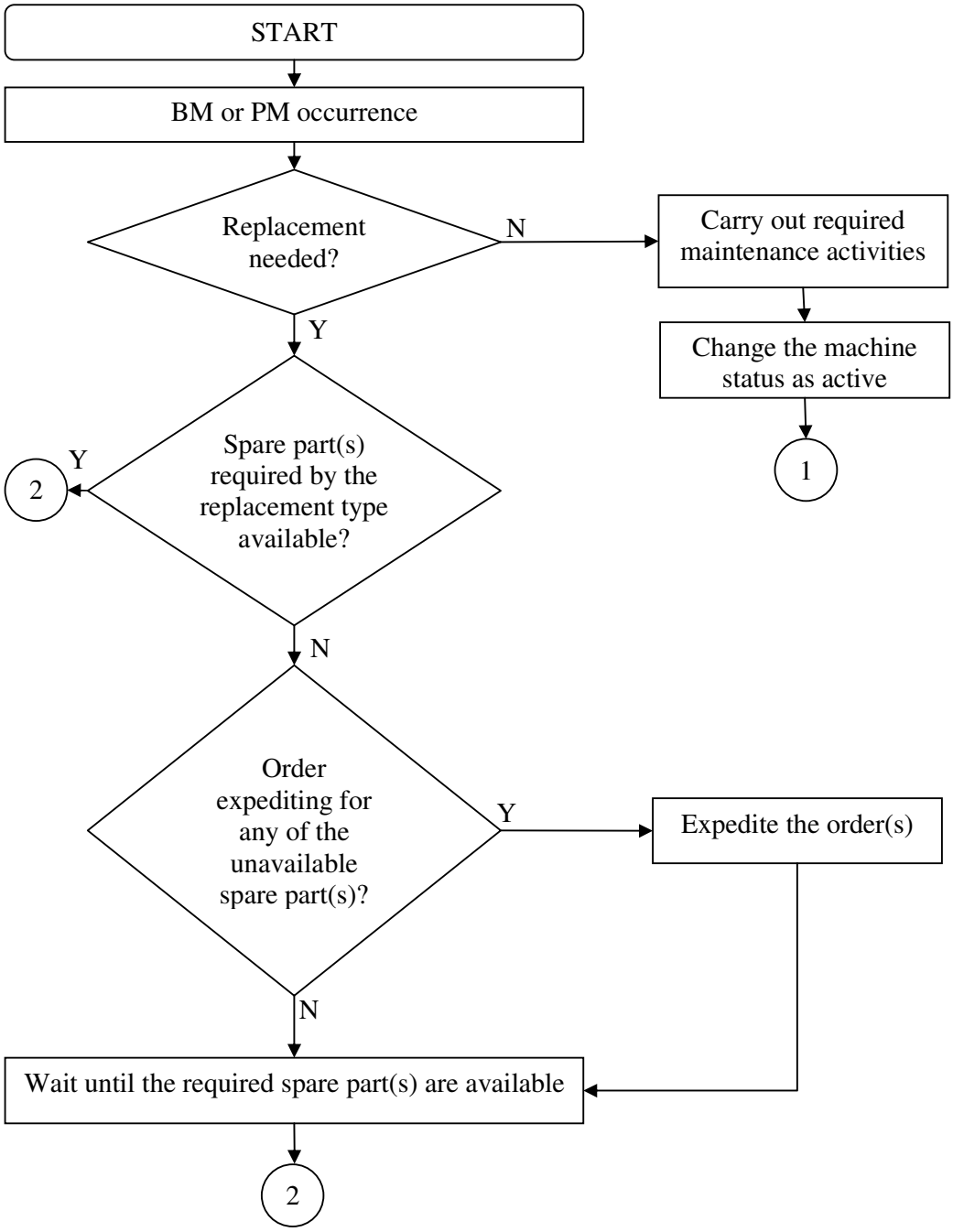

Figure 5.7 The control logic of the simulation model

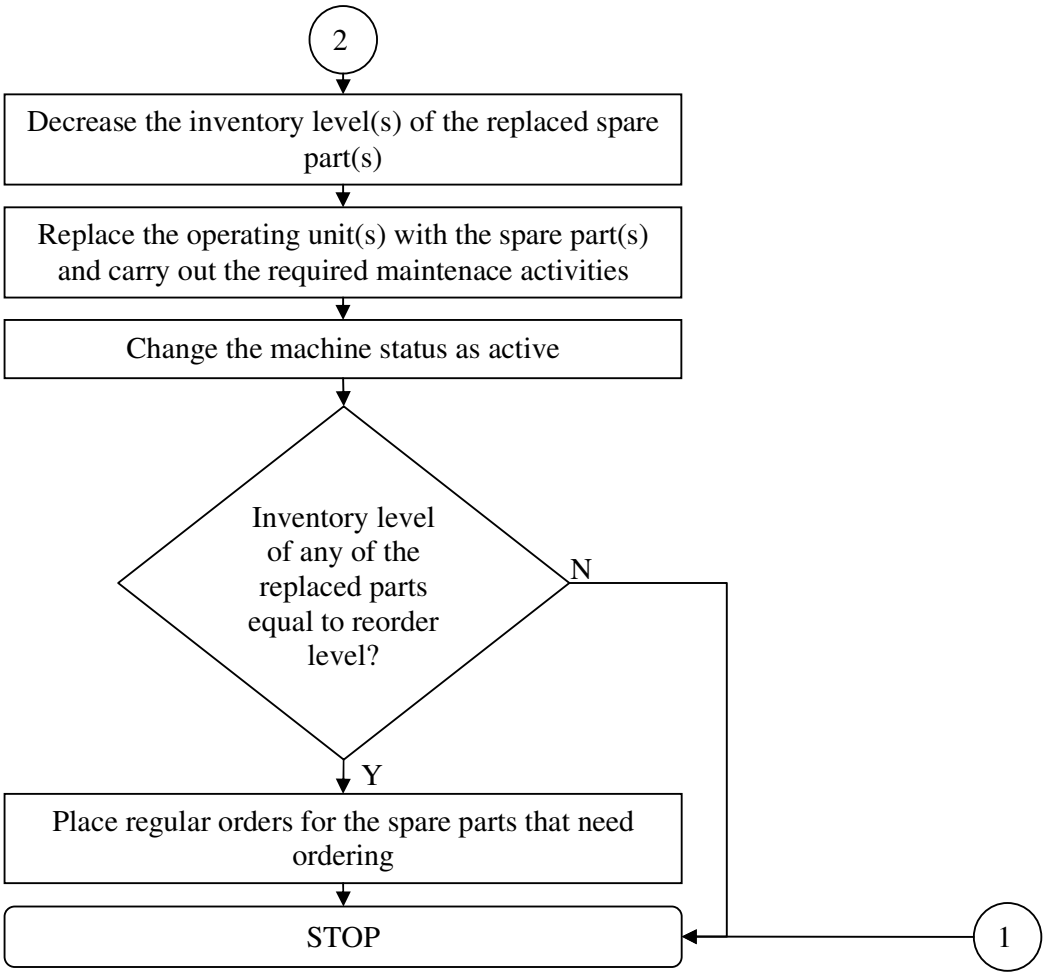

Figure 5.7 The control logic of the simulation model (from previous page)

If there is sufficient amount of spare part(s), the maintenance activity is carried out by using these spare parts. Otherwise it is checked whether the order for spare part(s) should be expedited depending on the urgency of the production needs. The order expediting probabilities of the spare parts, which have been determined based on the company records are given in Table 5.8.

When the inventory levels of all required spare parts are sufficient, the maintenance activity is carried out and the state of the machine is changed from “inactive” to “active”. The costs related to unit holding, regular and expedited ordering for each spare part are given in Table 5.9.

The simulation model is of non-terminating type and has to be warmed up to a steady state before experimenting with each of the input data sets. Warm-up period has been found as 75,000 minutes and each simulation experiment has been carried out for 432,000 minutes, the equivalent of 300 working days with three eight hour shifts.

Table 5.6 Replacement types

Machine Code	Replacement Type	Spare Part Code	Probability of Occurrence	
			BM	PM
M01	1	SP01	0.46	0.36
	2	SP02	0.27	0.28
	3	SP02 & SP03	0.27	0.36
M03	1	SP02	0.25	0.31
	2	SP04	0.13	0.23
	3	SP05	0.16	0.23
	4	SP06	0.21	0.15
	5	SP02 & SP03	0.25	0.08
M07	1	SP05	0.11	0.13
	2	SP06	0.14	0.13
	3	SP07	0.11	0.20
	4	SP09	0.14	0.20
	5	SP10	0.14	0.20
	6	SP11	0.18	0.07
	7	SP08 & SP12	0.18	0.07
M08	1	SP08 & SP13	1	1
M09	1	SP14	0.37	0.50
	2	SP15 & SP16	0.63	0.50
M12	1	SP17	0.32	0.38
	2	SP18	0.68	0.62

5.2.1.3.2 Validation and Verification of the Model. Validation of the simulation model aims at assuring that the model behaviour represents the real-world manufacturing system simulated. Through validation it is possible to determine, for example, whether the simplifications in the modelling process have caused unacceptably large errors in the results.

Table 5.7 Demand for spare parts

Machine Code	Replacement Type	Spare Part Code	Demand Quantity	Probabilities	
				BM	PM
M01	1	SP01	1	0.60	0.25
			2	0.40	0.75
	2	SP02	2	0.44	0.33
			3	0.56	0.67
	3	SP02	1	1	1
		SP03	1	1	1
M03	1	SP02	1	0.80	1
			2	0.20	----
	2	SP04	1	0.50	0.67
			2	0.50	0.33
	3	SP05	1	1	1
	4	SP06	1	0.25	1
			2	0.75	----
	5	SP02	1	1	1
		SP03	1	1	1
M07	1	SP05	1	0.75	0.50
			2	0.25	0.50
	2	SP06	1	1	1
	3	SP07	2	0.50	1
			3	0.50	----
	4	SP09	1	0.33	0.33
			2	0.67	0.67
	5	SP10	1	0.50	1
			2	0.50	----
	6	SP11	1	0.75	1
			2	0.25	----
	7	SP08	1	1	1
		SP12	1	1	1
M08	1	SP08	1	1	1
		SP13	1	1	1
M09	1	SP14	1	----	0.75
			2	0.50	0.25
			3	0.50	----
	2	SP15	1	1	1
		SP16	1	1	1
M12	1	SP17	1	0.33	0.40
			2	0.67	0.60
	2	SP18	1	0.62	0.50
			2	0.38	0.50

Table 5.8 Order expediting probabilities of spare parts

Spare Part	Order Expediting Probability	Spare Part	Order Expediting Probability	Spare Part	Order Expediting Probability
SP01	0.40	SP07	0.20	SP13	0.25
SP02	0.44	SP08	0.25	SP14	0.40
SP03	0.29	SP09	0.33	SP15	0.33
SP04	0.20	SP10	0.20	SP16	0.25
SP05	0.25	SP11	0.25	SP17	0.40
SP06	0.25	SP12	0.33	SP18	0.14

Table 5.9 Spare part properties

Spare Part Code	Unit Holding Cost ($/year)	Regular Order Cost ($/Order)	Expedited Order Cost ($/Order)
SP01	296	27	75
SP02	169	42	68
SP03	230	84	136
SP04	375	68	112
SP05	316	45	85
SP06	337	32	78
SP07	357	24	96
SP08	263	50	110
SP09	461	105	214
SP10	405	25	105
SP11	338	76	146
SP12	370	23	46
SP13	303	46	92
SP14	330	70	150
SP15	434	52	113
SP16	196	35	55
SP17	266	15	38
SP18	433	45	92

The model structure is validated together with a group of manufacturing and maintenance staff of the firm. The model is also validated quantitatively by comparing the simulation results with the record of the firm. As validation measure, we selected monthly throughput.

The actual monthly throughput and the simulation results are shown in Table 5.10. The mean simulated monthly throughput is 346 with a standard deviation of 23.36. The actual mean monthly throughput is 351 with a standard deviation of 22.35. With

these values, the test statistics (t_0) is computed as –0.67. Since $|t_0| = 0.67 < t_{0.025,9} = 2.26$, we can state that the simulation model is representative of the actual manufacturing system.

The simulation model should also be verified to ensure the accuracy of simulation code. In order to verify the simulation model, the Trace property of the ARENA simulation software has been used. The production of a motor block and the execution of a maintenance activity have been traced.

Table 5.10 Actual monthly throughput and the simulation model's estimates

Replication	Model's Estimates	Actual Monthly Throughput
1	380	340
2	342	325
3	361	325
4	380	380
5	323	362
6	323	380
7	323	325
8	323	343
9	361	370
10	342	362
Average	**346**	**351**

5.2.1.4 Analysis of the Effect of GA Parameters

The parameters of a GA namely population size, number of generations, crossover and mutation probabilities significantly affect the convergence speed of the GA and the accuracy of the optimum solution. In order to determine the most efficient GA parameters that minimize the total cost function given earlier, a set of experiments was performed in this study. The levels of the two quantitative input factors and the population size/generation number combination are given in Table 5.11.

Firstly, the performance of the algorithm was searched for under three different levels of total number of chromosomes generated. As expected, the best results are obtained when the total number of chromosomes generated was large, 1200 (see Figure 5.8). Next, by fixing the total number of chromosomes at 1200 (i.e., three

different combinations of population size and the number of generations 20/60, 30/40 and 60/20 resulted in 1200 chromosomes) the probability of crossover and mutation has been varied in three levels. As seen in Table 5.11, two indicator variables have been used to denote the three levels of population size/generation number combination in the regression model. Finally, to carry out the experiments, a full factorial design with ten replications has been employed.

Table 5.11 Experimental factors

Quantitative input factors	**Coded**	**Levels**		
		1	**2**	**3**
Crossover probability (%C)	x_1	0.30	0.60	0.90
Mutation probability (%M)	x_2	0.02	0.06	0.10
Population/generation combination (P/G)	z_1	z_2		
20/60	1	0		
30/40	0	1		
60/20	0	0		

Figure 5.9 provides the scatter plot that shows the value of objective function under each run. Based on this scatter plot, we could state that the runs that use a population size of 20 with 60 generations achieve the lowest total cost with the smallest spread.

Table 5.12 summarizes the results of the regression analysis. The factors with a p value that is smaller than 0.05 are statistically significant with a 95% level of confidence. As seen in Table 5.12, except for probability of crossover, all input factors are significant. This regression model suggests that the total cost can be minimized with small values of mutation probability and a population size of 20 with 60 generations.

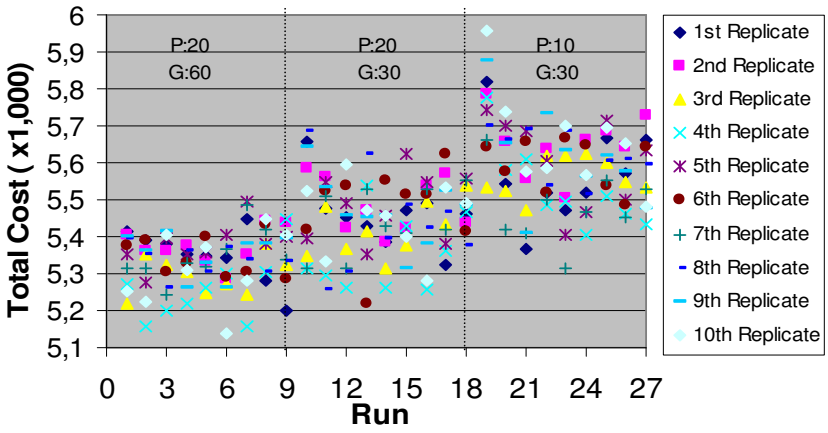

Figure 5.8 Scatter Plot of responses from 10 replications when the number of chromosomes generated is varied

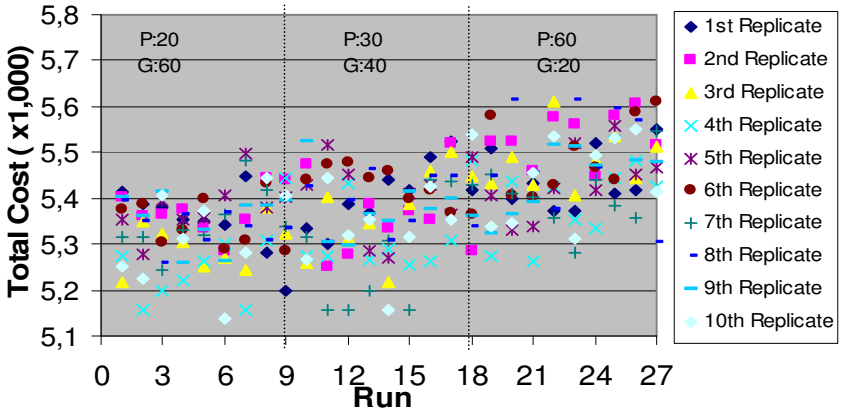

Figure 5.9 Scatter plot of responses from 10 replications when the number of chromosomes generated is fixed

Table 5.12 Regression analysis

Predictor	Coefficient	Standard Deviation	p	T
Constant	5432	18.17	0.000	299.01
x_1	-14.06	21.08	0.506	-0.67
x_2	579.4	158.1	0.000	3.66
z_1	-114.57	12.65	0.000	-9.06
z_2	-88.79	12.65	0.000	-7.02

5.2.2 Hybridizing GA with Simulated Annealing

Using the optimal GA parameters suggested in earlier section, next, we hybridized GA with SA. This hybrid algorithm employs the probabilistic acceptance criterion of SA for selecting new solutions, which permits some control over the acceptance of newly created solutions.

In this section, firstly, the working principle of the SA algorithm is explained. Following, the results of experiments to determine the best value of cooling parameter are presented.

5.2.2.1 Structure of the SA Algorithm

The steps of the SA algorithm as applied at the k^{th} iteration of the proposed algorithm are described as follows:

Step (1): Calculate the new temperature $T_k=T_0\ (\alpha)^k$, where α, T_k, T_0 are the cooling parameter, the temperature at the k^{th} iteration, and the initial temperature respectively.

Step (2): Calculate $\Delta E = E_j - E_i$ for the candidate solution proposed by GA, where E_i, E_j are the total annual cost values for the SA current solution and the candidate solution.

Step (3): If $\Delta E<0$, then accept the candidate solution and replace it with the worst solution of population pool. Let the candidate solution be the current solution of SA and go to step (4). Otherwise: If $\exp [-\Delta E/T_k] \geq U\ (0,1)$ then accept the candidate solution and replace it with the worst solution of population pool. Let the candidate solution be the current solution of SA and go to step (4). Else reject the candidate solution and go to Step (4).

Step (4): If all candidate solutions are tested, stop SA test process, otherwise go to Step (2).

The steps of the SA algorithm are presented as a flowchart in Figure 5.10.

As indicated in step 1, an exponential cooling schedule ($T_k=T_0\ (\alpha)^k$) has been used to decrease the temperature. In order to implement this schedule, the values of initial temperature and cooling parameter (α) must be determined at the beginning of the search process.

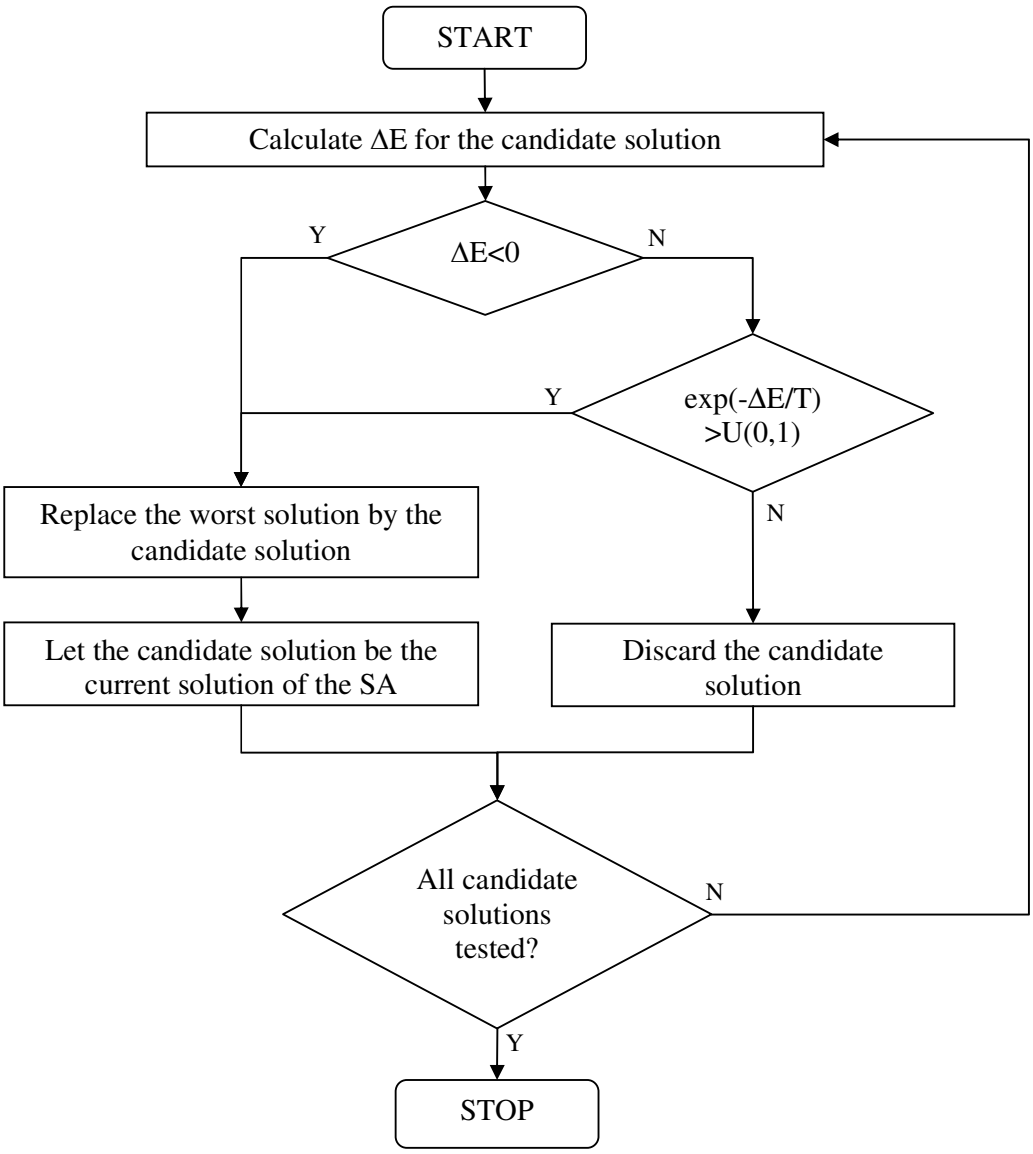

Figure 5.10 Flow chart of the SA algorithm

In this study, the value of the initial temperature has been set to a very large value (15000). In order to determine the best value of cooling parameter (α), a number of experiments has been performed and the results obtained are summarized in Table 5.13. Three values of α covering a relatively wide range have been selected. With α = 0.85, the decrease in temperature is very rapid and the algorithm lacks in exploration, concentrating more on exploitation in the neighborhood of a solution in the population pool. With α = 0.95, the temperature does not drop sufficiently and the method works as a simple GA technique. The cooling schedule with α = 0.90 provides a good compromise between the exploration and exploitation during the search process. Hence, in the following experiments, the value of cooling parameter is set to 0.90.

Table 5.13 Average performance of the HGA method with different values of the cooling parameter

α	**Average of Minimum Total Cost Values**
0.85	5338
0.90	5312
0.95	5389

5.2.3 Experimental Results

In this section, the results of experiments to evaluate the performance of pure GA and hybrid GA are presented.

5.2.3.1 Genetic Algorithm

Based on the results of the regression analysis discussed in section 5.2.1.4, this section presents the results of a GA optimization using a population size of 20, and 60 generations of evolution. The probability of crossover operation is set to 0.60. Mutation is performed immediately after the crossover with a probability of 0.02. To balance the disruptive nature of the chosen crossover and mutation, the elitism strategy is used with two elite chromosomes to preserve the best individuals.

The optimization process takes approximately one hour and the best solution is obtained after evaluating no more than 1200 alternatives. So, the ratio of search space investigated is very small when compared to the number of solution alternatives given in section 5.2.1.3.1. This shows the efficiency of the GA approach in accurately examining only a limited portion of the search space.

The convergence graph of the GA is presented in Figure 5.11. As shown in this figure, after 37 iterations the algorithm arrives at a solution that reduces the total cost from the initial value of 5690 to 5156 (a reduction of 9%). This recommended solution remains unchanged during the next 23 generations. Table 5.14 presents the optimal values suggested for reorder points and maximum inventory levels to keep for spare parts. Table 5.15 presents the optimum values of PM intervals for six critical machines. Based on the company records, the total annual maintenance cost and average monthly throughput are $10,968 and 351 motor blocks, respectively. The optimum solution suggested by GA resulted in $5,156 total annual maintenance cost and average monthly production of 373 motor blocks (see Table 5.16). These results imply 53% reduction in total annual maintenance cost and 6% improvement in average monthly production.

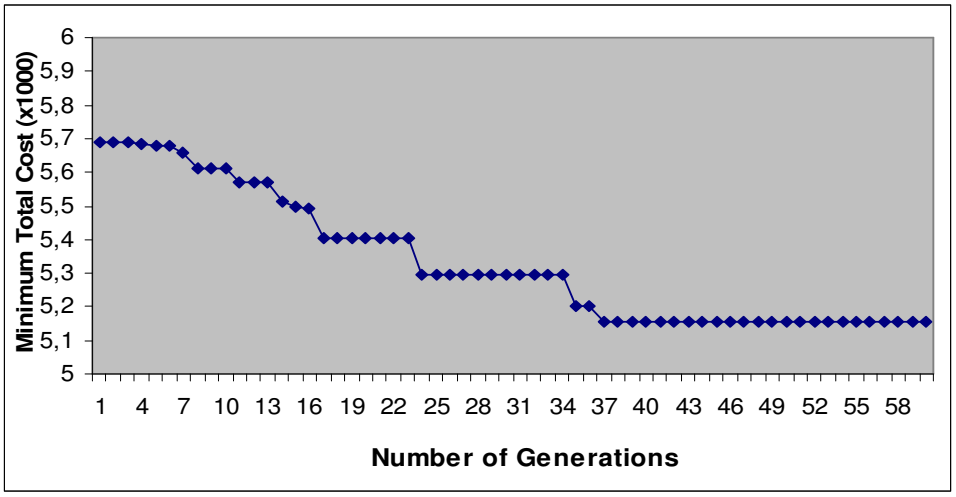

Figure 5.11 Convergence graph of GA

Table 5.14 Reorder, maximum stock levels of spare parts for the optimum solution

Spare Part Code	s	S	Spare Part Code	s	S	Spare Part Code	s	S
SP01	1	3	SP07	1	2	SP13	1	3
SP02	3	5	SP08	1	3	SP14	1	5
SP03	2	5	SP09	1	3	SP15	1	3
SP04	1	4	SP10	1	3	SP16	1	4
SP05	1	4	SP11	1	4	SP17	1	3
SP06	2	5	SP12	2	5	SP18	1	4

Table 5.15 PM intervals of the machines for the optimum solution

Machines	**M01**	**M03**	**M07**	**M08**	**M09**	**M12**
PM Intervals	1291	1348	1276	1304	1361	1406

Table 5.16 Actual monthly throughput and simulation model's estimates for the optimum solution

Replication	Model's Estimates	Actual Monthly Throughput
1	378	340
2	380	325
3	360	325
4	360	380
5	376	362
6	380	380
7	378	325
8	360	343
9	378	370
10	378	362
Average	**373**	**351**

5.2.3.2 Hybrid Genetic Algorithm

The features for the GA adapted in the hybrid approach have been borrowed from the results described in the previous section. In summary these are steady state approach, tournament selection, two-point crossover, random mutation, and a population size of 20.

The initial temperature is fixed to a large value (15000) and is reduced gradually according to an exponential cooling schedule (α=0.90). In the reported experimentation, the genetic operations have been performed 20 times for each

temperature. The number of temperature alterations was fixed to 60, giving 1200 fitness evaluations per run of the algorithm.

The convergence graph of HGA is presented in Figure 5.12. As shown in this figure, after 26 iterations the algorithm arrives at a solution that reduces the total cost from the initial value of 5690 to 5156 (a reduction of 9 %). This recommended solution remains unchanged for the next 34 generations. We noted that both HGA and pure GA suggest the same values for the decision variables studied.

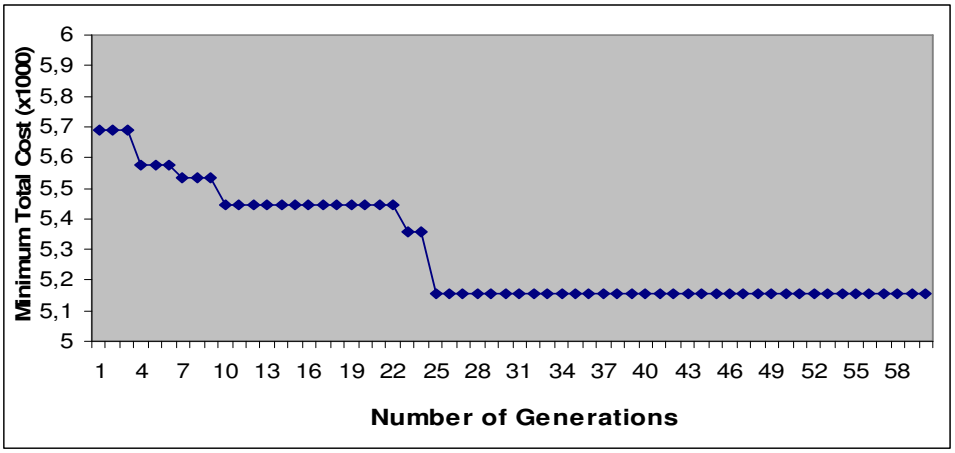

Figure 5.12 Convergence graph of HGA

5.2.3.3 Comparing GA and HGA

The pure GA and HGA have been run for ten different random number seeds. At the end of each experiment, the minimum total cost value has been noted for both algorithms.

The average of minimum total cost values obtained over ten experiments for GA is 5337.8 and the minimum of these total cost values over ten experiments is 5156 (see Table 5.17). Although the minimum total cost value suggested by the HGA is same as the minimum total cost value suggested by the GA, the average of minimum total cost values for the HGA was found to be lower than that of the GA.

Table 5.17 Comparison of results

	Average of Minimum Total Cost Values	Minimum of Minimum Total Cost Values
GA	5337.8	5156
GA/SA	5312.1	5156

The convergence of two algorithms is compared in Figure 5.13 and Figure 5.14. Figure 5.13 and Figure 5.14 show the mean total cost and the minimum total cost of population members at the end of each generation for GA and GA/SA, respectively. It is obvious from these figures that the proposed hybrid method (HGA) improves the convergence performance of the simple GA.

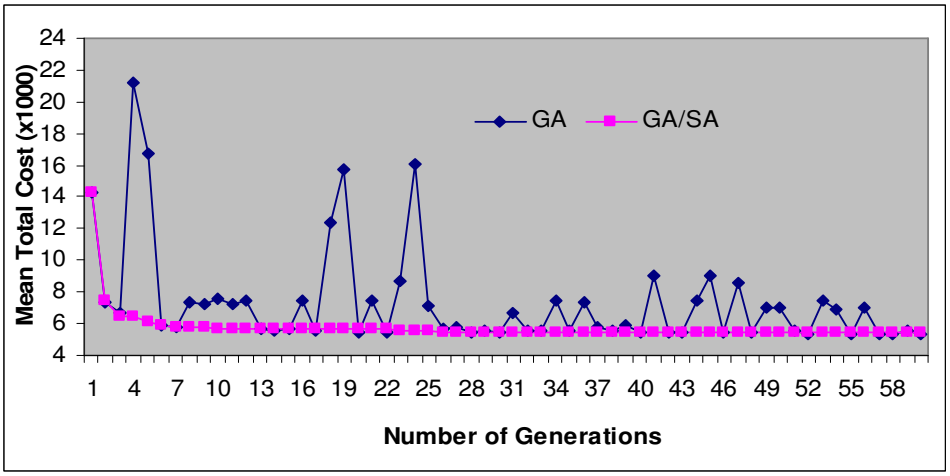

Figure 5.13 Mean total cost vs generation number: GA vs HGA

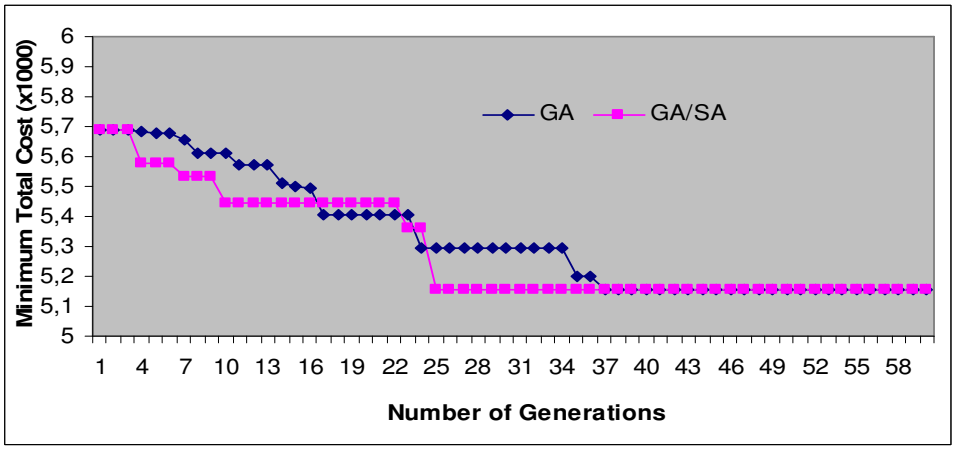

Figure 5.14 Minimum total cost vs generation number: GA vs HGA

CHAPTER SIX
CONCLUSION

The unavailability of spare parts at the time they are needed by the maintenance department is a major problem for many industrial organizations. The common approach to solve this problem is overstocking the spare parts at a substantial inventory carrying cost. However, a cost effective solution of this problem requires a trade-off between overstocking and shortages of spare parts. In order to deal with this trade-off, the problem should be solved by joint, rather than separate or sequential optimization of PM and spare parts inventory policies.

Joint optimization of maintenance and spare parts inventory policies usually leads to complex optimization problems where the objective function possesses no analytically trustable form and owns many local optima. In such cases the estimation of the values of the criterion function by simulating the corresponding system and the search for an optimal solution by GAs has been proved to be a powerful approach. However, GAs suffer from poor convergence properties. SA, on the other hand, has better convergence properties, but it cannot easily exploit parallelism. So, by hybridizing GAs and SA, the strengths of both algorithms can be retained

In this study, we have presented an approach that combines GAs and simulation for the joint optimization of spare part provisioning and PM policies of an automotive factory. A simulation model of the manufacturing system was developed, and a GA was integrated with this model to optimize the parameters of the simulation model. A set of designed experiments was carried out to determine best combination of GA parameters. Moreover, in order to improve the convergence properties of the GA, a hybrid GA has been proposed using the probabilistic acceptance rule of the SA within the GA framework.

The best solution proposed by the pure GA and Hybrid GA was compared with the current combination of control variables in terms of total annual cost and average

monthly production. It was found that total annual cost could be reduced by about 53% while achieving a larger amount of throughput.

One extension of this study could be to integrate this simulation-based joint optimization procedure into a Decision Support System framework. So that both the input data entrance step and also the link among three methodologies, i.e., simulation, GA and SA can be automated. In doing so, both the use of this hybrid approach in an industrial environment will be eased and also the system will always work with up-to-date data.

Although the proposed approach is good at finding the best combination of decision variables, the optimization process takes long time. This is mainly due to the fact that each evaluation of the GA objective function requires the execution of a pre-defined number of simulation model runs. This obstacle may be overwhelmed by constructing a regression metamodel of the simulation model. This regression model can be used as the objective function of the GA. Another approach to improve the efficiency of the proposed hybrid algorithm could be to design a parallel GA and distribute the task of a basic GA to different processors.

Finally, it must be pointed out that, there is usually more than one objective (low costs, low WIP, high revenue) when attempting to optimize a maintenance management system. This necessitates a multi-objective approach. Hence another extension of this study could be to employ a multi-objective simulation based GA optimization procedure for the joint optimization of spare parts inventory and maintenance policies.

REFERENCES

Alkhamis, T.M., & Ahmed, M.A. (2004). Simulation-based Optimization Using Simulated Annealing with Confidence Interval. *Proceedings of the Winter Simulation Conference,* 514-519.

Andradottir, S. (1998). A review of simulation optimization techniques. *Proceedings of the Winter Simulation Conference,* 151-158.

April, J., Glover, F., Kelly, J.P., & Laguna, M. (2003). Practical introduction to simulation optimization. *Proceedings of the Winter Simulation Conference,* 71-78.

Azadivar, F., & Shu, V. (1998). Use of simulation in optimization of maintenance policies. *Proceedings of the Winter Simulation Conference*, 1061-1067.

Azadivar, F. (1999). Simulation optimization methodologies. *Proceedings of the Winter Simulation Conference*, 93-100.

Azadivar, F., & Tompkins, G. (1999). Simulation optimization with qualitative variables and structural model changes: a genetic algorithm approach. *European Journal of Operational Research, 113*, 169-182.

Azadivar, F., & Wang, J. (2000) Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research, 38* (17), 4369-4383

Azzaro-Pantel, C., Bernal-Haro, L., Baudet, P., Domenech, S., & Pibouleau, L. (1998). A two-stage methodology for short-term batch plant scheduling: discrete-event simulation and genetic algorithm. *Computers and Chemical Engineering*, *22* (10), 1461-1481.

Barretto, M. R. P, Brito, N. M. J, Chwif, L., & Moscato, L. A. (1998). Scheduling with the LEO Algorithm, *Technical Paper, Society of Manufacturing Engineers*, 10-17.

Barretto, M.R.P., Chwif, L., Eldabi, T., & Paul, R.J. (1999). Simulation optimization with the linear move and exchange move optimization algorithm. *Proceedings of Winter Simulation Conference,* 806-811.

Beasley, D., Bull, D.R., & Martin, R.R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, *15* (2), 58-69.

Bodenhofer, U. (October 21, 2003). *Genetic Algorithms: Theory & Applications.* Retrieved May 10, 2005, from http://www.flll.uni-linz.ac.at/teaching/Ga/GA-Notes.pdf

Brady, T., & McGarvey, B. (1998). Heuristic optimization using computer simulation: a study of staffing levels in a Pharmaceutical manufacturing laboratory. *Proceedings of Winter Simulation Conference,* 1423-1428.

Breskvar, U. & Klajic, M. (2003). Interactive scheduling with genetic algorithms and visual event simulation model. *Proceedings of the IEEE Conference on Computer as a Tool (EUROCON 2003), 1*, 429-432.

Busetti, F. (December 24, 2000). *Simulated annealing overview*. Retrieved January 25, 2006, from http://www.geocities.com/francorbusetti/saweb.pdf

Cave, A., Nahavandi, S. & Kouzani, A. (2002). Simulation optimization for process scheduling through simulated annealing. *Proceedings of Winter Simulation Conference,* 1909-1913.

Cheu, R.L., Wang, Y., & Fwa, T.F. (2004). Genetic algorithm-simulation methodology for pavement maintenance scheduling. *Computer-Aided Civil and Infrastructure Engineering*, *19*, 446-455.

Chien, W.T., Lin. C., & Sphicas, G. (1997). A systematic approach to determine the optimal maintenance policy for an automated manufacturing system. *Quality & Reliability Engineering International, 13*, 225-233

Coley, D.A. (2003). *An introduction to genetic algorithms for scientists and engineers*. Singapore: World Scientific.

Da Silva, A.P.A. (2002). Tutorial Genetic Algorithms. *Learning and Nonlinear Models*, *1* (1), 45-60.

Dilworth, J.B. (1992). *Operations management: Design, planning and control for manufacturing and services.* New York: McGraw-Hill.

Ding, H., Benyoucef, L., & Xie, X. (2003). A Simulation – Optimization approach using genetic search for supplier selection. *Proceedings of the Winter Simulation Conference,* 1260-1267.

Dümmler, M.A. (1999). Using simulation and genetic algorithms to improve cluster tool performance. *Proceedings of the Winter Simulation Conference,* 875-879.

Fu, M. (2002). Optimization for simulation: theory vs. practice. *INFORMS Journal On Computing. 14* (3), 192-215.

Fujimoto, H., Tanigawa, Y., Yasuda, K., & Iwahashi, K. (1995). Applications of genetic algorithm and simulation to dispatching rule-based FMS scheduling. *Proceedings of IEEE International Conference on Robotics and Automation*, 190-195.

Gaither, N. & Fraizer, G. (2002). *Operations Management*. USA: South-Western.

Gen, M., & Cheng, R. (1997). *Genetic algorithms & engineering design.* New York: John Wiley & Sons.

Gharbi, A., & Kene, J.P. (2000). Production and preventive maintenance rates control for a manufacturing system: An experimental design approach. *International Journal of Production Economics, 65*, 275-287.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, *8* (1), 156-166.

Glover, F., Kelly, J. P., & Laguna, M. (1995). Genetic algorithms and tabu search: Hybrids for optimization. Computers and Operations Research, 22 (1), 111-134.

Glover, F. & Laguna, M. (1997). Tabu Search. Norwell: Kluwer Academic Publishers.

Glover, F., Laguna, M., & Marti, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics, 29*(3), 653-684.

Glover, F., & Kochenberger, G.A. (2002). *Handbook of metaheuristics*. Boston: Kluwer Academic Publishers.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization and machine learning.* Massachusetts: Addison Wesley.

Grupe, F.H., & Jooste, S. (2004). Genetic algorithms: A business perspective. *Information Management & Computer Security, 12* (3), 289-298.

Haddock, J., & Mittenthal, J. (1992). Simulation optimization using simulated annealing. *Computers and Industrial Engineering, 22* (4), 387-395.

Haupt, R.L., & Haupt, S.E. (2004). *Practical genetic algorithms* (2nd ed.). New Jersey: John Wiley & Sons.

Holland, J.H. (1992). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

Honkanen, T. (2004). Modelling industrial maintenance systems and the effects of automatic condition monitoring. PhD Thesis, Helsinki University of Technology.

Johnson, P.D. (2002). *Principles of controlled maintenance*. GA: The Fairmont Press.

Jones, M.H., & White, K.P. (2004). Stochastic approximation with simulated annealing as an approach to global discrete-event simulation optimization. *Proceedings of the Winter Simulation Conference,* 500-507.

Kabir, Z.A.B.M., & Al-Olayan, S.A. (1996). A stocking policy for spare part provisioning under age based preventive replacement. *European Journal of Operational Research*, *90*, 171-181.

Kendall, G. (July 15, 2002). *Simulated Annealing*. Retrieved February 16, 2006, from www.cs.nott.ac.uk/~gxk/aim/notes/simulatedannealing.doc

Kennedy, W.J., Patterson, W. J., & Fredendall, L. D. (2002). An overview of recent literature on spare parts inventories. *International Journal of Production Economics, 76*, 201-215.

Köchel, P., & Nielander, U. (2002). Kanban optimization by simulation and evolution. *Production Planning and Control, 13* (8), 725-734.

Kumar, S. (January 7, 2005). *Spare parts management-an IT automation perspective*. Retrieved May 6, 2005, from www.infosys.com/industries/resources/white-papers/automating_the_spare_parts _management_function.pdf

Law, A.M., & Kelton, W.D. (1991). *Simulation Modelling and Analysis*, New York: McGraw-Hill.

Lee, S.G., Khoo, L.P., & Yin, X.F. (2000). Optimising an assembly line through simulation augmented by genetic algorithms. *The International Journal of Advanced Manufacturing Technology*, *16*, 220-228.

Lee, H., & Kim, S.S. (2001). Integration of process planning and scheduling using simulation based genetic algorithms. *The International Journal of Advanced Manufacturing Technology*, *18*, 586-590.

Magoulas, G.D., Eldabi, T., & Paul, R.J. (2002). Global search strategies for simulation optimization. *Proceedings of Winter Simulation Conference,* 1978-1985.

Marsequerra, M., Podofillini, L., & Zio, E. (2001). Use of genetic algorithms for the optimization of spare parts inventory. *Proceedings of European Safety and Reliability Conference*, Torino, Italy, 1523-1530.

Marsequerra, M., Zio, E., & Podofillini, L. (2002). Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation. *Reliability Engineering and System Safety*, *77*, 151-166.

Marsequerra, M., Zio, E., & Podofillini, L. (2005). Multiobjective spare part allocation by means of genetic algorithms and Monte Carlo simulation. *Reliability Engineering and System Safety, 87* (3), 325-335.

Marzouk, M., & Moselhi, O. (2002). Simulation optimization for earthmoving operations using genetic algorithms. *Construction Management and Economics, 20*, 535-543.

Mitchell, M. (1996). *An introduction to genetic algorithms.* Cambridge: MIT Press.

Mobley, R.K. (2002). *An introduction to predictive maintenance* (2nd Edition). MA: Butterworth-Heinemann.

Niebel, B.W. (1994). *Engineering maintenance management*. New York: Marcel Dekker, Inc.

Olafson, S., & Kim, J. (2002). Simulation optimization. *Proceedings of the Winter Simulation Conference*, 79-84.

Patton, J.D. (1983). *Preventive maintenance.* New York: Instrument Society of America.

Paul, J.R., & Chanev, S.T. (1998). Simulation optimization using a genetic algorithm. *Simulation Practice and Theory*, *6*, 601-611.

Peng, X. (1998). Preventive maintenance and opportunistic maintenance planning for transient multi-unit systems. PhD Thesis, University of Florida.

Pham, D.T., & Karaboga, D. (2000). *Intelligent optimization techniques.* London: Springer-Verlag.

Pierreval, H., & Tautou, L. (1997). Using evolutionary algorithms and simulation for the optimization of manufacturing systems. *IIE Transactions, 29*, 181-189.

Popa, R., Aiordachioaie, D., & Nicolau, V. (2002). *Proceedings of the 16th European Meeting on Cybernetics and Systems Research*, 536-541.

Rayward-Smith, V.J., Osman, I.H., Reeves, C.R., & Smith, G.D. (1996). *Modern heuristic search methods.* West Success: John Wiley & Sons.

Reeves, C. (1995). *Modern heuristic techniques for combinatorial problems.* London: McGraw-Hill.

Reeves, C.R., & Rowe, J.E. (2002). *Genetic algorithms - principles and perspectives: A guide to GA theory.* Dordrecht: Kluwer Academic Publishers.

Robert, P.T., & Shahabudeen, P. (2004). Genetic algorithms for cost-effective maintenance of a reactor-generator system. *The International Journal of Advanced Manufacturing Technology, 23*, 846-856.

Rossetti, M.D., & Clark, G. (1998) Evaluating a queuing approximation for the machine interference problem with two types of stoppages via simulation optimization. *Computers & Industrial Engineering, 34*(3), 655-668.

Rutishauser, U. (May 25, 2002), *The Genetic algorithm-an example of an evolutionary algorithm.* Retrieved April 23, 2005, from http://www.urut.ch/pdfs/ga.pdf

Sarker, R., & Haque, A. (2000). Optimization of maintenance and spare provisioning policy using simulation. *Applied Mathematical Modelling, 24*, 751-760.

Schneider, N.L., Narayanan, S., & Patel, C. (1999). Application of genetic algorithms to airbase logistics. *Proceedings of IEEE Midnight-Sun Workshop on Soft Computing Methods on Industrial Applications*, 122-127.

Shenoy, D., & Bhadury, B. (1999). *Maintenance resource management: Adapting materials requirement planning,* London: Taylor&Francis.

Sherbrooke, C.C. (1968). METRIC: a multi-echelon technique for recoverable item control. *Operation Research, 16*, 122-41.

Shroff, P., Watson, D.W., Flann, N.S., & Freund, R.F. (2002). *Genetic simulated annealing for scheduling data-dependent tasks in heterogeneous environments.* Retrieved October 10, 2005, from http://www.cs.ucsd.edu/users/berman/hcw. papers

Shum, Y.S., & Gong, D.C. (2005). The application of genetic algorithm in the development of a preventive maintenance analytical model. *International Journal of Advanced Manufacturing Technology*, DOI 10.1007/s00170-005-0314-4.

Smith, D.E. (1973). An empirical investigation of optimum-seeking in computer simulation situation. *Operations Research*, *21*(2), 475–497.

Spieckermann, S., Gutenschwager, K., Heinzel, H., & Voss, S. (2000). Simulation-based optimization in the automotive industry: a case study on body shop design. *Simulation, 75* (5), 276-286.

Srivinas, M., & Patnaik, L.M. (1994). Genetic algorithms: A survey. *Computer*, *27* (6), 17-26.

Tekin, E., & Sabuncuoğlu, I. (2004). Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions, 36*, 1067-1081.

Wang, Z.G., Wong, Y.S., & Rahman, M. (2005). Development of a paralel optimization method based on genetic simulated annealing algorithm. *Parallel Computing, 31* (2005), 839-857.

Westerkamp, T.A. (1998). Evaluating the maintenance process. *IIE Solutions, 30* (12), 22-27.

Wiremann, T. (October 10, 2004). *Maintenance Inventory and Purchasing.*Retrieved September 12, 2005, from http://www.reliabilityweb.com/art04/maintenance_inventory_purchasing.pdf

Yang, T., Kuo, Y., & Chang, I. (2004). Tabu-search simulation optimization approach for flow-shop scheduling with multiple processors-a case study. *International Journal of Production Research, 42*(19), 4015-4030.

APPENDICES

Table A1 Information on manufacturing operations

Operation (OP) Identification	Operation Description	Machine Identification	Standard Time (minutes)
OP01	Volume control	------	11.98
OP02	Milling of reference surfaces	M01	4.95
OP03	Raw and finish milling of carter surface, drilling of oil gallery hole	M01	35.19
OP04	Raw milling of gasket surface	M02	9.32
OP05	Semi-finish milling of gasket surface, drilling and tapping of cover coupling	M03	29.16
OP06	Raw milling of front and back surfaces	M04	7.49
OP07	Raw boring of cylinder bores	M05	21.34
OP08	Raw milling of cap surface	M06	8.19
OP09	Drilling and tapping of cap coupling holes, milling of cap settling surface	M03	32.71
OP10	Finish milling of front and back side surfaces	M04	7.16
OP11	Drilling of front and back side surface holes	M07	31.35
OP12	Milling, drilling and tapping of right and left side surfaces	M08	29.19
OP13	Boring of front and back eccentric bearing	M07	31.87
OP14	Drilling of right and left side surface angular holes	M08	31.36
OP15	Drilling of main bearing oil hole	M09	7.56
OP16	Finish milling of cap surface	M06	6.33
OP17	First washing	------	10.05
OP18	Raw boring of main bearing and eccentric bearing	M10	12.23
OP19	Finish boring of main bearing and eccentric bearing	M11	25.65
OP20	Honing of main bearing	M12	8.70
OP21	Drilling of front and back side pin holes	M13	26.80
OP22	Finish milling of gasket surface	M14	9.13
OP23	Finish boring of cylinder bores	M09	15.28
OP24	Raw and finish honing of cylinder bores	M12	32.35
OP25	Vibrating	M14	12.34
OP26	Second washing	------	10.05
OP27	Water and oil gallery pressure test	------	40.68
OP28	Final control	------	13

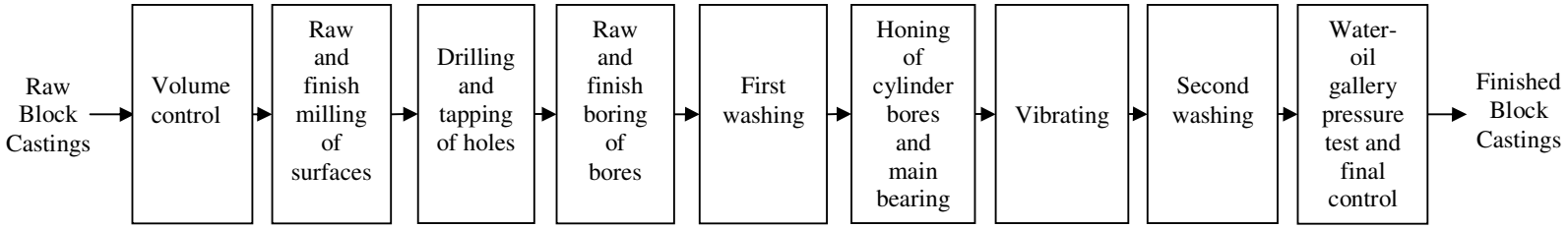

Figure A1 Flow of the manufacturing process