2008

ADAPTIVE, MULTI-OBJECTIVE JOB SHOP SCHEDULING USING GENETIC ALGORITHMS

Haritha Metta
University of Kentucky

ABSTRACT OF THESIS

ADAPTIVE, MULTI-OBJECTIVE JOB SHOP SCHEDULING USING GENETIC ALGORITHMS

This research proposes a method to solve the adaptive, multi-objective job shop scheduling problem. Adaptive scheduling is necessary to deal with internal and external disruptions faced in real life manufacturing environments. Minimizing the mean tardiness for jobs to effectively meet customer due date requirements and minimizing mean flow time to reduce the lead time jobs spend in the system are optimized simultaneously. An asexual reproduction genetic algorithm with multiple mutation strategies is developed to solve the multi-objective optimization problem. The model is tested for single day and multi-day adaptive scheduling. Results are compared with those available in the literature for standard problems and using priority dispatching rules. The findings indicate that the genetic algorithm model can find good solutions within short computational time.

KEYWORDS: Genetic Algorithms, Job shop scheduling, Multi-objective optimization, Adaptive scheduling, Asexual reproduction.

Haritha Metta
_____

02/28/2008
_____

ADAPTIVE, MULTI-OBJECTIVE JOB SHOP SCHEDULING USING GENETIC
ALGORITHMS

By

Haritha Metta

Dr. Fazleena Badurdeen

Director of Thesis

Dr. L.S. Stephens

Director of Graduate Studies

02/28/2008

RULES FOR THE USE OF THESIS

Unpublished thesis submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

<u>Name</u>   <u>Date</u>

_____

_____

_____

_____

_____

_____

THESIS

Haritha Metta

The Graduate School

University of Kentucky

2008

ADAPTIVE, MULTI-OBJECTIVE JOB SHOP SCHEDULING USING GENETIC
ALGORITHMS

<hr>

THESIS

<hr>

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in Mechanical Engineering
in the College of Engineering at the University of Kentucky

By

Haritha Metta

Lexington, Kentucky

Director: Dr. Fazleena Badurdeen

Assistant Professor of Mechanical Engineering

University of Kentucky Lexington, Kentucky

2008

Dedicated To My Parents

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

FIGURES

TABLES

LIST OF FILES

1. Haritha Metta Thesis: ~ 2 MB (File Size)

1 INTRODUCTION

1.1 Overview

Scheduling is broadly defined as the process of assigning a set of tasks to resources over a period of time (Pinedo, 2001). Effective scheduling plays a very important role in today's competitive manufacturing world. Performance criteria such as machine utilization, manufacturing lead times, inventory costs, meeting due dates, customer satisfaction, and quality of products are all dependent on how efficiently the jobs are scheduled in the system. Hence, it becomes increasingly important to develop effective scheduling approaches that help in achieving the desired objectives.

Several types of manufacturing shop configurations exist in real world. Based on the method of meeting customer's requirements they are classified as either open or closed shops. In an open shop the products are built to order where as in a closed shop the demand is met with existing inventory. Based on the complexity of the process, the shops are classified as single-stage, single-machine, parallel machine, multi-stage flow shop and multi-stage job shop. The single-stage shop configurations require only one operation to be performed on the machines. In multi-stage flow shops, several tasks are performed for each job and there exists a common route for every job. In multi-stage job shops, an option of selecting alternative resource sets and routes for the given jobs is provided. Hence the job shop allows flexibility in producing a variety of parts. The processing complexity increases as we move from single stage shops to job shops. Various methods have been developed to solve the different types of scheduling problems in these different shop configurations for the different objectives. These range from conventional methods such as mathematical programming & priority rules to meta-heuristic and artificial intelligence-based methods.

Job shop scheduling is one of the widely studied and most complex combinatorial optimization problems. A vast amount of research has been performed in this particular area to effectively schedule jobs for various objectives. A large number of small to medium companies still operate as job shops. Despite the extensive research carried out it

appeared that many such companies continue to experience difficulties with their specific job shop scheduling problems. Therefore developing effective scheduling methods that can provide good schedules with less computational time is still a requirement. Most of the real world manufacturing companies aim at successfully meeting the customer needs while improving the performance efficiency.

This objective is very challenging, particularly in a job shop, where the demand is highly unpredictable. Uncertainties at the planning and on on-line execution stages disturb the pre-planned schedule to a great extent. Deviations from pre-established schedules occur when the job shop experiences both external disturbances (e.g. urgent job arrivals) and internal disruptions (e.g. machine breakdowns) (Oats et al., 1999). One approach to deal with this schedule disruptions is to adjust the predictive schedule slightly without altering the entire schedule. However, this may not be effective in optimizing the objectives overall. On the other hand, changing the schedule to adapt to emerging situations can be difficult to perform manually. Therefore, in such situation it becomes necessary for companies to explore the scheduling of jobs adaptively, on a real-time basis effectively.

The motivation for this research comes from a real world job shop scheduling problem faced by a local-furniture manufacturing company. A variety of products such as office furniture, home and dining furniture including tables, chairs, desks, shelves, cabinets etc. are manufactured for individual customers in the plant. The jobs arrive on a daily basis and each has a specific processing time and due date. The aim is to schedule the incoming jobs efficiently on the machines to reduce the time they remain in the system and to meet the due dates. However events such as machine breakdowns, high priority job arrivals often interfere with scheduling to a great extent. Currently, the scheduling is performed by a simple observation analysis. The due dates of all the existing jobs (new and the old incomplete jobs) are first compared and based on observations the jobs are re-organized and assigned to machines manually. This method is very time consuming and could be extremely inefficient in optimizing the minimization of average flow time (which is the interest of the company) and meeting the due dates. This type of situation arises in most job shop plants.

In this research, we are developing a method to optimize the scheduling process that improves the customer satisfaction through better adherence to deliver dates and simultaneously minimizing the resource usage.

1.2   Scheduling Nomenclature

Prior to describing the problem statement some general scheduling nomenclature relevant to the job shop problem will be described here. Job shop scheduling problem can be defined as the processing of n jobs on m machines, each with a unique processing sequence on the machines. The processing time of a job j on a machine i released at release time R_j with a due date d_j is denoted by p_{ij}. Any scheduling problem can be denoted by a three field representation $\alpha \mid \beta \mid \gamma$ (Gen and Cheng, 1999). Here α represents the machining environment, β the processing characteristics of the job including the constraints and γ the objectives to be optimized. The relevant variables for the α, β and γ fields, for this research are briefly described below.

The machine environment in this research is a job shop and is denoted by j_m.

Job shops (j_m): Each job has its own routing sequence on machines. Any operation O_{ij} can be performed if and only if all the tasks preceding that particular operation are completed.

In this research, β represents the precedence constraints as described below:

Precedence constraints (*prec*): This term implies that one or more operations have to be performed before a job can start its processing on a particular machine. Different types of precedence constraints such as chains, intrees and outtrees exist in literature. In this research, precedence constraints exist between the operations for each job.

Preemption (*prmp*): This term is used to indicate that operation O_{ij} of job j can be interrupted anytime during its processing on machine i. This happens in a situation when a higher priority job must be processed. In this research no preemption is allowed.

Sequence dependent set-up times (s_{jk}): In most of real world applications if a machine has to process job k after finishing job j, a changeover time and cost must be incurred to prepare the machine for job k. These often vary depending on which job is processed after the other. In this research, the set-up times are assumed to be sequence-independent and as included in the processing time.

The γ field contains the objectives to be optimized in scheduling and those related to this research are:

Completion time (C_j): The time at which job j is completely processed.

Makespan ($C_{\max}$): The time taken to complete all the jobs in a system. Minimizing makespan improves the machine utilization.

Flow time ($F_j = C_j - R_j$): The flow time for job j is given by the time span between the completion time C_j and the ready time r_j. In this research we assume that all jobs are available at the start of processing. Hence r_j is zero. Therefore F_j is equal to C_j. Usually, a minimum flow time in a scheduling system minimizes the job lead times and reduces the inventory cost.

Mean flow time ($\sum_{j=1}^{n}(F_j / n)$): The average of flow time of all jobs in a system is defined as the mean flow time. One of the objectives in this research is minimization of mean flow time.

4

Tardiness ($T_j = \max\{C_j - d_j, 0\}$): Measure of tardiness indicates the system's efficiency in meeting the due dates set by the customer/company. It is defined as the maximum value between 0 and the difference of completion time and due date. If a job is completed before its due date, then a job is not tardy. Minimizing tardiness improves delivery performance and enhances customer satisfaction.

MeantTardiness ($\sum_{j=1}^{n} T_j / n$): The average tardiness of all individual jobs in the system is know as mean tardiness. The other objective in this research is minimizing the mean tardiness.

The two objectives mean flow time and mean tardiness optimized in this research do not converge i.e. they conflict with each other and the optimization involves a trade-off.

1.3 Problem Statement

Based on the above scheduling nomenclature, the research problem in this thesis is defined as a job shop problem with precedence constraints and optimizing mean flow time and mean tardiness can be denoted as ($J_m \mid prec \mid (\sum_{j=1}^{n}(C_j / n), \sum_{j=1}^{n}(T_j / n)))$).

The demand for high variety and low volume products poses many challenges in today's real time manufacturing systems. There is a growing need to improve the scheduling operations to meet shorter due dates and to best utilize the machines. Moreover, the scheduling approaches must improve the flexibility of scheduling operations. With increased flexibility disruptions could be handled with less impact on the system. Flexibility also helps in achieving the desired objective more effectively. The focus of the

$J_m \mid prec \mid (\sum_{j=1}^{n}(C_j / n), \sum_{j=1}^{n}(T_j / n))$ in this research considers the above issues.

However, looking at the furniture manufacturing problem there lies an additional challenge of having to adapt and modify the schedule on a daily basis as jobs enter the system. Therefore to efficiently schedule the operations, an adaptive system that modifies the schedule on a daily basis based on the current situation is required. This type of adaptive approach is very important to perform effective rescheduling of jobs. However, the jobs already set up/processed on any machines ideally should not be unloaded, to avoid re-set-up costs. Therefore, addressing all the above requires an adaptive, multi-objective optimization to minimize mean flow time and mean tardiness.

Job shop scheduling problem is NP-hard by nature. This complexity is further increased when additional constraints are added to solve the real world problem. The exact methods could solve only small size problems within acceptable time periods. Although they produce exact solution, they often simplify the instances. Meta-heuristics are semi-stochastic approaches that can produce near optimal solutions within less computational time. These approaches adapt to the problem situation. Among the meta-heuristic methods, genetic algorithms are techniques based on human evolution that were widely used to solve large optimization problems. The properties of genetic algorithms such as the use of a population of solutions, problem–independence enables them to be effectively used for job shop scheduling.

Asexual reproduction genetic algorithm involves the formation of offspring's from a single parent. These methods always produce feasible solutions and consume very less computational time as compared to sexual reproduction methods. Therefore these properties of asexual reproduction methods enabled them to be effectively used for the complex job shop problem involved in this research.

1.4 Research Objective

The objective of the research described here is,

- To determine best job schedule for the $(J_m \mid prec \mid (\sum_{j=1}^{n}(C_j / n), \sum_{j=1}^{n}(T_j / n)))$ problem using an asexual reproduction genetic algorithm.
- To efficiently solve the adaptive job shop scheduling problem above.

1.5 Organization of Thesis

This thesis is organized as follows: Chapter 2 gives a detailed overview of job shop scheduling literature. Chapter 3 presents the problem description and the methodology followed to develop the genetic algorithm-based adaptive job shop scheduling model for minimizing mean flow time and mean tardiness. Chapter 4 explains the problems tested and experimentation to evaluate the effectiveness of the GA model. Chapter 5 provides the results for the experimentations followed by the discussion of trends observed. Finally, chapter 6 provides the conclusions and future research directions.

2   LITERATURE REVIEW

This chapter discusses the state of art literature in areas related to the current research problem. Section 2.1 provides an overview of job shop scheduling problem and defines the performance measures related to this research. Section 2.2 provides a comprehensive review of literature dealing with various solution methodologies including Exact Procedures (Mathematical Modeling, Branch and Bound algorithms), Approximate Algorithms, Meta-heuristic Methods and Genetic Algorithms. Section 2.3 discusses literature related to multi-objective job shop scheduling and relevant works. A description of the working of multi-objective Genetic Algorithms is also provided. Sections 2.4 provide a brief review of reactive schedule literature related to job shops.

2.1   The Job Shop Scheduling Problem

Scheduling is defined as the allocation of shared resources to tasks over a given period of time (Pinedo, 2001). The general job shop scheduling problem can be described by a set of n jobs $\{J_i\}_{1 \leq i \leq n}$ which is to be processed on a set of m machines $\{M_j\}_{1 \leq j \leq m}$. The problem can be characterized as follows:

1. Each Job must be processed on each machine in an order given by a predefined sequence of operations
2. Each machine can process only one job at a time
3. Each job J_i is processed on machine M_j which is defined by the operation O_{ij}
4. Each operation O_{ij} requires an uninterrupted processing on machine M_j and preemption is not allowed
5. The processing times for each operation are known in advance
6. Two operations, $\{O_{ij}, O_{i,j+1}\}$ of the same job cannot be processed at the same time.

The job shop scheduling problem is one of the well-known and widely studied problems in the scheduling literature. Job shop scheduling is an NP hard problem (Sadeh, 1991), (Lorenco, 1995). The complexity characteristic and the close resemblance of the problem to the general domain of problems captured the interest of a significant number of researchers. Based on the optimizing criteria, the scheduling problem can be classified as the completion times related and due date related problem.

The nature of the scheduling environment plays a vital role in determining the job schedules. Typically, in a static environment the number of jobs and the arrival times are known in advance. If the arrival times of jobs are unknown the scheduling system is considered as dynamic. A dynamic scheduling system encounters the difficulties of randomness such as machine breakdowns, unexpected job orders etc. which are experienced in real world problems (Madureira et al., 2001). Most of the research during the last three decades has concentrated on the deterministic job shop problem making it one of the well developed models in the scheduling theory (Moon and Lee, 2000). In dynamic scheduling, the goal is not to find a single optimum but to continuously change the solution that adapts to the varying environment. Therefore these constraints increase the complexity and the computational time of these problems, even for a small problem (Vlazewicz et al., 2001).

Mean Flow Time and Mean Tardiness

The solution of any optimization problem is evaluated by an objective function. Objectives that are associated with cost, resources and time such as the makespan (the time at which the last job leaves the system), flow time, tardiness are minimized while the objectives related to production rate, machine utilization are maximized. Considerable amount of research is done in this area (Satake et al., 1999 ; Calavrese et al., 2001).

Most of the literature in the job shop problem considered makespan as the scheduling criteria (Moon and Lee, 2000). Among the completion time objectives, makespan is

frequently used in job shop scheduling problem as it is directly associated with machine utilization (Kubiak et al., 1996). Flow time minimization has recently gained a lot of importance in recent research work due to its effect on resources and inventory (Suresh and Mohanasundaram, 2005). Also know as lead-time, flow time is given by the length of time a job remains in its system. On the other hand, mean flow time is the average of the flow times of individual jobs. In scheduling, the mean flow time seems to be a more important objective than makespan, as makespan aims at minimizing the schedule duration, but in real time industry this duration is frequently defined by the time period of the process (Aldakhilallah and Ramesh, 2001).

In industrial production, jobs are released with subject to capacity constraints, material constrains, etc. Due date indicates a time when a job must be completed, to adhere to the projected delivery date. It is generally used to improve the customer service levels (Kuluc and Khraman, 2006; Cicirello and Smith, 2001). This is an important factor in the competitive world as customer satisfaction becomes one of the major priorities. Due date criterion can also be used in situations where priorities are assigned to complete a job on time with respect to a certain customer. In these situations, makespan is no longer economical in achieving the desired objective (Baykasoglu et al., 2008). Mean tardiness is a measure of the average of individual job tardiness. Minimizing tardiness is used as the objective when due dates are tight and to minimize late deliveries in a system (Gordon et al., 2002).

2.2   Solution Methodologies and Related Work

Over the past few decades several optimization techniques have been proposed for the job shop problem ranging from simple and fast dispatching rules to sophisticated branch-and-bound algorithms. However, with the rapid increase in the speed of computing and the growing need for efficiency in scheduling, it becomes increasingly important to explore ways of obtaining better schedules at some extra computational costs (Blazewicz et al., 1996).

2.2.1 Exact Procedures

Various solution techniques broadly classified under exact and approximate methods were developed for the Job shop problem (Dimopoulos and Zalzala, 2000). While considering the mathematical formulation methods, the mixed integer programming format developed by Manne (1960) was one of the most common forms of mathematical formulations. Later, Greenberg (Greenberg, 1968) developed a method based on Manne's integer programming formulation. However, the integer programming methods were practically infeasible (Giffler and Thompson, 1960) and computationally difficult (French, 1982). The reason is high problem simplification and numerous constraints associated with these problems. The success with mathematical programming was often possible with Lagrange Relaxation approaches (Manne, 1960; Fisher, 1973a; Fisher, 1973b) and decomposition methods (Chu et al., 1992; Kruger et al., 1995).

Research on exact solution techniques for job shop scheduling has also formed heavily on Branch and Bound techniques. Some of the early work in this area was performed by Brooks and White (1965) and Ignall and Schrage (Iima, 1999). The popular one machine decomposition problem was first successfully solved by McMahon and Florian (1975) which was considered as the best exact solution method for a long time. This was followed by several research papers such as Carlier and Pinson (1989), Brucker, et al. (1994), Boyd and Burlingame (1996). Although literature provides information about considerable improvements made by branch and bound algorithms, large size problems still can not be solved by these exact methods (Lourenco, 1995).

2.2.2 Approximate Procedures

The high computational time and the complexity involved with the exact procedures led several researchers' attention to the development of approximate methods for solving large sized problems (Franklin, 1969). Approximate procedures are adapted to large size problems to obtain near optimal solution within reasonable computational time. Glover and Greenberg (1989) in their paper discussed the importance and necessity of approximate procedures. The approximate procedures include the local search methods,

11

meta-heuristics, priority dispatching rules (pdr's), artificial intelligence and bottleneck - based heuristics. Literature shows a vast amount of work performed in this area on job shop scheduling.

Nasr and Elsayed (1990) presented two efficient heuristic to minimize the mean flow time in a general job shop type machining system with alternative machine tool routings. Their methods were based on decomposing a big problem into multiple sub-problems and solving them individually. They also developed a greedy procedure for the case of adding alternative machines by including a penalty cost. Kubiak et al. (1996) proved that there is an optimal job schedule with the shortest processing time (SPT) job order, for a reentrant job shop with one *hub* machine, where job enters a certain number of times. They derived a dynamic programming algorithm to find the optimal schedule under the bottleneck assumption and the hereditary order assumption. Their objective was to minimize total flow time.

Moon and Lee (2000) developed a heuristic to solve the job shop scheduling problem with alternate routings by dividing the problem into two problems; allocation and sequencing problem. They presented two different approaches to solve the two problems. The performance measures considered were mean flow time, makespan, maximum lateness and total absolute deviation from due dates. Aldakhilallah and Ramesh (2001) developed cyclic scheduling heuristics for the reentrant job shop scheduling environments. Their approach considered a repetitive production re-entrant job shop with a predetermined operations sequence on a particular single product. Their objective was to minimize both cycle time and flow time simultaneously.

Minimizing tardiness in job shop scheduling has been considered only in few papers. Vepsalainen and Morton (1987) studied and tested several dispatching rules, and presented the apparent tardiness cost (ATC) as the one that achieved the best results among them. Anderson and Nyirenda (1990) developed two rules combining two different dispatching rules. He, et al., (1993) developed an effective heuristic that minimizes the total tardiness of jobs. They used a heuristic exchange neighborhood of

asymptotic time complexity in this problem. Although this heuristic was effective the search process was time consuming. Pinedo and Singer (1999) presented a heuristic based on the shifting bottleneck procedure. This heuristic produced close to optimal solutions on 10 x 10 problems.

Some of the recent papers in this area include the works by Asano and Ohta (2002) that developed a heuristic-based algorithm having the tree search procedure to solve the minimum total weighted tardiness problem. Their work produced a sub-optimal solution within a shorter computational time. Bontridder (2005) proposed a neighborhood local search method to minimize the total weighed tardiness in a generalized job shop problem.

2.2.3 Meta-heuristic Methods

Meta-heuristics are semi-stochastic methods used for solving hard optimization problems (Kuluc and Kahraman, 2006). These techniques are the most recent developments in approximate methods to solve complex optimization problems. The neighborhood strategies developed in some of the works of Matsuo, et al. (1988) and Van Laarhooven, et al. (1987) form the basis for the formation of meta-heuristic methods (Blum and Roli, 2003). Meta-heuristic approaches unlike traditional approaches have the capability to adapt to the problem environment (Yamada, 2003). This provides an opportunity to apply these methods for real world problems. For complex real world problems, meta-heuristics are often applied with some other approach to enhance the problem solution. Meta-heuristics techniques can also be used to solve dynamic scheduling problems through combining with fuzzy sets theory techniques. While several methods are being addressed in literature, Tabu Search, Simulated Annealing (Laarhoven and Aarts, 1987) and Genetic algorithms (Davis, 1985) are the most successful meta-heuristic methods. The success of these methods is defined by their capability in producing good solutions (near optimal) in less computational time.

Tabu search (Baykasoglu et al., 2002) is a Meta-heuristic that guides the search direction to explore the solution space beyond local optimality. Glover (1986) derived the Tabu search heuristic. He defined Tabu search as a "as a meta-heuristic that is superimposed on

another heuristic". In this procedure, the Tabu search algorithm stores the previous search history (list of obtained solutions) in its memory. When the search process is carried out in a new neighborhood the algorithm tries to find the best solution by excluding earlier solutions stored in the memory. Therefore this procedure forbids (makes tabu) moves in new neighborhoods, by guiding the search process away from solutions that resemble previous ones.

Several heuristics based on Tabu search were developed for the job shop scheduling problem (Della et al., 1995). Laguna, et al. (1994) provided some of the earliest Tabu search approaches by creating three search strategies with simple move definitions. Following this work, literature provides several research papers and advancements in Tabu search heuristics. Sun and Batta (1996) proposed divide and a conquer scheme for the large scale job shop scheduling problems. Their approach was to decompose each job into cells and further applying iterative procedures to solve the individual cell scheduling problems. Armentano and Schrich (2000) presented a tabu search approach to minimize total tardiness in a job shop problem. Their method uses a set of dispatching rules to find the initial solution followed by the neighborhood search method based on critical paths of jobs. The advantage of this method is that it avoids local minima but a proper termination condition has to be set, which otherwise may end in the method not providing a good result (Peres and Paulli, 1997).

Simulated Annealing (SA) is a random search technique that originates from the analogy between annealing process and the search for the minimum in a general process. It was first developed by Kirkpatrick, et al. (1983) who adapted the work of Metropolis, et al. (1953) to constraint optimization problems. The SA algorithm starts with a randomly generated set of initial solutions and at a high starting temperature 'T'. The algorithm replaces the present solution with a solution from its neighborhood if that solution is "better" than the current one. A "better" solution in this algorithm could be the one whose objective function value is less / greater (minimization problem/maximization problem) than the current value, or some times even a value that is greater / lesser than the current one. The latter solutions are accepted with a probability. The probability is a function of

difference in objective function values and the temperature T. The value of temperature gradually decreases during the search process, thereby the solutions are replaced more number of times at the beginning and less replacements occur towards the end of the process. The above steps are repeated until a termination criterion is reached. This heuristic has a good ability to avoid the solutions being trapped in local minima because of the inclusion of the probability function. But the major shortcoming of this algorithm lies in the high computational cost for obtaining an exact solution (Wu and Wang, 1998). Research on literature provides papers on the modification of this algorithm to improve the solution accuracy and convergence.

Considerable research has been done on scheduling using SA. Ponnambalam et al. (2001) developed a SA approach for the job shop scheduling algorithm to minimize makespan. Sadeh (1996) developed a focused SA approach to solve the job shop scheduling problem to minimize tardiness and inventory costs. In this paper, a meta-heuristic procedure is developed that dynamically inflates the costs associated with the inefficiencies, there by improving the effectiveness of the SA procedure. Wu and Wang (1998) used the SA to obtain the minimum total tardiness. Their method could search for all of the feasible solutions and also has the capability of exploring cost non-decreasing configurations. Another work of Wu and Wang (1998) proposed a revised SA algorithm to minimize the total tardiness.

One among the meta-heuristic methods, Genetic Algorithms (GA's) is a well known evolutionary approach used to solve various optimization problems. In this research we use a GA-based approach to find best solutions to the job shop scheduling problem. Therefore brief descriptions of the GA operations relevant to the research are presented separately in the next section.

2.2.4 Genetic Algorithms

Inspired by principles of natural evolution, GA's are one of the well known types of evolutionary algorithms (Holland, 1985). GA's are iterative procedures where each iteration is termed as generation. The process begins with an initial population (N_{POP}) of chromosomes each of which is a solution to the problem. Similar to the biological process, reproduction in GA's can be of two types; sexual and asexual. In sexual reproduction, two parents combine and exchange genes to form offsprings, whereas in asexual reproduction, a single parent creates one or more offsprings. Operators analogous to the evolutionary process such as crossover and mutation are used to obtain genetic diversity among the population. Most of the times the selection of these operators is based on the type of reproduction involved. In the next few paragraphs, the working of sexual and asexual reproduction processes in GA is presented.

In sexual reproduction the GA's work similar to that of human genetics. From the initial population, the fitness values for each of the chromosomes are computed. Based on these values, chromosomes are selected to perform *crossover and mutation* operations. Crossover is performed on two individual chromosomes by using a crossover operator to form two new off-springs. This operator forms the basis for the sexual reproduction. Mutation is the process of altering the genes of a single chromosome to obtain a new chromosome. Different rates are used to determine the percentage of the population undergoing these operations. Once crossover and mutation operations are performed, chromosomes to transition to the next generation are chosen. This loop is continued until the termination criterion is reached. The best chromosomes are retained to asses the quality of the solutions to the problem being addressed.

In asexual reproduction, the operation of the GA is similar to that what occurs in plants and some single-cell species such as bacteria (Mitchell, 1998). The fitness value for every chromosome of the initial population is computed. Genes in the chromosome are rearranged, i.e. *only mutation* and offspring are formed by a single parent. Various strategies of chromosome rearrangement have been presented in the literature to perform

asexual reproduction. Once rearrangement and mutation operations are performed, the chromosomes are sorted to find the best individual. This loop is continued until the termination criterion is reached. In the research of this thesis, asexual reproduction with two mutation operations was applied to find best solutions.

The following pseudo code describes the general working of a GA with asexual reproduction, (crossover is not performed):

```
t = 0;
initialize (K(t=0));
evaluate (k(t =0));
While notTerminated() do
Kp(t) = k(t).selectparents();
Kc(t) = Mating(Kp);
Mutate1(Kc(t) );
Mutate2(Kc(t) );
evaluate(Kc(t));
K(t+1) = buildnextgenerationfrom(Kc(t), k(t));
t=t+1;
end
```

A number of approaches have been utilized in the application of GA to scheduling problems (Willis et al., 1997). The combination of selection, rearrangement, crossover and mutation methods greatly affect the performance of a GA. During past years, several methods were developed from chromosome representation to crossover and mutation methods.

Encoding is the process of transforming information from one format to other (Cheng et al., 1999) and representing the solution to a problem in the form of a chromosome. In GA's, encoding a solution into chromosome is a major issue. Several encoding techniques were created in literature, to which classical GA was difficult to apply directly

(Yamada and Nakano, 1997). During recent years the following nine representations for the job shop scheduling problem have been proposed:

Table 2-1 Chromosome Representations in Job Shop

Direct Representation	Indirect Representation
1) Operation based	1) Preference list based
2) Job based	2) Priority rule based
3) Job pair relation based	3) Disjunctive graph based
4) Completion time based	4) Machine based
5) Random keys	

These representations can be grouped into two categories, direct and indirect. If a schedule is encoded into a chromosome and GA's are used to evolve the chromosome to find a better schedule the approach is defined as direct (see Davis, 1985; Falkenauer and Bouffouix, 1991; Della et al., 1995). On the other hand, if a sequence of dispatching rules are encoded and if GA's are used to evolve those chromosomes to find a better sequence of dispatching rules then the approach is termed indirect (See Nakano and Yamada, 1991; Giffler and Thompson, 1960). In this research, job based representation was applied due to its simplicity and suitability to the problem (as the desired output was individual job schedules based on their priorities). Also this method avoids confusion in decoding the chromosome.

Selection method plays an important role in working of GA. A good selection method ideally identifies better chromosomes from a given population to advance to the next generation.   Crossover operation (Gen and Cheng, 1999) is performed in sexual reproduction to generate off-springs from two parent chromosomes. They include single cut point crossover (Gen and Cheng, 1999), Heuristic crossover (Blazewicz et al., 1996), partially mapped crossover, order-based crossover etc. In this research, asexual reproduction is performed therefore no crossover is used.

The offsprings in asexual reproduction are created by chromosome rearrangement methods performed on a single parent. These methods include inversion, transduction, transformation, conjugation, transposition, translocation etc (Holland, 1975). This operation as mentioned earlier can be called as a mutation operation. However, these rearrangement operations are much efficient that the simple mutation operations (performed when a crossover is used) (Mitchell et al., 1994). Hence they can produce efficient solutions even by not having the crossover operators. In this research we are using the inversion method for the chromosome rearrangement (mutation 1 operation). The reasons are to develop a simple, effective GA that can produce good results for large combinatorial optimization problems. Another reason is to enhance the efficiency of the adaptive GA by inversing a string of genes there by producing an entire new offspring (creating diversity in search space).

Several mutation operators exist in literature. These include insertion, displacement, reciprocal exchange mutation etc. In this thesis reciprocal exchange mutation is used as the mutation 2 operation. The reason for using this method is to reduce the great variation in chromosome properties, and still trying to avoid the local minima.

Literature provides few works performed in GA using asexual reproduction. The earliest works of Mitchell et al. (1994) and Banzhaf et al. (1998) stressed the importance of analyzing the effectiveness of the chromosome rearrangement methods as when implemented and used with a GA, improve its performance (Mitchell, 1996; Mitchell et al., 1994). Inspired by these works, several authors have used chromosome rearrangement mechanisms besides crossover and mutation for GA's. The methods such as inversion (Holland, 1992), conjugation (Harvey, 1992) transduction (Furuhashi et al., 1994; Nawa et al. 1997; Nawa and Furuhashi, 1998; Nawa et al. 1999) translocation (Oates, 1999) and transposition (Siomes and Costa, 1999) were previously used as the main genetic operators in the GA. All of these works proved the effectiveness of the asexual reproduction methods.

The application of asexual methods for hard combinatorial optimization problems was performed by few researches. Braun (1993) developed a genetic algorithm based traveling salesman problem with purely two mutation strategies. Their results produced good solutions for the traveling salesman problem. Following this work, Chatterjee et al. (1995) developed a GA with asexual mutation for the traveling salesman problem through a generalized mutation strategy. The algorithm was applied to natural and artificial problems. Their results produced good solutions. Chakroborthy and Mandal (2005) developed an asexual GA for the general single vehicle routing problem. Their algorithm was mutation based and could handle various types of vehicle routing problems. The results produced optimal and near-optimal solutions for forty six related problems from literature with less computational effort.

All the above works show that asexual reproduction methods could be used to solve combinatorial optimization problems with certain additional benefits of low computational time. However in Job shops, very few works have been performed using asexual reproduction. The work of Tay and Kwoh (2005) applied the Clonal Selection principle of the human immune system to solve the Flexible Job-Shop Problem with recirculation. While focusing on various practical issues, their method was based on an antibody representation that creates only feasible solutions and a bootstrapping antibody initialization method. They also developed a novel way of using elite pools to prevent premature convergence. The results of their study were obtained against benchmark FJSP instances. Cornforth (2007) developed an approach that combines a multi-agent system in dynamic environment to obtain best solutions with respect to completion time and cost. He used two methods; sexual and asexual reproduction. The results confirm the advantage of evolutionary optimization agent rules in a static or dynamic environment. His work focused on all types of dynamic system including job shops. Arthur et al., (1994) solved a multi-processor scheduling problem, a variation of job shop problem using a GA. They developed a serial and a parallel model GA to solve the multiprocessor scheduling problem. They used the asexual reproduction for their parallel GA and their results proved a better performance of parallel GA as compared to serial GA in finding optimal solutions.

In asexual reproduction feasible schedules are produced all the time. Therefore the vast amount of computational time involved in repairing infeasible solutions is reduced. Also this method simplifies the complexity of the developed adaptive job shop problem to a greater extent. Literature during recent times provided information on application of 2 mutation rates for GA's to solve for optimization. However, the efficiency of the crossover and mutation operators was always a discussed area in GA's. But for the type of situation dealt in this research which needs efficient results within less computational time the asexual methods was found to be efficient to solve small size problems with global convergence efficiency in less computational time.

The tree diagram in Figure 2-1 provides several solution approaches for solving the job shop scheduling problem (complied from Jain, 1998).
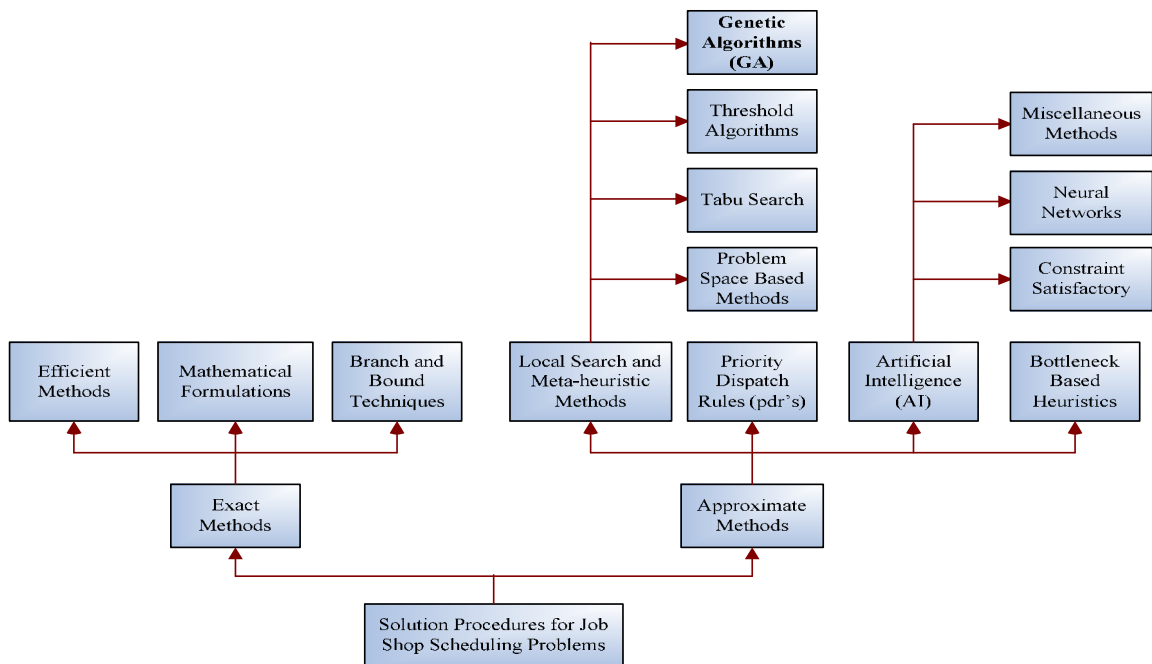
Figure 2-1 Solution Approaches for Job Shop Problems

2.3 Multi-Objective Job Shop Scheduling with Genetic Algorithms

In real-world production environments scheduling must be done often to achieve several objectives simultaneously. Multi-objective optimization aims at optimizing several performance criterion of an objective function vector (Belton and Elder, 1996). These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term" optimize" means finding such a solution which would give the values of all the objective functions acceptable to the designer" (Osyczka and Andrzej, 1985). These types of problems differ from the single-objective problem, in a sense that the multi-objective problem does not have a single best solution. One of the approaches to deal with these solutions is the Pareto method. In this approach, a set of solutions known as the Pareto-optimal solutions are usually formed. Any solution of this set is optimal with respect to certain condition that is no improvement can be made on one objective without degrading the other objective of the vector (Suresh and Mohanasundaram, 2005). Pareto Front is the line joining the minima of each of the Pareto points.

In this research mean flow time and mean tardiness are considered as the two objectives to be optimized. While these both are conflicting objectives, there is a need for human judgment to find a balance between them. Flow time is a critical indicator of manufacturing lead time (Aldakhilallah and Ramesh, 2001). Also the work in process (WIP) levels is proportional to the flow time. On the other hand, mean tardiness is related to customer-delivery performance. Enhancing this measure retains the customers as well as maintains the goodwill (Asano and Ohta, 2002). Therefore minimizing the above two performance measures enhances the profitability in a direction which most of the today's manufacturing companies aim at.

Several techniques have been developed over the past years in operations research literature to solve the multi objective optimization problem. Rosenberg (1967) suggested, but not performed, a genetic search for the simulation of genetics and the chemistry of population of a single–celled organism with multiple properties or objectives. The early work of GA application was performed by Schaffer (1984) who applied VEGA approach

in his algorithm. Since then many techniques were developed (Coello, 2000) to deal with multi-objective fitness functions ranging from naïve combination methods to game theory strategies. Some of the well known approach applied to multi-objective problems are weighed sum approach, e - constrain methods and goal programming (Srinivas and Deb, 1994). In this research, weighted sum approach is applied to formulate the multi-objective fitness function value as it is based on adapting to priorities which could be expressed through weights easily. Computationally, this approach is very easy, effective and can generate a strong set of non-dominated solutions.

Itoh et al. (1993) developed a two fold lookahead search method to solve the classical job shop problem. The objective of his research was to minimize mean tardiness and mean flow time in the scheduling system. Iima et al. (1999) proposed an autonomous decentralized scheduling algorithm for a complex job shop problem having sequencing dependent set up times and a parallel station having a single and a multi-function machine. The objective is to minimize the total tardiness and to maximize the working time of the multi-function machine. Esquivel et al. (2002) studied on the generation of Pareto optimal schedules in classical and flexible job shops. Balas, et al. (1998) worked on solving the job shop scheduling problem with due dates. His work was on optimizing a bicriteria problem involving minimax objectives based on the terminology of T'Kindt and Billaut (2002). Some of the recent papers include the work of Vilcot, Esswein and Billaut (2006) who presented the multi-objective optimization problem for the flexible job shop in printing and boarding industries. Tagour and Saad (2002) developed a GA approach for multi-objective optimization in Agro–alimentary workshop. They considered the cost of the outdated products, the cost of the distribution discount, makespan and the initial cost of production as their performance measures.

Among several search methods, GA's are particularly suited for multi-objective optimization, as they can explore the solution space in multiple directions. They not only have the ability to find global optima but can also be worked with complex fitness functions including discontinuous and noisy functions. Most of the optimization techniques need prior information about the problem. Since GA's use a class of points

they may be able to find multiple pareto points easily. This encourages researchers to apply GA's to solve these types of problems.

While the limitations of the GA's are due to stochastic errors associated with genetic operators the GA's tend to converge to a single solution with finite population. Also sometimes the chromosome representation might be difficult in certain applications. As described earlier, the performance of GA greatly depends on the choice of fitness function, the other parameters of a GA – population size, crossover and mutation rates, the type and strength of selection process. These parameters must be chosen properly to obtain best outcome of the method (Cohon and Marks, 1975). Premature convergence problems occur in small populations, and hence care must be taken in considering these types of populations.

2.4   Reactive Scheduling

Most of the work published in scheduling literature is predictive or of pre-assumed nature. During recent times a great deal of effort has been made in generating job shops schedules that can overcome both stochastic and dynamic disruptions of the production floor (Raheja and Subramaniam, 2002). These disruptions can vary from small to large magnitude. However, majority of the uncertainties such as urgent jobs, machine breakdowns, unavailable resources etc. are prevalent in most of the production floors. In such situations, an adaptive scheduling procedure that can adjust itself to the urgent situations is necessary to efficiently deliver jobs in the production environment (Biskup and Piewitt, 2000). Recovery or repairing methods that adjust the predictive schedules to accommodate these minor uncertainties are necessary for successful scheduling (Subramaniam and Raheja, 2003). These procedures aim at reducing the time and resource consumption that takes place whenever a minor disruption occurs. In this section a brief review of the relevant work is presented.

One of the earlier works in this area is that of Holloway and Nelson (1974). They applied a multi-pass procedure to generate schedules in a timely manner. They presented the effectiveness of periodic scheduling/rescheduling in dynamic environment. Several

approaches based on Heuristic Methods, Artificial Intelligence (AI), Fuzzy Logic, Genetic Algorithms (GA), Neural Networks (ANN) were developed over past years (Raheja and Subramaniyam, 2003) to successfully repair the predictive schedules during disruptions. Among the heuristic based approaches the Right–Shift Rescheduling (RSR) and Affected Operation Rescheduling (AOR) (Szelke and Kerr, 1994) heuristic were most prevalent. The RSR heuristic essentially shifts the job operations forward in time scale to accommodate to the disruptions, whereas the AOR heuristic reschedules only the affected job operations. The underlying concept in AOR is to move the start times of the affected jobs forward in time scale while adhering to the constraints. This is performed to maintain the initial job sequence.

While discussing about fuzzy logic and AI methods literature shows a considerable amount of work being performed in this area on reactive scheduling. Multi-agents were used in most of AI related approaches where the intelligent system has the knowledge about schedule repairs. Among the ANN approaches procedures involving the training of neural networks in a single pass were most prevalent. Case Based Reasoning approaches were frequently used in Fuzzy Logic methods to identify the case that best suits the disturbed schedule (Shafaei and Brunn, 1999). Constraint based scheduling concepts were also widely used (Reeja and Rajendran, 2000).

Most of the GA based repair approaches used the crossover and mutation operators to accommodate the schedule disruptions there by generating better schedules. GA proved to be efficient in repairing the predictive schedules, but the high computational effort associated with these natural processes became an issue.

3 METHODOLOGY

This chapter focuses on the methodology used for adaptive scheduling of jobs in a job shop environment. A detailed explanation of the GA model developed to schedule jobs for multiple days is provided in this chapter. A brief description of the operation of the job shop system is also provided in this chapter.

3.1 Description Job Shop Scheduling Problem

This section provides a detailed description of the job shop scheduling problem dealt in this research. To formally define the problem using scheduling terminology, we have a set of n jobs to be processed on m different machines, where in each job has its own machining sequence. The objective of this research is to generate best job schedules that minimize mean flow time and mean tardiness. The problem is denoted by

$$J_m \mid prec \mid (\sum_{j=1}^{n}(C_j / n), \sum_{j=1}^{n}(T_j / n))$$

where;

J_m denotes a job shop with m machines,

$prec$ denotes precedence constrains on jobs

C_j denotes the completion time of job j

d_j denotes the due date of job j

T_j denotes the tardiness of job j

3.2 Operation of the Job Shop Scheduling System

In this section the operation of the job shop system is explained. The production plant operates for 8 hours everyday; five days a week. All jobs enter the system on a daily basis. For each day, a best schedule is to be established. If multiple copies of a job are available they are scheduled as a single batch. On the very first day of scheduling, it is assumed that all jobs enter the system fresh and are free to be scheduled on any machine. However, for any subsequent days, there will be jobs carried forward from the previous

day, some of which could be setup on machines and partially processing an operation and new jobs entering the system.

Once a schedule is performed for the first day, the incomplete job operations for that day are stored and retrieved back to be scheduled on the following day. For any following day the partially completed operations for jobs from the previous day are unchanged from their machines and given first preference. This implies that irrespective of any situation, all the jobs with partially completed operations are placed first in the scheduled. The remaining operations are scheduled along with the new jobs entering on that day. This explains that all the incomplete operations on a particular job are carried over and remain on the same machine the next day, i.e. no preemptions of operations for a job. Restricting jobs with partially completed operations from being reassigned reduces the set-up changeovers on machines.

The schematic below provides a brief description of the system operation.
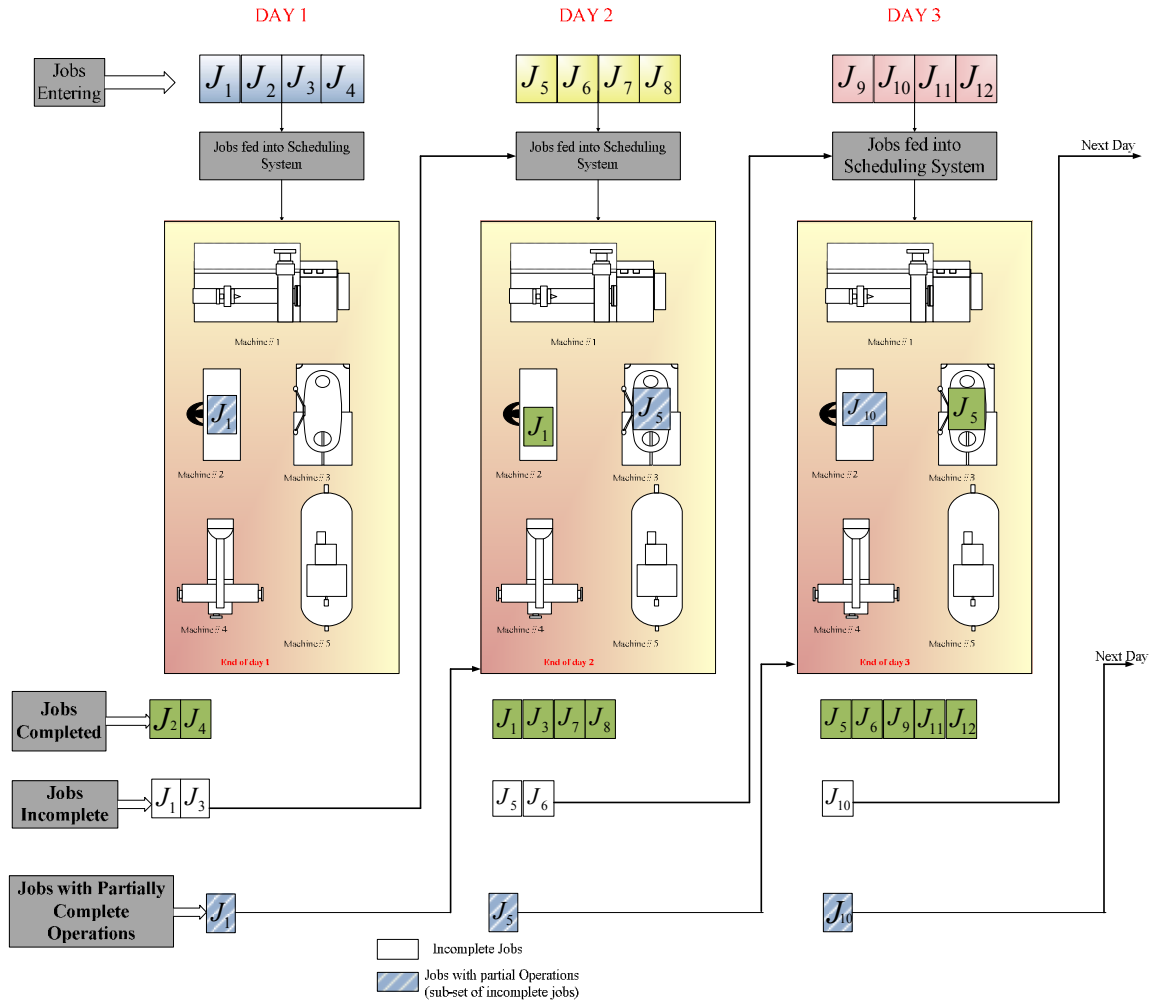
Figure 3-1 Job Shop System Operation

3.3 Multi-objective Genetic Algorithm for Job Shop Scheduling Problem

A detailed description of the multi-objective genetic algorithm developed in this research is provided below. The following assumptions are considered while formulating the solution approach for the job shop scheduling problem:

- All jobs are ready at the time of processing
- Preemption of jobs is not allowed
- Setup times are sequence-independent
- Setup times are added to processing times

- Shift Break times are not considered in processing times

The working of the MOGA for both the situations (First and Subsequent days onwards) is described through the following flowchart.

```
                          Start

                    Input Data
                    Acquisition

                                              Data Acquisition for
              Partial          YES            partial jobs and carried
              Operations?                     forward jobs

                    NO
              Create Initial                  Create Initial
              Population                      Population

        Create Start and End time and        Create Start and End
        Idle time matrices                    time and Idle time
                                              matrices

        Compute Weighed Fitness
        Function Values

                    Mating

                  Mutation 1

                  Mutation 2

              Retain Best Chromosome

  Select Best        NO     Termination      YES    Display
  Population                condition               Optimal Job        END
                            reached?                Shcedule
```

Figure 3-2 Flow Chart Representation of Working of GA

3.3.1 Chromosome Representation

The chromosomes representation for this problem is job based representation. In job based representation, a list of n jobs is formed. A schedule is constructed according to the sequence of jobs. For a given sequence of jobs, all the operations of the first job in the list are scheduled first, followed by operations of the second job which is followed by operations of the third job and so on. This process continues until all operations of job n are scheduled. The first operation of the job under treatment is allocated to the best available position in the schedule on the corresponding machine subject to the constraints. The process is repeated with all jobs in the sequence.

Each gene in a chromosome is divided into three parts for ease of job identification. The left most part represents the sequence index for jobs on a particular day. The second part of the string represents the job number, which is the unique source of job identification. The third part denotes the current day number. All the three parts can be represented only through numerical values. This type of representation aids in proper identification of jobs during multiple day scheduling. The chromosome representation for *first day jobs* and for the subsequent day*s* differs slightly. Hence, both the representations are separately explained below.
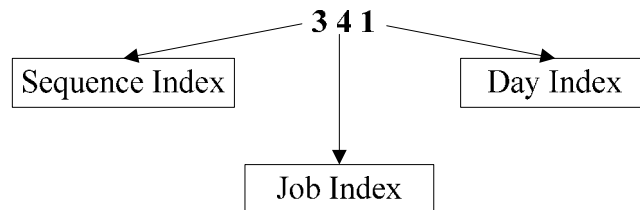
Figure 3-3 Gene Representation of a Chromosome

3.3.1.1 First Day Representation

Any permutation of jobs corresponds to a feasible schedule. The figure below describes the chromosome representation for jobs scheduled on day 1.

111	221	341	431

Figure 3-4 Chromosome Representation for First Day

3.3.1.2 Representation for following days

As this research involves dealing with adapting job schedules over multiple days the length of the chromosome is not fixed from one day to other. The job-based representation is applied with a slight modification for all following days. As the jobs with partially completed operations are prioritized each chromosome will have them assigned first to the sequence. The remaining part of the chromosome contains the previous day's jobs and incoming jobs for that particular day the sequence of which is determined randomly.

112	**111**	**252**	**221**

Partial Jobs from earlier days   New incoming jobs on day 2
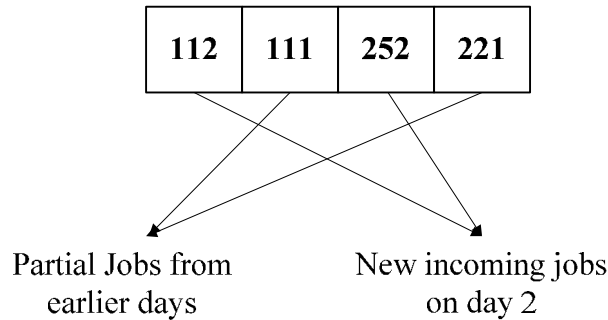
Figure 3-5 Chromosome Representation for Following Days

3.3.2 Fitness Function

The objective of this research is to obtain the best job schedule that minimizes the mean flow time and mean tardiness, the reasons for which were explained previously. Since this is a multi-objective problem the fitness function must incorporate both the performance measures. The mean flow time is the average of the times of all jobs spending in the system. Mean flow time is expressed as;

$$F(s) = \sum_{j=1}^{n} C_j / n$$

32

Where;

- S is a schedule
- F(S) – mean Flow time of the schedule S;

This research focuses on problem which involves changing jobs everyday; that is everyday we get different number of jobs and minimizing mean flow time would allow us to look at minimizing average flow time jobs in the system at any particular time. As makespan is related to completion time of last job, it world not be an appropriate criteria in this situation.

The second performance criterion in this research is mean tardiness. Tardiness is a performance measure related to the jobs due date. Mean Tardiness is given by the following equation where d_j denotes due date of job j. $T(S)$ is the mean tardiness for the schedule S.

$$T(s) = 1/n \sum_{j=1}^{n} \max(0, C_j - d_j)$$

Weighed sum approach is used to formulate the fitness function. Weighed sum approach assigns a weight w_j to each normalized objective function value to convert the multi-objective problem to a single objective problem with a scalar objective. The fitness function for the current problem is denoted as shown below.

$$\text{Weighed Average Fitness Value} = \alpha \, F(S) + \beta \, T(S)$$

$$\text{Weighed Average Fitness Value} = \alpha \sum_{j=1}^{n} C_j / n + \beta \, \max(0, \sum_{j=1}^{n} (C_j - d_j)) / n$$

$$\text{where,} \quad 0 \leq \alpha, \beta \leq 1 \text{ and } \alpha + \beta = 1$$

Based on the priority of the two objectives values of α and β could be chosen to obtain different schedules to solve the job shop scheduling problem.

3.3.3 Computations for Following/Subsequent Days

Any day other than the first includes jobs with partially completed operations and new jobs. Jobs with already setup/partially complete operations remain on their respective machines. The sequence to process the new jobs and other operations for jobs carried forward from the previous day are established using the chromosome. The start and end times for all operations on all machines is completed accordingly to determine fitness value.

3.3.4 Reproduction Probability

The multi-genetic optimization in this research involves a minimization problem. Therefore we cannot use the weighted fitness function value to directly select chromosomes. Therefore a reproduction probability is computed for each chromosome.

f_i = weighed fitness value for chromosome i

$Af_i = \sum f_i / f_i$ = adjusted fitness value

$P_i = Af_i / \sum Af_i$ = reproduction probability

An example is illustrated in Table 3-1 below.

Table 3-1 Reproduction Probability Calculation

Chromosome	Weighted Average Fitness Value (f_i)	Individual Adjusted Fitness $Af_i = \sum f_i / f_i$	Reproduction Probability ($Af_i / \sum Af_i$)
Ch 1	8.45	5.00	0.134
Ch 2	8.11	5.21	0.140
Ch 3	6.22	6.79	0.183
Ch 4	7.62	5.85	0.158
Ch 5	6.00	7.04	0.190
Ch 6	5.87	7.19	0.194
Cumulative	$\sum f_i = 42.28$	$\sum A_i = 37.09$	Cumulative Probability = 1

From Table 3 -1, it is clearly observed that the probability of the best chromosome is higher as compared to others converting the original minimization problem to an equivalent of a maximization problem.

3.3.5 Mating

Mating procedure is applied to randomly select pairs of chromosome to perform the genetic operations. In this research no crossover is performed, therefore mating is not really necessary. However, we want to explore an alternative of an elitist strategy in selection. To facilitate that chromosomes are considered in pairs for the genetic operation.

If we have N chromosomes, the number of parent pairs formed is N (N-1)/2. Each parent pair is given an equal probability to be chosen for crossover operation.

3.3.6 Mutation Operation

Two types of mutations are performed in this research to explore the solution space globally and locally which are briefly discussed here:

a. Chromosome Inversion (Mutation 1)

In this method a string of chromosome from a single parent is randomly selected at a cut-point. The offspring is formed by reversing all the genes from that cut-point and added to the original parent's before cut-point. Based on the rate of mutation 1 this operation continues on every chromosome. Figure 3-6 illustrates an example of this process.
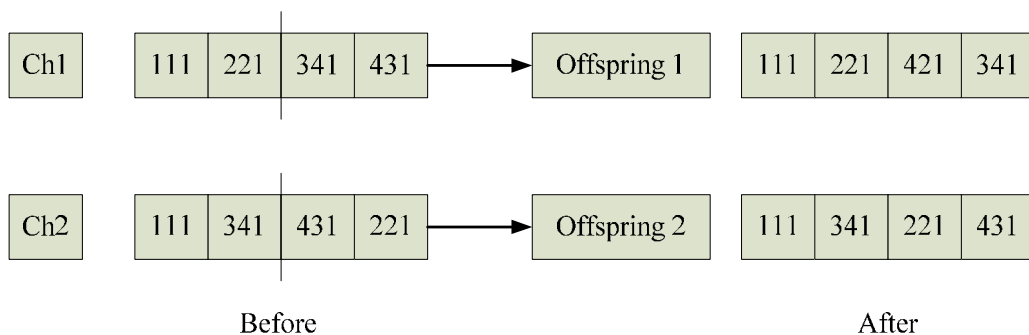
Figure 3-6 Mutation 1 Operation

For any following day the mutation 1 operator is applied as above, except that the operations already setup on machines for carried forward jobs are kept fixed.

b. Reciprocal Exchange Mutation (Mutation 2)

The second mutation operation in this research is the reciprocal exchange mutation. . In this method two genes of a chromosome are randomly selected and their positions are swapped. Based on the rate of mutation this operation continues on every chromosome. Figure 3-7 explains the mutation procedure on chromosome 1.

111	221	**341**	431

Before

341	221	**111**	431

After
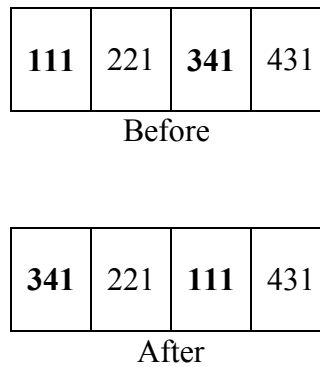
Figure 3-7 Mutation Operator

For any following day the mutation 2 operator is applied as above, except that the operations already setup on machines for carried forward jobs are kept fixed.

The application of two mutation strategies is very beneficial to obtain genetic diversity and to avoid local optimum. The normal mutation in a genetic algorithm only exchanges very few gene properties and is not very efficient in finding globally diverse solutions. However, a second mutation like inversion helps in generating more diverse/different chromosomes and hence enhances the efficiency.

3.3.7 Selection

The population size remains constant among the generations. The initial population is selected randomly for the first generation. From second generation onwards, the

population for a generation is selected at the end of previous generation. In this research, three methods were used to select the population for subsequent generations.

Strategy 1

In this strategy, the chromosomes after mutation are sorted and the N (Initial population) best chromosomes from all the parents and offspring are selected as initial population for the next generation. The number of chromosomes selected is equal to the population size.

Strategy 2

In this strategy, the chromosomes after mutation are passed on to the selection procedure. This procedure selects the best chromosome from the two parents and the two offspring's to be passed onto the next generation. Thus, an elitist strategy is applied to select the best from each pair and discard the weaker ones. Figure 3-8 shows an example.
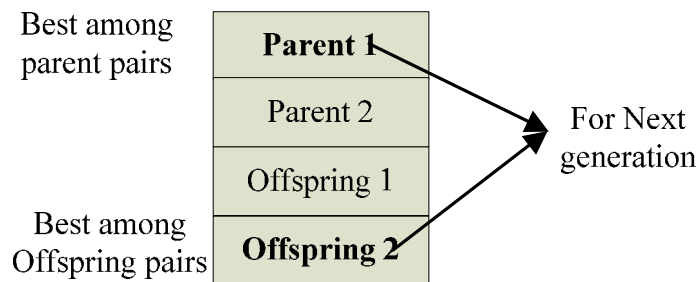
Figure 3-8 Strategy 2 Description

Strategy 3:

Strategy 3 is a combination of strategy 1 and 2. With this method, the part of the chromosomes is chosen by strategy 1 and the reminder by strategy 2, based on a user defined percentage.

3.3.8 Job Shop Scheduler Software

The MOGA with the features explained above was developed using the Visual C++ software. The input data acquisition is performed through a user interface in visual C++.Net 2003. This section explains the operation of MOGA software program for the job shop scheduling problem. The operational procedure for the first day is different to that of following days. Hence a step by step description of the operations of the software for both the situations is provided below.

3.3.8.1 Procedure for First Day

The following information in Table 3-2 is gathered through the data acquisition initially:

Table 3-2 Input Data Acquisition

GA Parameters	Job Information
1) Initial Population	1) Number of New Jobs
2) Number of Jobs	2) Number of Machines
3) Mutation 1 Rate	3) Processing Times for each job
4) Mutation 2 Rate	4) Due Dates for job
5) Selection Strategy (S1, S2 or S3)	5) Number of days job is in the system
6) Number of Trials	
7) Weight for Mean Flow Time	
8) Weight for Mean Tardiness	

The user interface to collect GA information (column 1 in Table 3-2) is shown in Figure 3-9.
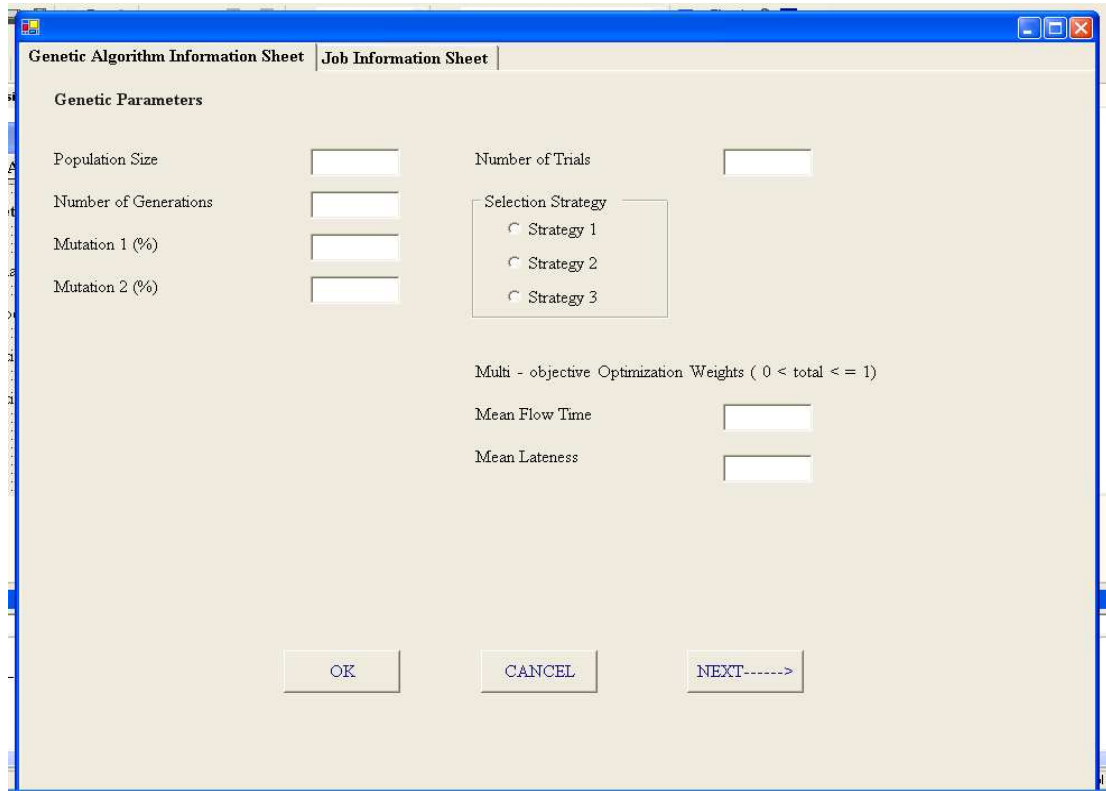
Figure 3-9 Screenshots of Graphical Interface; Input Data Entered

The second interface collects job related information (column 2, Table 3-2) as shown in Figure 3-10.

Figure 3-10 Processing Times Information

When the interface for job information is accessed, the user is prompted to add information on jobs with partially complete operations and other job related data. . As the first day contains no jobs with partially completed operations a value of zero is entered. Once these details are entered the button "PROCEED TO ENTER PROCESSING TIMES" creates a data grid for the given values. This button also guides the user to enter the job identity, processing times and corresponding due dates. The job identities are entered according to the gene representation described in section 3.3.1. Also the due dates are given in days. Processing time and machine sequence for jobs is entered using the OPERATION i (i = 1, 2, 3…. #of machines) columns. The machines and processing times on each machine for every job are entered in the following format machine # (Processing time). For example in Figure 3-10 job 111 is first processed on machine 2 for 100 minutes, machine 3 for 80 minutes and so on.

41

The total number of days a job has been in the system is also captured using the "DAYS IN SYSTEM" field.

Once the data is entered, the MOGA is run for required number of generations. The output is generated in the form of the job sequence and an array for start and end times and idle times on each machine. The start and end time array sequentially records the job number, followed by a listing of all operations listing the machine number, start time on that machine and end time as shown in Figure 3-11.

```
11. POST MUTATION - START AND END TIMES .txt - Notepad
File  Edit  Format  View  Help
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    221    3    0    800    2    800    1500    1    1500    2300    111    2    0    100    3    800    880    1    2300
    221    3    0    800    2    800    1500    1    1500    2300    111    2    0    100    3    800    880    1    2300
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200

Here are Start and End times of all jobs

    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    221    3    0    800    2    800    1500    1    1500    2300    331    1    0    200    2    1500    2300    3    2300
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    221    3    0    800    2    800    1500    1    1500    2300    331    1    0    200    2    1500    2300    3    2300
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    221    3    0    800    2    800    1500    1    1500    2300    331    1    0    200    2    1500    2300    3    2300
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    111    2    0    100    3    100    180    1    180    1080    331    1    1080    1280    2    1280    2080    3    2080
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    221    3    0    800    2    800    1500    1    1500    2300    331    1    0    200    2    1500    2300    3    2300
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    331    1    0    200    2    200    1000    3    1000    1780    111    2    0    100    3    100    180    1    200
    221    3    0    800    2    800    1500    1    1500    2300    331    1    0    200    2    1500    2300    3    2300
```
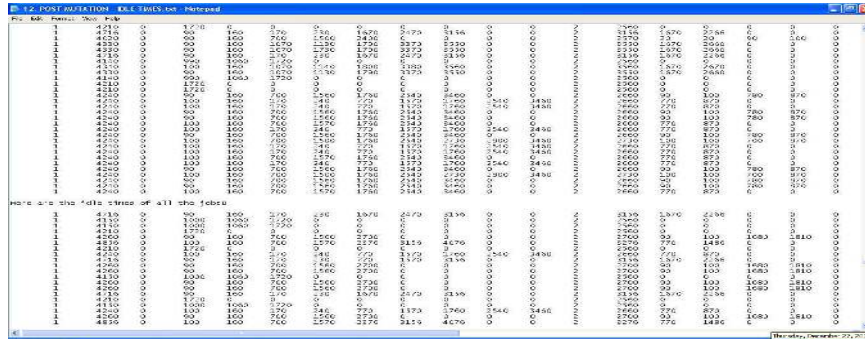
Figure 3-11 Screenshot of Start and End Time Matrix for First Day

Similarly, the idle times on each machine are stored in a separate matrix. The idle time matrix lists the machines in sequence, the start and end of idle time on that machine when a job is not being processed, as shown in figure 3-12. This idle time matrix is to be used to update the job operations and to create Gantt charts. The idle time matrix also assists in identifying the validation of operations and helps in further modifications.

42

Figure 3-12 Screenshot of Idle Time Matrix for Subsequent Days

Both the start-end time and idle time matrices are created for every chromosome of the initial population and are updated after every operation. The mean flow time and mean tardiness for every chromosome is calculated from the start and end time matrix. The computed individual values are normalized and used for fitness value computation. Reproduction probability, mutation 1, mutation 2 and selection process for the next generation are performed as described in sections 3.3.3, 3.3.4, 3.3.5 and 3.3.6.

The best chromosome and corresponding fitness value for each generation is saved and retrieved when needed. This procedure continues for given number of generations. An option of running the process for a given set of trial is also provided to ensure repetitive experimentation with same parameters.

3.3.8.2   Operation Procedure for Subsequent Days

To process the jobs for subsequent days, the computations are slightly more complicated and additional information is required. All the input information listed above along with the following data is required at this stage:

- Number of jobs with partially complete operations
- Number of Jobs incomplete
- Identity of jobs with partially complete operations and machine details
- Due dates for jobs with partially complete operations

43

- Number of days the job with partially complete operations is in system
- Number of Jobs Carried Forward to next day
- Carried forward job identity, processing times and due dates

The same interface shown in Figure 3-13 is used for data entry. However, when a value greater than given is entered for number of jobs with partial operation, a new data acquisition box is opened to collect data on those jobs with partially completed operations as shown in Figure 3-13.

Figure 3-13 Screenshots of Interface; Jobs from day 2 onwards

The "Number of Partial Operations" field acquires the information of the job with partially complete operation and the days the job is in the system. The "New Jobs" field takes information about incoming jobs on a particular day. The "Carried Forward Jobs"

field takes remaining operations information for jobs from earlier days. The last two digits of each string are used for matching job identity. This is used in acquiring information to calculate the fitness values. The rest of the fields are filled in a similar manner as explained for the first day.

After acquiring the required information the initial population is formed from data given by user. All the computations remain similar to the operational procedure for the first day.

4 EXPERIMENTATION

This chapter presents the details of the experimentations conducted on the Adaptive, multi-objective genetic algorithm (AMOGA). Experimentation was conducted in two stages. In the first stage, single day (first day) experimentation is conducted. In the second stage, the multiple day experimentation is performed. The details of the problems tested are given below.

4.1 First Day Experimentations

Testing is performed on three different problems including the well known Fisher and Thomson FT06 problem. The purpose of this single day testing was to validate the effectiveness of the AMOGA in finding effective solutions to optimize the selected objectives.

4.1.1 Initial Testing Parameters

Several test cases were generated to test the performance of AMOGA. Table 4-1 shows the input parameters considered for the testing all the problems.

Table 4-1 Input Parameters

Index	Parameters	Values
1	Population Size	(i) 10 (ii) 20 (iii) 30
2	Number of Generations	(i) 50 (ii) 100 (iii) 200
3	Mutation 1 Rate	(i) 20 (ii) 40 (iii) 60 (iv) 80
4	Mutation 2 Rate	(i) 2 (ii) 5 (iii) 10

Three different strategies were considered for selection as described earlier. Table 4-2 explains the three selection strategies.

Table 4-2 Selection Strategy

	Strategy	Description
1	S1	Selects the best among all the parents and off-spring pairs
2	S2	Selects best among (parent+off-spring pair)
3	S3	A combination of S1 and S2

The objective of AMOGA is to minimize the mean flow time and mean tardiness. Since this is a multi-objective problem, several weight combinations for the individual objectives are tested. Table 4-3 explains the weights used for testing.

Table 4-3 Weights for Objectives

	Mean Flow Time	Mean Tardiness
1	0	1
2	0.2	0.8
3	0.4	0.6
4	0.6	0.4
5	0.8	0.2
6	1	0

4.1.2 Priority Rules

Priority rules are probably the most frequently applied heuristics for solving (job shop) scheduling problems in practice. This is because of their ease of implementation and low computational time. Several priority rules were developed to solve the job shop

scheduling problem. Among these the most commonly applied are Shortest Processing Time rule (SPT), Earliest Due Date rule (EDD) and Longest Processing Time rule (LPT). The SPT rule is most frequently applied to obtain the best job schedule with respect to flow time whereas the EDD rule is applied for tardiness objectives.

Since SPT and EDD rules are used for single objective optimization, they both are individually applied to the test problems. The individual results from the above priority rules are then compared to those from AMOGA by considering the extreme solutions, i.e. the best mean flow time, and the best mean tardiness of the Pareto optimal or near optimal solution set as the reference.

4.1.3 Test Problems

The three problems are presented in the following section. Details for each problem are provided separately.

Test Problems 1 – FT06 Benchmark

The FT06 benchmark problem was developed by Fisher and Thomson (1963). The problem involves a set of 6 jobs (1 through 6) to be processed on 6 different machines (M_1 through M_6), where every job has its own machining sequence. The due dates for this problem are adapted from the work of Ponnambalam, et al. (2001). Table 4-4 shows the processing times (min) and due dates (min) for the FT06 problem.

Table 4-4 FT06 Benchmark problem

Jobs	Processing time (min)[*]						Due dates (min)
1	3(001)	1(003)	2(006)	4(007)	6(003)	5(006)	52
2	2(008)	3(005)	5(010)	6(010)	1(010)	4(004)	94
3	3(005)	4(004)	6(008)	1(009)	2(001)	5(007)	68
4	2(005)	1(005)	3(005)	4(003)	5(008)	6(009)	70
5	3(009)	2(003)	5(005)	6(004)	1(003)	4(001)	25
6	2(003)	4(003)	6(009)	1(010)	5(004)	3(001)	45

[*]Note: 3(001) → First operation is on machine # 3, processing time 1 min

The FT06 benchmark problem is extensively tested in this section. The initial parameters are already provided in Table 4-1, Table 4-2 and Table 4-3. All possible combinations of the parameters from Tables 4-1, 4-2 and 4-3 were tested on this problem. Each parameter set is run for 100 trials for a specified weight set 2 (0.2, 0.8). The objective of this extensive experimentation is to identify the best parameter set which produces minimum weighed fitness value. Those parameters can then be used for the subsequent experimentations. The parameters in Tables 4-1 through 4-3 generate 1458 different combinations. Therefore initial experimentations were performed on 27 parameter combinations as shown in Table 4-5.

Table 4-5 Initial Parameter Combinations

Number of generations = 50, Mutation 1 Rate = 20%				
Population Size	**Mutation 2 Rate (%)**	**Strategy**		
		S1	S2	S3 (S1=65%, S2=35%)
10	2	S1	S2	S3 (S1=65%, S2=35%)
10	5	S1	S2	S3 (S1=65%, S2=35%)
10	10	S1	S2	S3 (S1=65%, S2=35%)
20	2	S1	S2	S3 (S1=65%, S2=35%)
20	5	S1	S2	S3 (S1=65%, S2=35%)
20	10	S1	S2	S3 (S1=65%, S2=35%)
30	2	S1	S2	S3 (S1=65%, S2=35%)
30	5	S1	S2	S3 (S1=65%, S2=35%)
30	10	S1	S2	S3 (S1=65%, S2=35%)

Based on the experimental results for this problem, the best parameters were formed. They were applied to another two problems to generate the best solutions.

Test Problem 2

The second test problem is a 4 job, 3 machine problem adapted from Johnson et al, (1974). The due dates were formed based on the total processing time of the individual jobs. Table 4-6 shows the jobs with the corresponding processing times, machining sequences and due dates in minutes.

Table 4-6 Test Problem 2

Jobs	Processing time (min)[*]			Due dates (min)
1	2(004)	1(006)	3(002)	21
2	1(005)	2(004)	3(002)	10
3	1(002)	3(003)	2(007)	2
4	2(004)	3(003)	1(005)	11

The experimentation on the test problem 2 was conducted by applying the best parameters generated from FT06 problem. For initial evaluation of AMOGA's performance the experimental results were compared with those from priority rules. In this phase, separate experiments were run by assigning highest priority to the desired objective and correspondingly evaluating the results. However to generate the Pareto solutions, the problem is tested for all the weight ratios listed in Table 4-3.

Test Problem 3

A third test problem with 3 jobs to be processed on 3 machines is considered. This problem is adapted from Pinedo (1995). The due dates for this problem are assigned in a similar fashion as shown for the second test problem. Table 4-7 shows the problem considered with processing times and due dates in minutes.

Table 4-7 Test Problem 3

Jobs	Processing time (min)[*]			Due dates (min)
1	1(005)	2(010)	3(004)	18
2	3(004)	1(005)	2(006)	24
3	3(005)	2(003)	1(007)	16

The experimentation procedure for test problem 3 is similar to that of test problem 2. Evaluation of AMOGA is performed by initially performing the experiments for single objective optimization. Also runs were conducted for each of the weights to generate the Pareto solutions.

4.2 Experimentation for Adaptive Scheduling

The experimentation for adaptive scheduling is conducted with the best parameter set identified after experimentation with the benchmark problems. The experiments are run for three successive days and certain specific characteristics and trends in AMOGA were gathered. The job details for each day and due dates are provided at the beginning of the scheduling process. The due dates for the jobs are assigned based on the total processing time of the jobs. Few of the shorter jobs were assigned to longer due dates and some of the longer jobs were given shorter due dates. This type of situation is considered to evaluate the effectiveness and flexibility of the AMOGA to the manufacturing environment. The experimentation in this section are continuous, that is the jobs that are left over on any day are carried forward to the next day along with the new jobs. Therefore we observe the scheduling process for a set of days and not for a single day.

4.2.1 Test Problem

A three day test problem was developed for experimenting on the adaptive job shop scheduling process with AMOGA as explained below.

The FT06 benchmark problem is slightly modified to generate data for the first day. Processing times were increased by a factor of 20 and due dates modified accordingly. Table 4-8 (a) shows the modified FT06 problem formed for experimental analysis.

Table 4-8 Test Problem for Mutiple day scheduling

(a) First day Job Information

Jobs	Processing times (min)						Due date (Days)
1	3(020)	1(060)	2(120)	4(140)	6(060)	5(080)	2
2	2(160)	3(100)	5(200)	6(200)	1(200)	4(080)	4
3	3(100)	4(080)	6(160)	1(180)	2(020)	5(140)	3
4	2(100)	1(100)	3(100)	4(060)	5(160)	6(180)	3
5	3(180)	2(060)	5(100)	6(080)	1(060)	4(020)	1
6	2(060)	4(060)	6(180)	1(200)	5(080)	3(020)	2

(b) Day 2 Job Information

Jobs	Processing time (min)*						Due date (Days)
7	1(020)	3(050)	4(050)	2(040)	5(030)	6(060)	2
8	4(090)	3(070)	1(120)	5(250)	2(025)	6(080)	4
9	5(090)	6(015)	3(030)	2(150)	4(025)	1(120)	3

(c) Day 3 Job Information

Jobs	Processing time (min)*						Due date (Days)
10	3(000)	2(030)	6(040)	1(070)	4(060)	5(050)	1
11	2(100)	1(080)	5(000)	3(040)	4(050)	6(190)	2

4.2.2 Experimentation

On the second day of continuous job shop scheduling process by AMOGA, a set of 3 *new jobs* (7,8 and 9) that are to be processed on 6 machines (1 through 6) are considered entering the system. The due dates and processing times for these jobs were generated. Table 4-8(b) describes the new jobs entering the system on day 2.

On day three 2 *new jobs* (10 and 11) enter the system as shown in Table 4-8(c).

The multi-day adaptive schedule was tested for the GA parameters identified as best through the single day experimentation for all weight contributions in Table 4-3. For each weight combination a set of 100 trials are run to identify the pattern of AMOGA performance and to generate the set of non-dominant solutions. However for comparison of performance for multiple days it is necessary to choose a particular set of weights for comparison. Therefore $\alpha = 0.2, \beta = 0.8$ is chosen.

At the end of the first day (480 minutes) for the schedule obtained from AMOGA will have jobs belonging to following categories.

- Category 1: Jobs whose last operation is partially completed

  All these jobs unaltered are and scheduled first in the AMOGA. Hence they are fed in a specified manner to the AMOGA during the experimentation process. The feeding procedure to the AMOGA is described earlier in chapter 3.

- Category 2: Jobs with a partially completed operation that also has other operations to be processed on one or more other machines.

  All the partial completed operations are unaltered and scheduled first on the following day similar to category 1. However, since these jobs have further processing requirements and those operations are scheduled along with new jobs using the GA.

- Category 3: Jobs with no partially completed operations

All the jobs under this category are scheduled along with new jobs using the GA. Therefore, at the end of the day all the remaining operations jobs in the system on that day are recorded and carried forward to the following day.

4.2.2.1 Day 2

The experimentation of all subsequent days is performed after identifying all the carried forward jobs and entering their data into AMOGA. Once the new jobs and carried forward jobs are fed into the system, experimentation is conducted with the same parameters and weights for objective function values for all subsequent days.

4.2.2.2 Day 3

The AMOGA was to determine the sequence to process jobs on the job shop as explained.

5   RESULTS AND DISCUSSIONS

This chapter presents the results obtained from the experimentations conducted on the developed AMOGA. The results are also compared with results published in the literature.

5.1   Single Day performance

In this section, the results for the problems tested for a single day are presented. The trends observed for each of the test problem are explained separately under the respective sections. The test problems were also solved by applying priority rules and results compared with those from AMOGA.

FT06 Benchmark Problem

Priority Rules

This problem is first solved by applying priority rules. We first apply SPT rule to identify mean flow time for the given jobs. From literature the minimum mean flow time obtained using SPT rule is **52.7 minutes** (Kaschel et al., 1999)

Similarly, lowest rule is applied to the FT06 problem to obtain best job sequence with respect to mean tardiness. Since there is no result in literature on the application of EDD to the FT06 problem, the schedule was calculated by applying the EDD concept to the problem. Figure 5-1 shows the best schedule obtained by applying the EDD rule. The minimum mean tardiness obtained for EDD sequence is **2.5 minutes**.
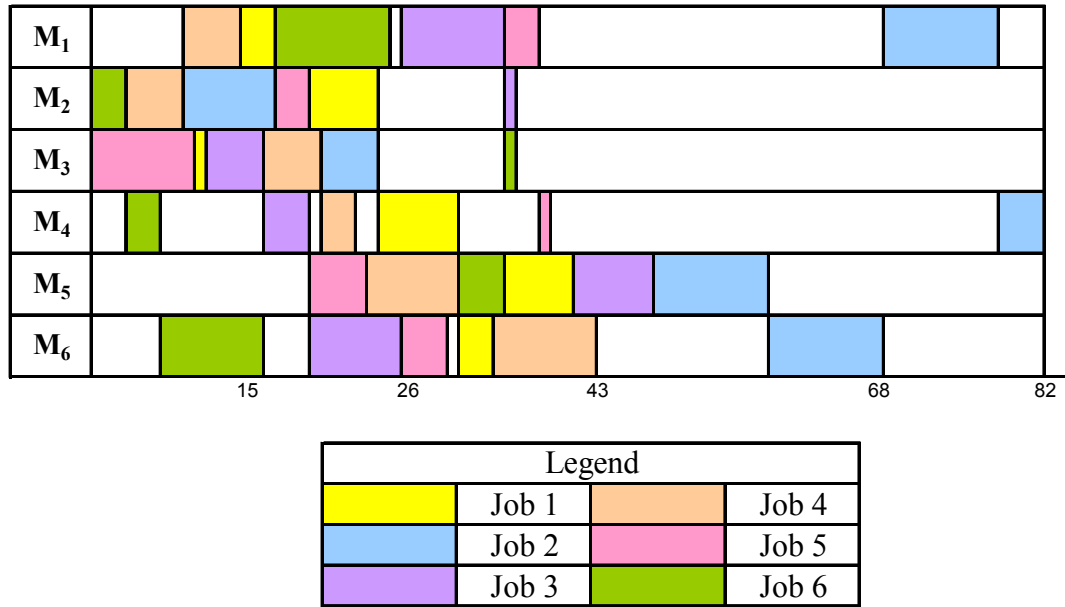
Figure 5-1 Gantt chart for the FT06 schedule using EDD rule

Results from Literature

Very few work from literature used mean flow time minimization (Ponnambalam et al., 2001) in their studies of single objective optimization. The work of Suresh and Mohanasundaram (2005) was based on multi-objective optimization using SA. In their work they considered makespan and mean flow time as the optimization criterion. The best mean flow time they obtained is **44.17 minutes**. There is no prior evidence on due date values on this problem.

Results from AMOGA

The AMOGA was then used to solve the FT06 problem. The experimentation is performed by taking a combination of 27 parameter sets from Table 4-1, Table 4-2 and Table 4-3 and running the AMOGA for 10 trials of 10 runs each. The objective of this extensive experimentation is to identify the schedule that produces minimum weighed fitness value and repeatability of solutions.

Table 5-1 shows the results obtained from the initial experimentations with 27 parameter sets with objective weights of mean flow time = 0.2 and mean tardiness = 0.8. The best selection obtained with this weight combination was mean flow time = 44.17 minutes and mean tardiness = 0.67 minutes. The values in the table are frequency at with AMOGA formed this solution after 100 trials. It must always be noted that there are different job sequences that gave the same result.

Table 5-1 Frequencies of Best Solutions

Population Size	Mutation 2 Rate (%)	Frequency		
		S1	S2	S3
	2	12	25	24
10	5	18	26	15
	10	16	25	20
	2	19	24	17
20	5	11	26	28
	10	27	26	28
	2	17	20	21
30	5	13	36	29
	10	23	30	33

To analyze these initial results further, several scatter plots were developed for different population sizes, mutation 2 rates, and strategies to identify the most effective parameter combination that generates best solution most frequently. Figure 5-2(a) shows the variation of results based on population size and Figure 5-2 (b) based on mutation rate 2 for the different selection strategies.
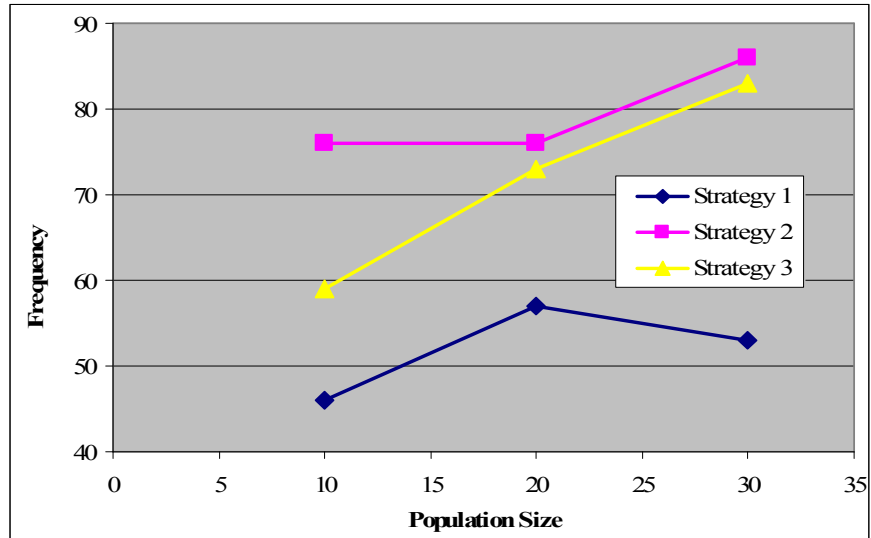
5-2(a): With respect to population size
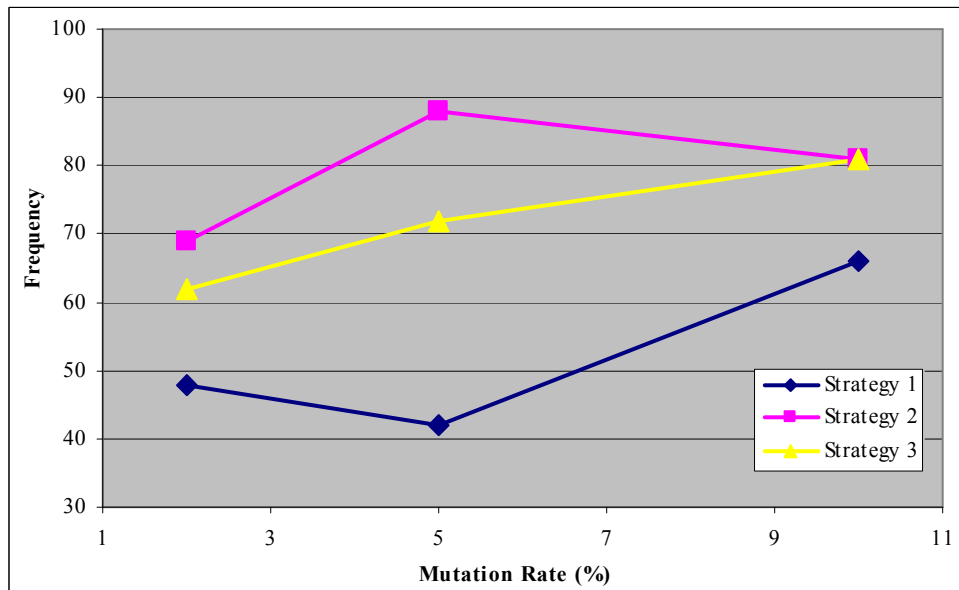
5-2 (b): With respect to mutation rates

Figure 5-2 Variations of Frequency of Best Solution

From Figure 5-2(a), it can be observed that strategy 2 is performing well for all the population sizes. From Figure 5-2 it can be observed that on an average all the strategies are performing better with 10% mutation rate while strategy 2 is generating better results

over all it always gives the highest frequency. Therefore strategy 2 with 10% mutation rate is expected to produce good solutions.

Based on the initial results the combination of strategy 2 a population size of 20 and 10% mutation rate is likely to produce good solutions. This experimentation was however conducted to mutation rate of 20% with 50 generations. To identify the performance of several mutation 1 rates and numbers of generation's further experimentation was conducted. Figure 5-3 shows the performance of strategies with respect to mutation 1 rates. From the figure it is clearly seen that the number of best solutions increase with increase in mutation 1 rate. Overall, a rate of 80% for mutation 1, for strategy 2 is performing the best.
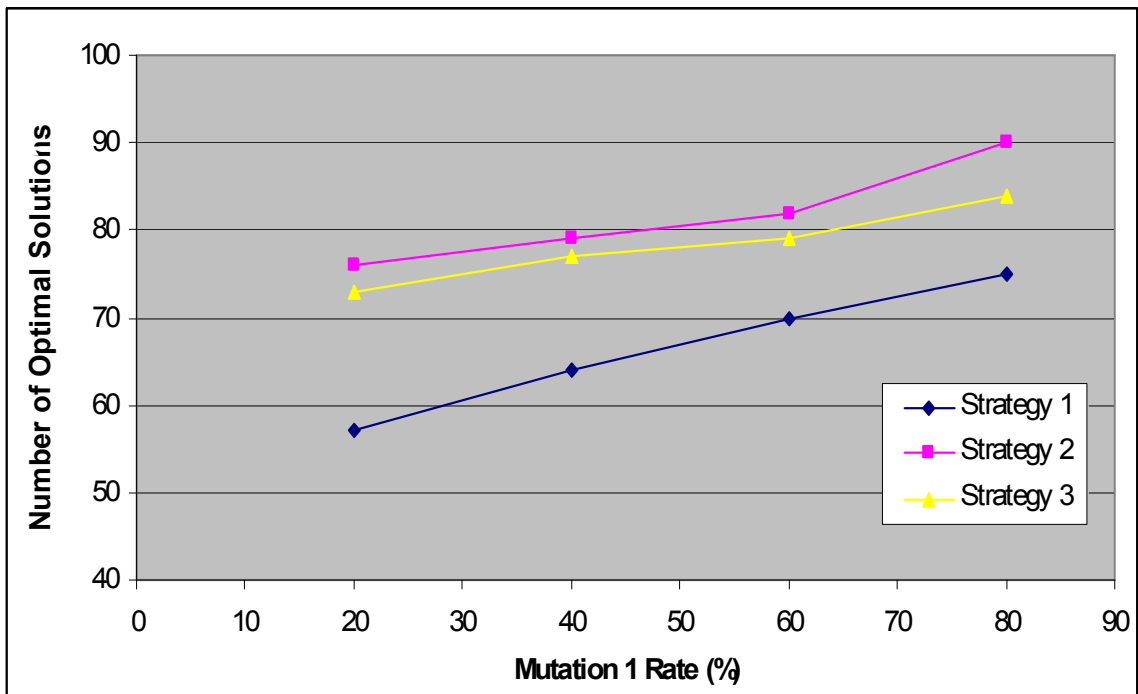
Figure 5-3 Variation of number of best solutions with respect to mutation 1 rate

Convergence diagrams were also used to further evaluate the performance of the AMOGA. For the above parameter set, the GA is run individually for 50, 100 and 200 generations. Figure 5-4 shows a comparison of convergence trends for each of the three generations considered. All testing was done with $\alpha = 0.2$ and $\beta = 0.8$ objective weight.
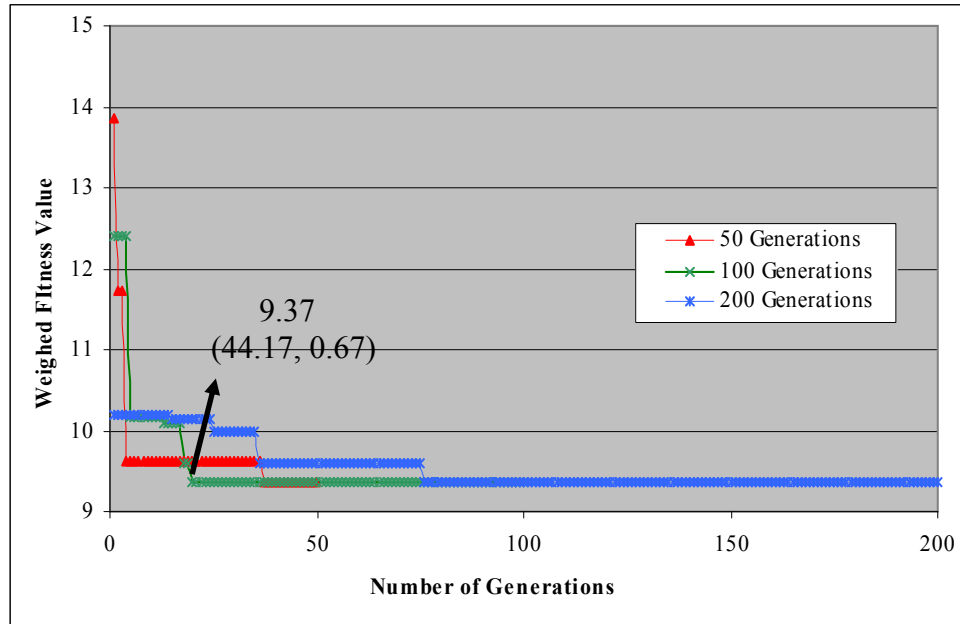
Figure 5-4 Convergence diagram for AMOGA problem

From the above convergence diagram, it can be observed that, in all three cases, the AMOGA is converging before 100 generations are completed. There is no improvement in the results after 100 generations. Therefore it appears that running the AMOGA for more than 100 generations, most likely does not have any significant benefit.

Based on the above results and analyses it can be deduced that Strategy 2 gives better results when compared to the other ones. This strategy combined with 100 generations, 80% mutation 1 rate, population size of 20, and mutation 2 rate = 0.1 is likely to present schedules that minimizes the mean flow time and mean tardiness for the developed

AMOGA problem. Therefore, these parameters will be used for all subsequent experimentation.

As the current research problem involves multi-objective optimization of mean flow time and mean tardiness, the best parameter set is run for 100 trials for each of the objective weights mentioned in Table 4-3. Table 5-2 shows the best mean flow time and mean tardiness value obtained from each weight combination. As mentioned earlier, a weight value of 0 makes the problem a single objective optimization.

Table 5-2 Pareto results for FT06 problem

Objective weight*		Best result	
Mean Flow time	**Mean Tardiness**	**Mean Flow time (min)**	**Mean Tardiness (min)**
0	1	53.50	0.00
0.2	0.8	44.17	0.67
0.4	0.6	44.17	0.67
0.6	0.4	44.17	0.67
0.8	0.2	44.17	0.67
1	0	44.17	0.67

Figure 5-5 shows the Pareto Front plot for the multi-objective optimization of the FT06 problem.
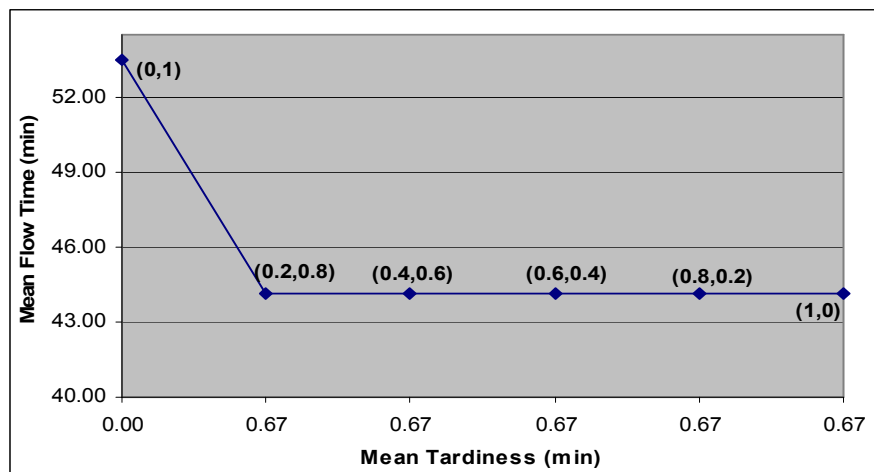
Figure 5-5 Pareto Front for FT06 problem (Weights indicated in parentheses)

From the above graph, it is clearly observed that as the weight for a given objective function increases the corresponding importance of that value increases. Thereby, greater emphasis is given to minimize that functional value. For example, as we vary the weights from (0, 1) minimization of mean tardiness to (1, 0) minimization of mean flow time problem the values for both flow time and tardiness vary accordingly.

From the Pareto results, the best job schedule from GA that minimizes the mean flow time is recorded. This is given by the job sequence $1 - 6 - 4 - 2 - 5 - 3$. A Gantt chart is created to indicate the position of jobs on a time scale. The minimum mean flow time obtained by AMOGA is **44.17 minutes**.

Visual representation of results is very useful in actual implementations. The assignments of jobs to machines, their start and completion times as well as idle times on machines are readily apparent with Gantt Charts. The following figures present ant Charts for three different situations:

   a) Minimizing mean flow time only (weights $\alpha = 1$ and $\beta = 0$)

The job sequence for this case is 1-6-4-2-5-3 and the mean flow time is 44.17 minutes.

   b) Minimizing mean flow time and mean tardiness (weights $\alpha = 0.2$ and $\beta = 0.8$)

The job sequence for this case is 1-6-5-4-3-2 and the mean flow time is 44.17 minutes and mean tardiness is 0.67 minutes

   c) Minimizing mean tardiness only

The job sequence for this case is 5-6-1-4-3-2 and the mean tardiness is 0 minutes.
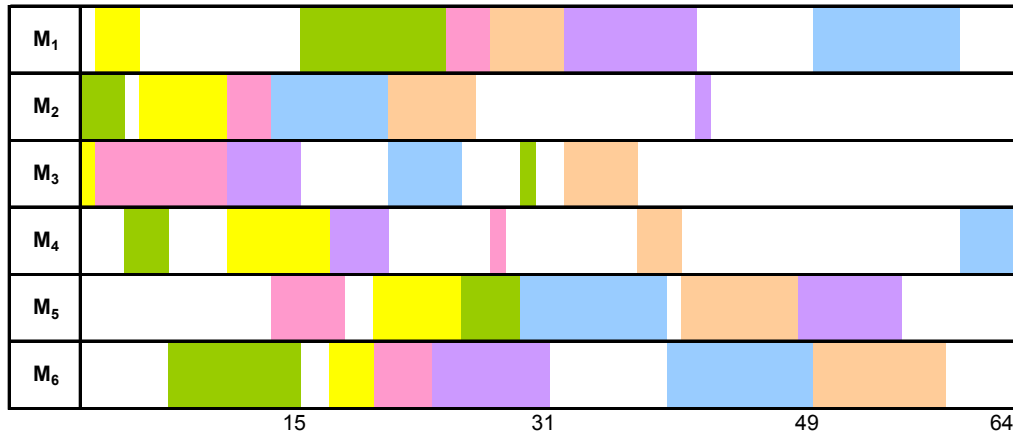
Figure 5-6 Gantt chart for minimum flow time schedule from AMOGA
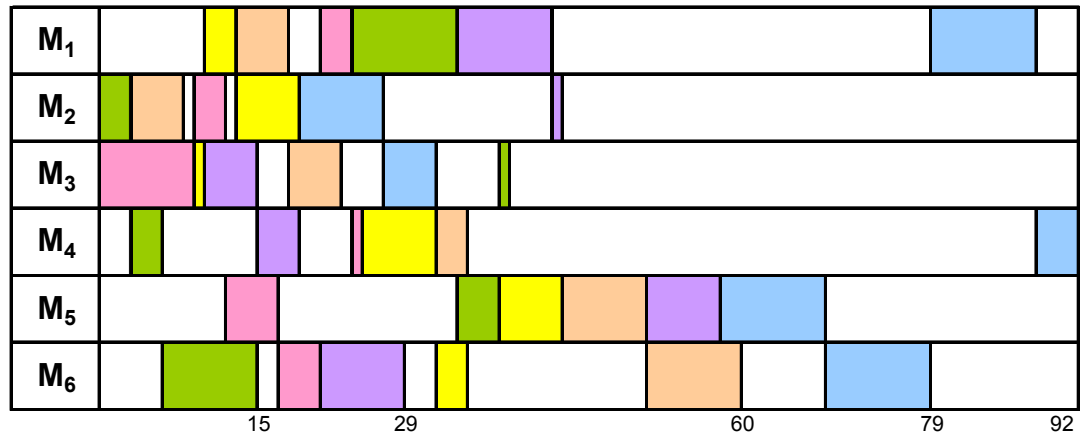
Figure 5-7 Gantt chart diagram for minimum tardiness schedule from AMOGA

63

Since there are no prior results to compare multi-objective performance, we are therefore comparing them with respect to individual objectives. From the results, AMOGA is performing well for single-objective case. Therefore we assume that it is performing well with multi-objective case. Although the environments are varied not much variation in results is obtained due to the smaller problem size.

Table 5-3 Comparision of Results

Optimization Criteria	AMOGA	Priority rules	Benchmark Results	Multi-objective Literature
Mean Flow time	**44.17**	SPT - 52.7	44.17	44.17
Mean Tardiness	**0**	EDD- 2.5	NA	NA

Test Problem 2

This 3 x 4 problem was described in the previous chapter. All the results after experimentation for this problem are explained below.

Results from Priority Rules

The test problem 2 is first solved by applying SPT rule to find minimum mean flow time schedule. The best job schedule is represented through the Gantt chart shown in Figure 5-8. The minimum mean flow time obtained through SPT is 16.75 minutes.
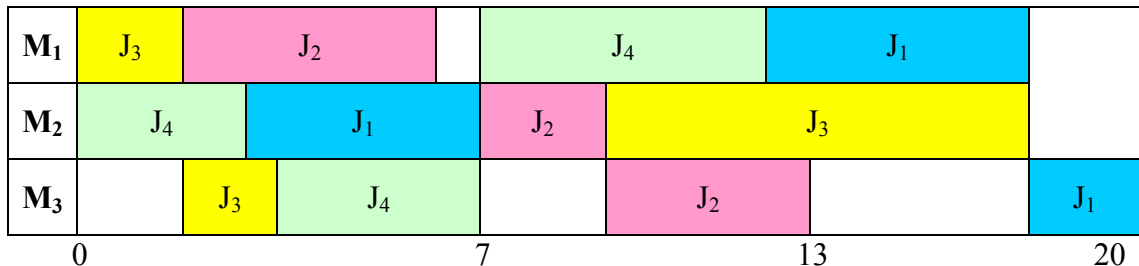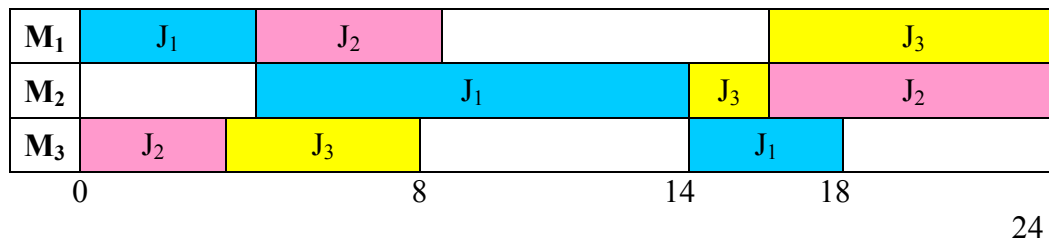

Figure 5-8 Gantt diagram for best job schedule from SPT rule

Similarly, the EDD rule is applied on the test problem 2 to find the best sequence that minimizes mean tardiness. The best job sequence obtained through EDD rule gives a value of 5.5 minutes. Figure 5-9 shows the Gantt chart schedule for minimum tardiness.
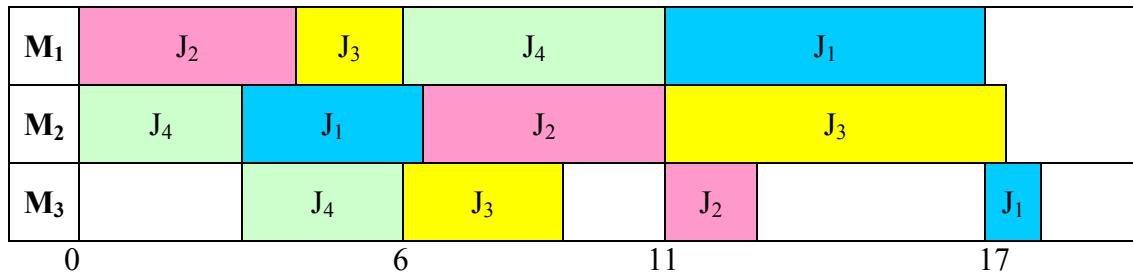
Figure 5-9 Gantt diagram for best job schedule from EDD rule

Results from AMOGA

AMOGA is used to solve the test problem 2. The test problem in AMOGA is solved for the weight combinations listed in Table 4-3. Table 5-4 shows the results obtained for each combination for 100 trials.

Table 5-4 Pareto results for Test Problem 2

Objective weight*		Best result	
Mean Flow time	Mean Tardiness	Mean Flow time (min)	Mean Tardiness(min)
0	1	16.25	5.25
0.2	0.8	16.25	5.25
0.4	0.6	16.25	5.25
0.6	0.4	16.25	5.25
0.8	0.2	16.25	5.25
1	0	16.25	5.5

As we observe, the best fitness values for weighs 1, 0 and 0, 1 is similar for mean flow time. The best job sequence for minimum mean flow time and mean tardiness is however different. As this is a very small problem the GA is able converge to the best solution very quickly. Also it is noticed that the best job sequence that minimizes mean flow time is different from best job sequence that minimizes mean tardiness. This shows that the objectives considered are diverging objectives, that is increasing one objective decreases the value of other objective.

The minimum mean flow time obtained by AMOGA from (1, 0) weight combination is 16.25 minutes. The best job sequence is given by $4 - 1 - 2 - 3$. Figure 5-10 shows the Gantt chart for the best schedule.
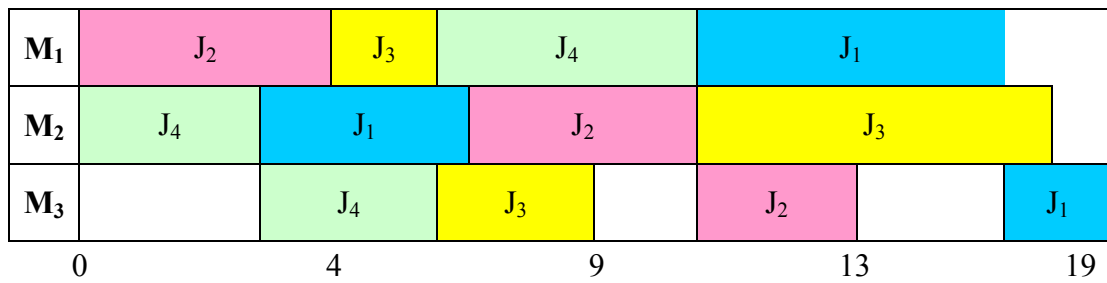
Figure 5-10 Gantt diagram for minimum mean flow time schedule from AMOGA

Similarly, the minimum mean tardiness obtained by AMOGA from (0, 1) weight combination is 5.25 minutes. The best job sequence is given by jobs $2 - 4 - 1 - 3$. Figure 5-11 shows the Gantt chart for the best schedule.
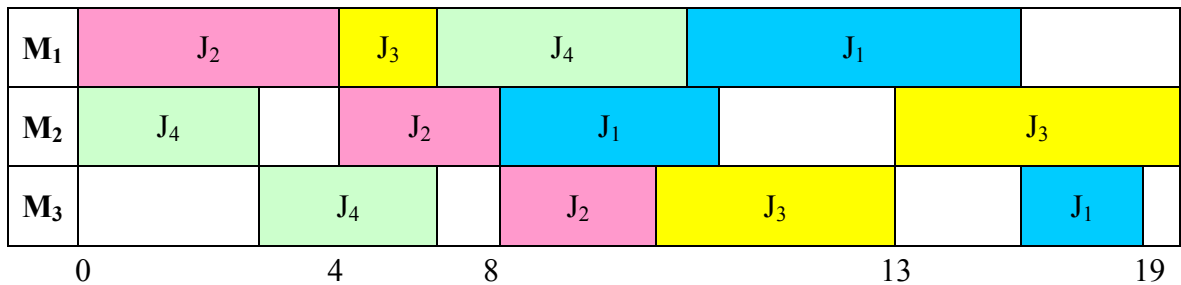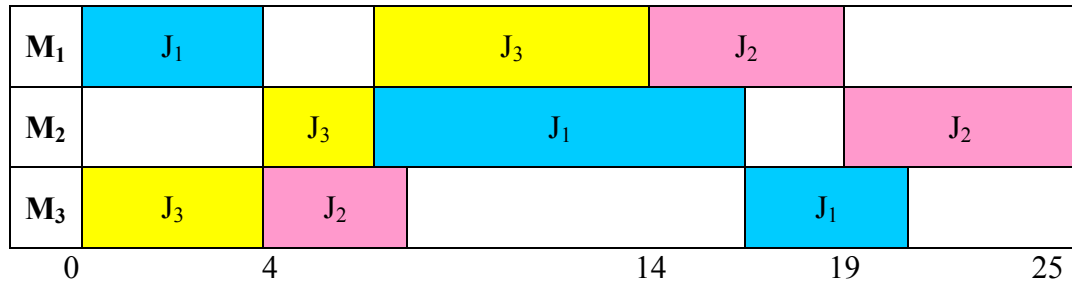

Figure 5-11 Gantt diagram for minimum mean tardiness from AMOGA

66

Since there are no prior results to compare multi-objective performance, we are therefore comparing them with respect to individual objectives. The performance of AMOGA is better than obtained best both SPT and EDD rule. This indicates that AMOGA is being able to converge to better solutions even for small sized problems. Table 5-5 summarizes the results from both the methods.

Table 5-5 Comparison of Test Problem 2 results

Optimization Criteria	AMOGA	Priority rules
Mean Flow time	**16.25**	SPT - 16.75
Mean Tardiness	**5.25**	EDD - 5.5

Test Problem 3

The 3 x 3 test problem was described in the previous chapter. All the results of this test problem are explained below.

Results from Priority Rules

The test problem 3 is first solved by applying SPT rule. The best job schedule is represented through the Gantt chart shown in Figure 5-12. The minimum mean flow time obtained through SPT is **22.66 minutes**.

Figure 5-12 Gantt diagram for schedule from SPT rule

Similarly, the EDD rule is applied on the test problem 3 to find the best sequence that minimizes mean tardiness. The best job sequence obtained through EDD rule gives a value of **2 minutes**. Figure 5-13 shows the Gantt chart schedule for minimum tardiness.
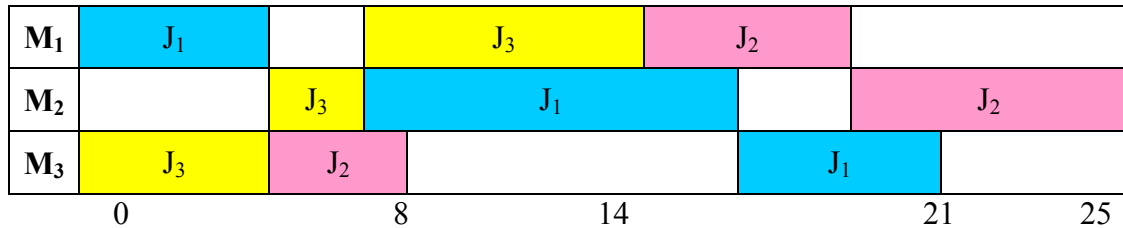
Figure 5-13 Gantt diagram for schedule from EDD rule

Results from AMOGA

AMOGA is used to solve the test problem 3. The test problem is solved in AMOGA for the weight combinations listed in Table 4-3. Table 5-6 shows the results obtained for each combination for 100 trials.

Table 5-6 Pareto results for Test Problem 3

Objective weight*		Best result	
Mean Flow time	Mean Tardiness	Mean Flow time (min)	Mean Tardiness (min)
0	1	21	2
0.2	0.8	21	2
0.4	0.6	21	2
0.6	0.4	21	2
0.8	0.2	21	2
1	0	21	2

The results for all the weight combinations remained the same for this problem. As we observe this is a small 3 by 3 problem where the best job sequence to minimize mean flow time and the sequence to minimize mean tardiness are the same. This is because of the smaller size of the problem.

68

The minimum mean flow time obtained by AMOGA from (1, 0) weight combination is 21 minutes. The best job sequence is given by jobs 3 – 1 – 2. Figure 5-14 shows the Gantt chart for the best schedule.

Figure 5-14 Gantt diagram for best mean flow time from AMOGA

Similarly, the minimum mean tardiness obtained by AMOGA from (0, 1) weight combination is 2 minutes. The best job sequence is given by jobs 3 – 1 – 2. In this problem the best schedule for both mean flow time and mean tardiness is same. The Gantt chart for the best schedule is similar to Gantt chart for best mean flow time schedule as shown in Figure 5-14.

Comparison of results

Since there are no prior results to compare multi-objective performance, we are therefore comparing them with respect to individual objectives. The performance of AMOGA is much better than SPT rule. However, the results are identical with EDD rule. Table 5-7 summarizes the results from both the methods.

Table 5-7 Comparison of Test problem 3 results

Optimization Criteria	AMOGA	Priority rules
Mean Flow time	**21**	SPT – 22.66
Mean Tardiness	**2**	EDD - 2

Therefore for all the three test problems of varying sizes the performance of AMOGA is much better than priority rules and at least as good as some formed through other multi-objective problems earlier. Hence, it can be reasonably considered that the AMOGA is effective in finding good solutions to the multi-objective optimization of minimizing flow time and tardiness in the job shop.

5.2 Experimentation for Adaptive Scheduling for Subsequent Days

The objective in this section is to perform adaptive scheduling. In the previous section the GA was validated. This section explains experimentations for adaptive multi-objective optimization. This section presents the results for all the experimentations conducted for the adaptive job shop problem. We tested on a single test problem for 3 successive days. The adaptability of the developed GA to the assigned priority was observed during these experimentations. All the results are summarized separately under each section.

5.2.1 First Day Results

The results for first day of the multiple day problem is provided in this section. The experiments are conducted using the best parameter set derived from previous experimentation.

Similar to the previous experimentation, each combination of weights with best parameter set is run for 100 trials and the best result is recorded. Tables 5-8 shows the best mean flow time and mean tardiness values obtained from each weight combination. Figure 5-15 shows the Pareto Front plot for the above results.

Table 5-8 Pareto Results for Modified FT06 day 1 Problem

Objective weight*		Best result	
Mean Flow time	Mean Tardiness	Mean Flow time	Mean Tardiness
0	1	973.33	3.33
0.2	0.8	866.67	16.67
0.4	0.6	866.67	16.67
0.6	0.4	866.67	16.67
0.8	0.2	866.67	16.67
1	0	866.67	16.67
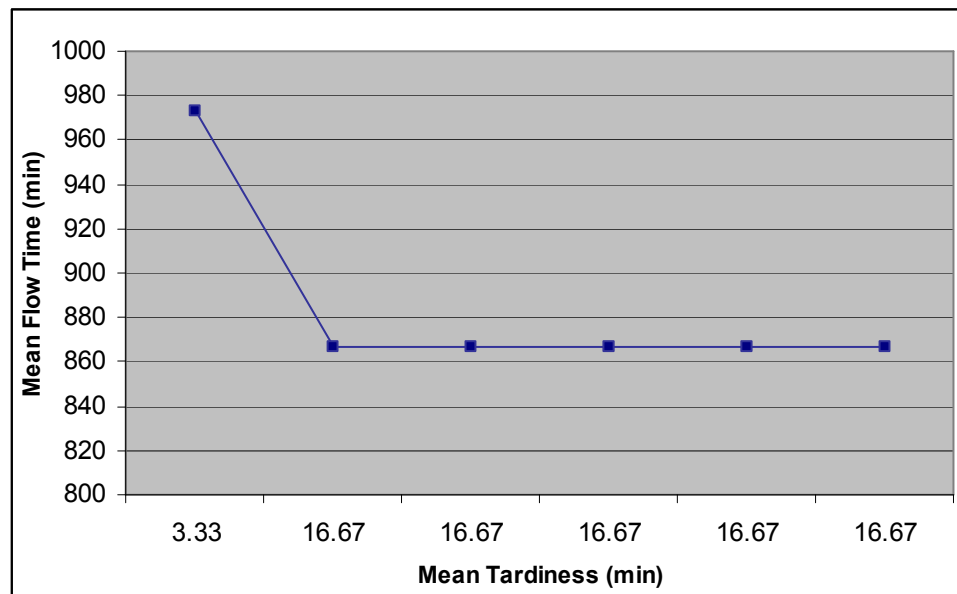
Figure 5-15 Pareto Results for Modified FT06 Problem – Adaptive Job Schedule

In order to obtain the best job schedule the weigh combination of $\alpha = 0.2$ and $\beta = 0.8$ was chosen for further experimentations.

The best sequence that minimizes both mean flow time and mean tardiness was selected. It was found that two job sequences slightly different from each other delivered the same values for the performance criteria. The first being 11 – 61 – 51 – 21 - 41 -31, while the

other was 61 -11 - 51 -21 - 41 -31. The Gantt chart for the first sequence is shown in Figure 5-16. The jobs are identified based on the job number and the day on which they enter the production plant. All the jobs of a particular day will have the day's index as suffix for proper identification.
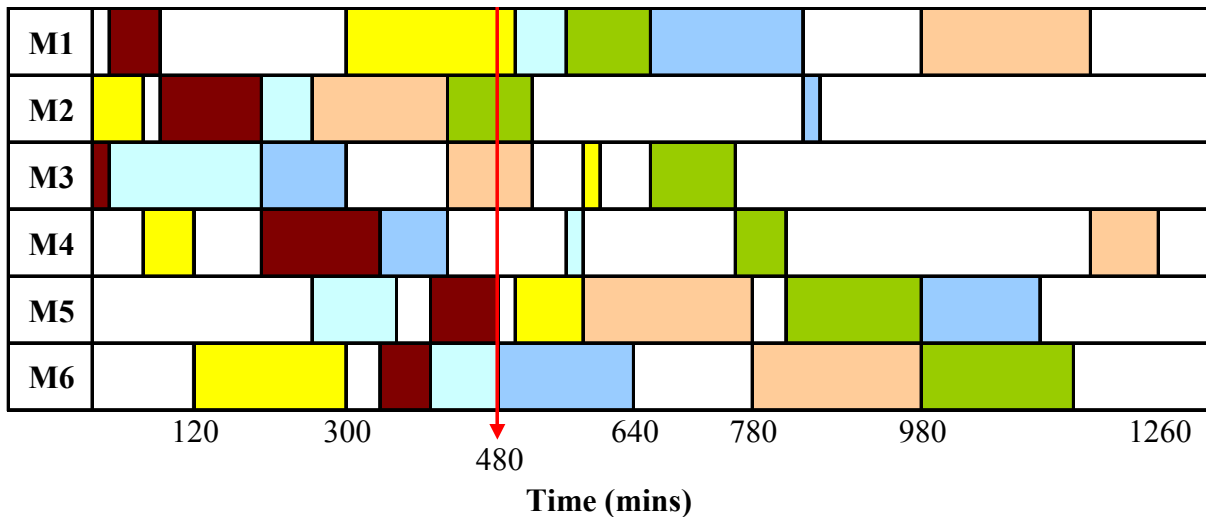
Figure 5-16 Gantt Chart for the Best Job Schedule- Modified FT06 Problem

End of Day 1

From the Figure 5-16 it can be clearly observed that only job 1 is finished at the end of the first day (8 hour run). The rest of the jobs 2, 3, 4, 5 and 6 still have some operations to be performed on the machines. All these jobs have to be carried forward to the next day for processing. Some simple rules are followed when carrying forward the jobs to the next day, as described earlier. Jobs carried forward and corresponding processing times at the end of the first day are shown in Table 5 – 9. Since these jobs have already been in

72

the system for a day, their due date is effectively reduced by one day in Table 5- 9. The text in bold indicates operations that are partially completed.

Table 5-9 Remaining Operations at end of first day

Jobs	Processing times (min)						Due date (Days)
11	0	0	0	0	0	0	1
21	0	**3(040)**	5(200)	6(200)	1(200)	4(080)	3
31	0	0	6(160)	1(180)	2(020)	5(140)	2
41	**2(040)**	1(100)	3(100)	4(060)	5(160)	6(180)	2
51	0	0	0	0	1(060)	4(020)	0
61	0	0	0	**1(020)**	5(080)	3(020)	1

The partially completed operations for J_{21}, J_{41} and J_{61} are kept unchanged when moving to the following days and are processed first on the corresponding machines. The remaining operations for these jobs are included in the AMOGA for scheduling.

5.2.2 Day 2 Results

For day 2 scheduling all the first day jobs with remaining (not yet started) operations are considered along with new jobs. The AMOGA determines the best schedule for all these jobs. Table 5-10 shows the data for jobs carried forward from the first day (only those operations that have not begin yet) and new jobs arriving the second day.

73

Table 5-10 Remaining Day 1 + New Day 2 Operations Scheduled by GA

	Jobs	Processing time (min)*						Due dates (Days)	Days In System (Days)
Carried Forward	21	0	0	5(200)	6(200)	1(200)	4(080)	3	1
	31	0	0	6(160)	1(180)	2(020)	5(140)	2	1
	41	0	1(100)	3(100)	4(060)	5(160)	6(180)	2	1
	51	0	0	0	0	1(060)	4(020)	0	1
	61	0	0	0	0	5(080)	3(020)	1	1
New	**72**	**1(020)**	**3(050)**	**4(050)**	**2(040)**	**5(030)**	**6(060)**	**1**	**0**
	82	**4(090)**	**3(070)**	**1(120)**	**5(250)**	**2(025)**	**6(080)**	**2**	**0**
	92	**5(090)**	**6(015)**	**3(030)**	**2(150)**	**4(025)**	**1(120)**	**3**	**0**

The experimentation on day 2 is performed in a manner similar to that of day 1. The same parameter set is used with all sets of weights listed in Table 4-3. The results for the minimum mean flow time and mean tardiness for 100 trials is shown in Table 5-11 and Figure 5-17. When calculating the mean flow times for the continuous day operations all the jobs that entered the system on the first day and were carried forward, 480 minutes added to their flow time.

Table 5-11 Pareto results for Day 2

Objective weight*		Best result	
Flow time	Tardiness	Mean Flow Time (min)	Mean Tardiness (min)
0	1	973.75	166.875
0.2	0.8	970	166.875
0.4	0.6	890.625	193.125
0.6	0.4	890.625	193.125
0.8	0.2	890.625	193.125
1	0	890.625	193.125
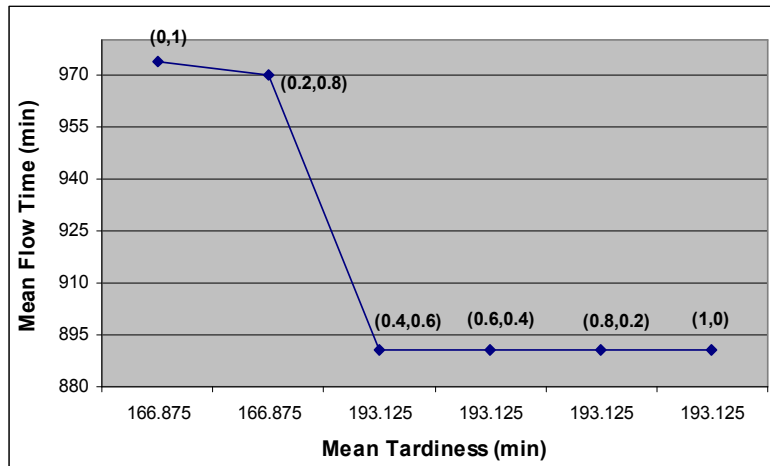
Figure 5-17 Pareto Chart for the Day 2 problem

From the Pareto chart it is noticed the best values for individual objectives and intermediate weights vary considerably. To proceed scheduling operations on the third day the best sequence obtained for objective weights of α = 0.2 and β = 0.8 were considered.

Figure 5-18 shows the Gantt chart for best sequence obtained through AMOGA. The job sequence obtained for this schedule is 61 – 21 – 72 – 41 – 51 - 31 - 82 – 92. The minimum mean flow time obtained for this best schedule is 970 minutes while the mean tardiness was 166.875 minutes.
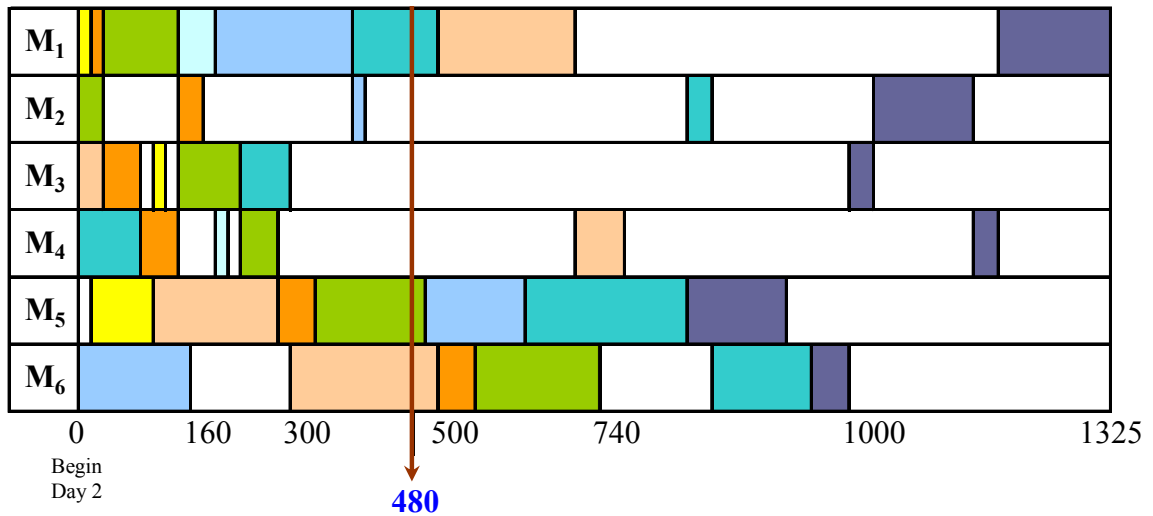
Figure 5-18 Gantt chart for Day 2 problem generated from AMOGA result

Legend							
	Job 61		Job 21		Job 41		Job 82
	Job 31		Job 51		Job 72		Job 92

End of Day 2

From the Figure 5-18 it can be observed that at the end of day 2 jobs 82, 41, 21, 72, 31 and 92 have partially completed operations. All these jobs must be carried forward to day 3 for processing. Table 5-12 shows the left over jobs and corresponding processing times at the end of day 2. Text in bold indicates the partially completed operations and remaining processing times.

Table 5-12 Remaining Operations at the end of Day 2

Jobs	Processing time (min)*						Due dates (Days)
82	0	0	**1(020)**	5(250)	2(025)	6(080)	2
41	0	0	0	0	**5(010)**	6(180)	2
21	0	0	0	**6(020)**	1(200)	4(080)	3
72	0	0	0	0	0	6(060)	1
31	0	0	0	0	0	5(140)	2
92	5(090)	6(015)	3(030)	2(150)	4(025)	1(120)	3

5.2.3 Day 3 Results

For the third day of scheduling all the data for carried forward jobs and new jobs are considered. Therefore, the AMOGA determines the best schedule for all these jobs. Table 5-13 shows data for carried forward jobs with remaining operations (not partially completed) and entering on day 3.

Table 5-13 Remaining Day 2 + New Day 3 Operations Scheduled by GA

Jobs	Processing time (min)*						Due dates (Days)	Days In System (Days)
82	0	0	0	5(250)	2(025)	6(080)	1	1
41	0	0	0	0	0	6(180)	1	2
21	0	0	0	0	1(200)	4(080)	2	2
72	0	0	0	0	0	6(060)	0	1
31	0	0	0	0	0	5(140)	1	2
92	5(090)	6(015)	3(030)	2(150)	4(025)	1(120)	2	1
13	**3(010)**	**2(030)**	**6(040)**	**1(070)**	**4(060)**	**5(050)**	**1**	**0**
53	**2(100)**	**1(080)**	**5(020)**	**3(040)**	**4(050)**	**6(190)**	**2**	**0**

77

Partially completed operations are left on the machine unchanged and the sequence for job shown in Table 5-13 are determined through the AMOGA. The GA parameters used previously are kept the same. The results for the minimum values of mean flow time and mean tardiness after 100 trials are shown in Table 5-14 and Figure 5-19.

Table 5-14 Pareto results for Day 3 Problem

Objective Weight*		Best result	
Flow time	Tardiness	Mean Flow Time (min)	Mean Tardiness (min)
0	1	960.625	375.625
0.2	0.8	960.625	375.625
0.4	0.6	960.625	375.625
0.6	0.4	960.625	375.625
0.8	0.2	951.25	396.25
1	0	948.125	397.5
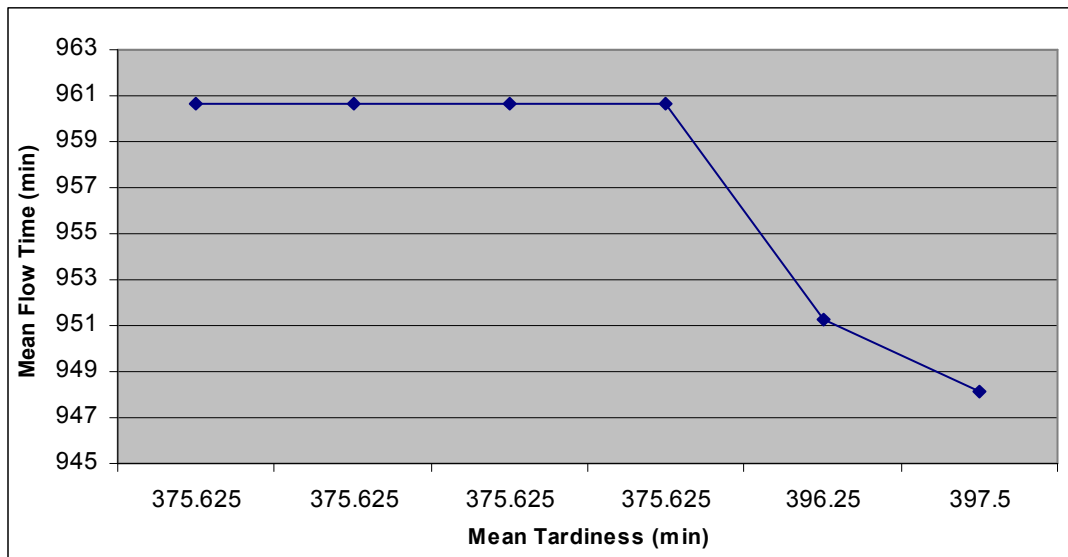
Figure 5-19 Pareto chart for Day 3 Problem

Figure 5-20 shows the Gantt chart for best sequence obtained through AMOGA for $\alpha = 0.2, \beta = 0.8$. The job sequence obtained for this schedule is $72 - 31 - 21 - 41 - 92 - 82 - 13 - 53$. The minimum mean flow time obtained for this schedule is 960.625 minutes and the mean tardiness is 375.625 minutes.
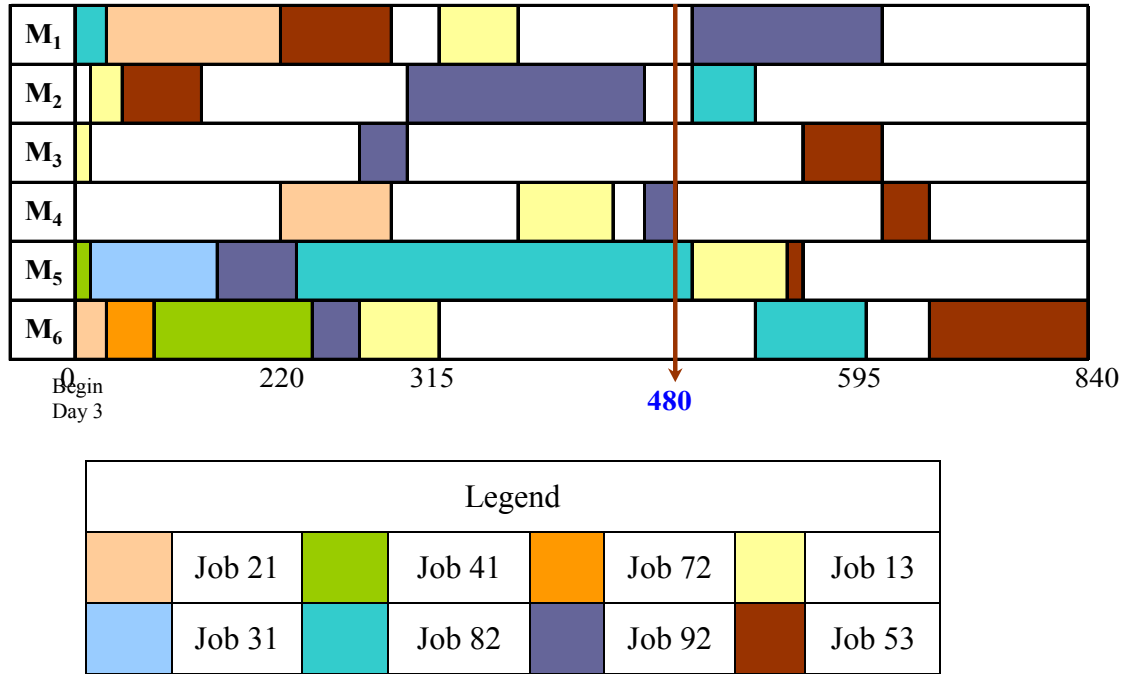
Figure 5-20 Gantt Chart for day 3 problem generated from AMOGA schedule

5.2.4   Summary of Results – Adaptive Job Shop Scheduling

From the above Gantt schedules we can observe the results generated by the AMOGA. To describe the adaptive scheduling the three days best schedule are considered into a single chart. All the jobs that are not completed on day 1 are carried forward to day 2 and similarly to day 3. Figure 5-22 shows the Gantt chart for the continuous job shop scheduling problem which is based on the integration of Gantt charts for the individual days. The adaptive scheduling over the three days super-imposed over the physical layout is shown in Figure 5-21.
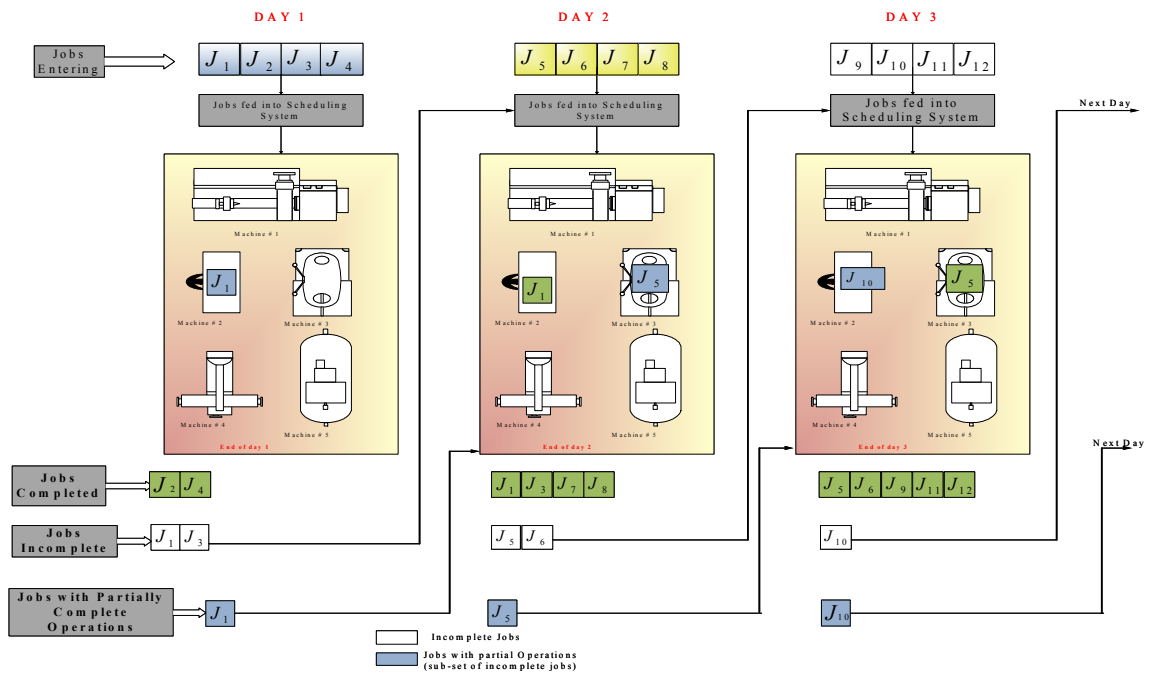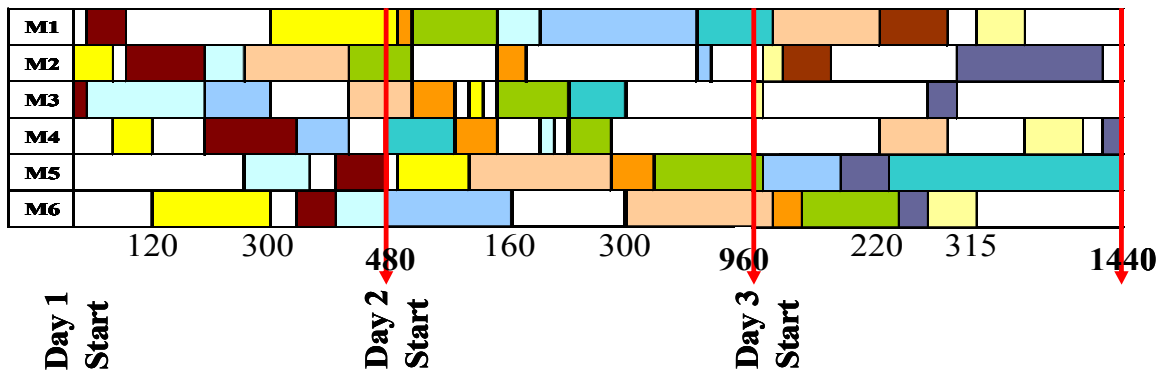
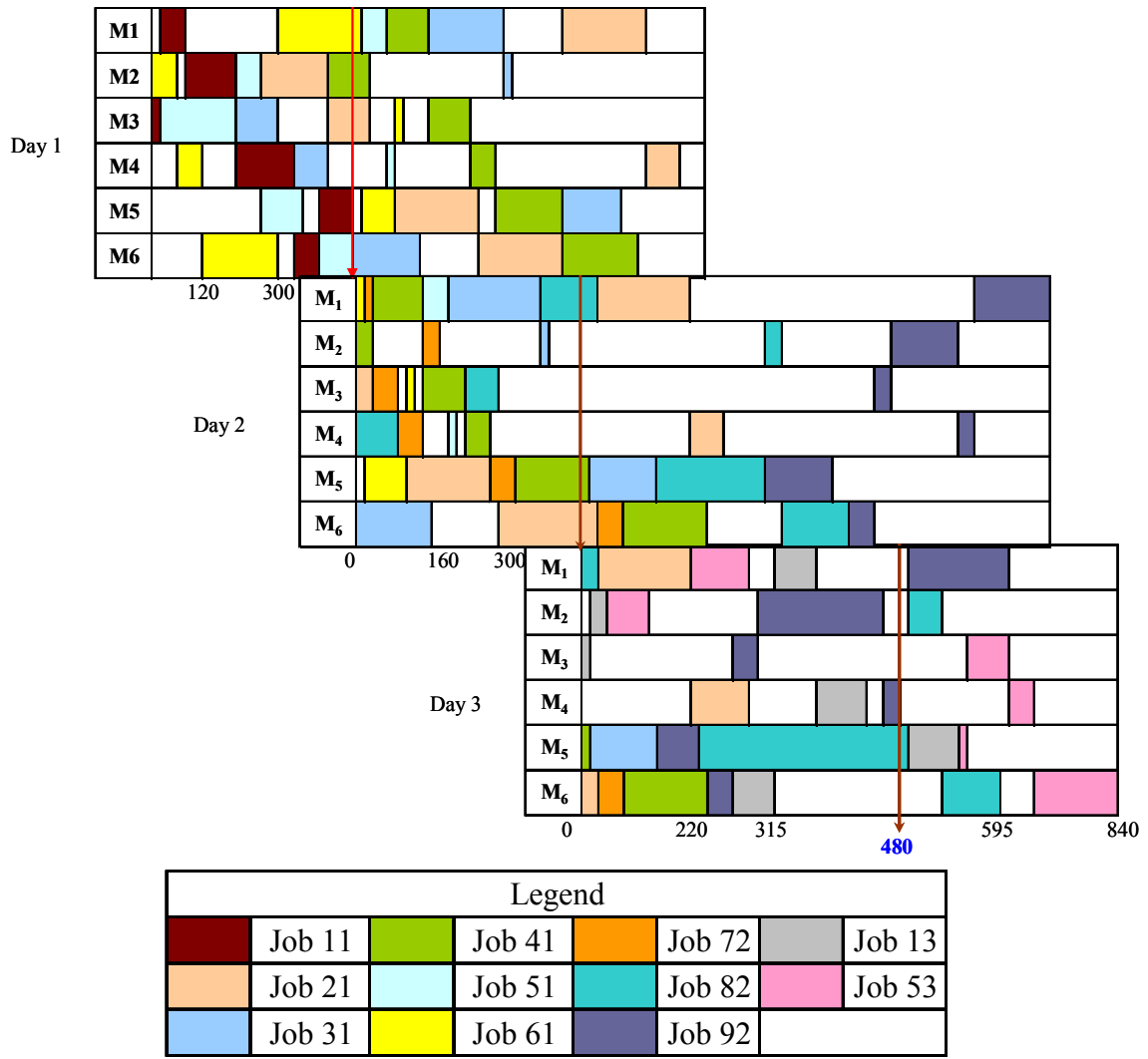**Figure 5-21 Gantt Chart for Adaptive Job Shop Problem - 3 days continuous schedule**

Figure 5-22 A Summary of the Adaptive Job Shop problem

As we observe at the end of first day job 11 is finished. As a higher priority is given to minimizing the mean tardiness, jobs with shorter due dates must be completed first. If only tardiness is prioritized job 51 must be processed first. But, the total processing time for this job is greater than 480. Hence among the shorter due date jobs, only job 11 can be completed on day 1. Therefore the AMOGA is following priorities while scheduling. At the end of day 2, we observe that although jobs 21, 31 and 41 are in the system from day 1, jobs with shorter due dates (such as jobs 51 and 61) are finished first. Also since the job 71 coming on day 2 have a shorter due date, the AMOGA finished most of the operations for that job. Hence the AMOGA is delaying jobs with later due dates to finish

jobs coming in on following days with shorter due dates. This adaptation is effectively happening in the AMOGA. However, if mean flow time minimization was given a higher priority, the AMOGA should be releasing jobs to minimize the time spent in system. The figure compares the problem framework described in the methodology to the AMOGA's best job schedule obtained in this research.

6 CONCLUSIONS AND FUTURE RESEARCH

The experimentations, results and discussions for the tested problems are presented in the chapter 4 and 5. This chapter provides the conclusions and future research options of this research.

In this research, adaptive scheduling problem in a real time job shop environment was solved, for the multi-objective optimization of minimizing mean flow time and mean tardiness. The objectives were chosen because minimizing mean flow time minimizes the manufacturing lead time. On the other hand minimizing mean tardiness helps to meet the delivery dates effectively. An asexual reproduction genetic algorithm with two mutation strategies was used to solve the adaptive scheduling problem. Adaptive scheduling was considered because in real life manufacturing environments scheduling based on given priorities is very important to achieve desired objectives.

In order to evaluate the effectiveness of the AMOGA developed to solve the adaptive job shop scheduling problem the effectiveness of the model was first tested using a single day dates. For the single day, extensive analyses were conducted on the FT06 benchmark problem and several other problems. The experimentations with these problems confirmed that the AMOGA is able to find good solutions to the problem addressed. Though previous results were not available to evaluate the weighed objectives, the results found were better or comparable to those formed in literature and by applying dispatching rules. The GA parameters were varied to determine the best set that produced good solutions more frequently.

The multiple-day continuous job shop problem was tested in a similar manner. However, to schedule the previous day's remaining jobs on the following day we followed certain scheduling rules. All the jobs whose last operation is partially completed were scheduled first. For the jobs with partially completed operation that also have other operations to be processed on one or more other machines, the partially completed were unaltered and

scheduled first on the following day, the remaining operations were scheduled along with new jobs using GA. All the jobs with no partially completed operations were scheduled along with new jobs.

Overall the results indicate that the AMOGA developed for performance of the adaptive scheduling of jobs in a job shop environment is able to generate comparable or better results to that formed in literature. The results show considerable adaptability to the Mean tardiness objective that was considered in this research. Currently AMOGA generated best schedule for each day. But scheduling could also be performed for desired time period (say weekly basis) by expanding the time horizon. Also the Figure 5-22 showing the job schedules for individual days considers a real time situation where jobs are already being processed on a system and scheduling starts from a certain day not necessary from day 1.

6.1   Unique Features

Adaptive job scheduling

In this research the adaptive scheduling problem, where incomplete jobs from one day are passed on to the following day to be scheduled along with new jobs was considered. Adaptive job scheduling enables minimizing the disruption in the production floor, but meets the desired objectives.

Ability to display machine idle times

The current job shop scheduler has the capability to generate the machine idle times for each schedule. This information can help the scheduler in managing a variety of operations. A display of machine idle times can help in managing the dynamic, unpredictable environments such as

- Predictive maintenance scheduling
- Scheduling unexpected, immediate delivery new jobs
- Manage worker idle times and break times

Asexual reproduction and multiple mutation strategies

Asexual reproduction is not as widely used, but in this case combining two mutation strategies have been effective in finding good solutions. Moreover, the solutions are always legal, thereby producing a significant reduction in computational time.

6.2   Future Work

The software program developed in this research has the ability to accommodate minor schedule disruptions. However, for greater extent of accommodation further details are to be incorporated in the problem with special features.

The jobs and processing times are manually input to the scheduler. This is time consuming and likely to cause errors particularly when scheduling larger problems. This can be upgraded by modifying the software to capture data from any data files available on the computer. This can considerably reduce the time consumed in entering the job details.

The results generated are currently in a text format. To improve this display interface the results can be delivered through a Gantt chart, which could help in visual interpretation of data.

The AMOGA was not tested with very large size problems. However, the scheduler can be upgraded to solve large sized problems and further trends in results could be observed by extending the GA parameters to include including new strategies, Niching and seeding concepts.

A comparison of results for same problem with sexual reproduction in terms of quality of solutions and computational time could be performed.

APPENDIX I

AMOGA Software Interface

Day 2 Test Problem

Modified FT06 Day 1 Test Problem:

The following Figure shows the processing times and due dates interface for the modified FT06 test problem for the first day.

The following Figure shows the processing times and due dates interface for the modified FT06 test problem on day 2.

Job Information Sheet — Job Information

Number of Jobs partially completed on their last operation	0
Number of jobs with Partial Operation (< than or equal to jobs from previous day)	3

New Jobs	3
Carried forward Jobs (from earlier excluding jobs which are partially completed on their last operation)	5
Total Jobs	8
Total Number of Machines	6
Execution Start and End Times	8:55:3 8:56:0

Job	Machine	Processing Time	Due Date	Days in System
561	1	20	1	1
341	2	40	2	1
121	3	40	3	1

PROCEED TO ENTER PROCESSING TIMES

JOBS	DUE DATES	OPERATION1	OPERATION2	OPERATION3	OPERATION4	OPERATION5	OPERATION6	DAYS IN SYSTEM
121	3	5(200)	6(200)	1(200)	4(080)	2(000)	3(000)	1
231	2	6(160)	1(180)	2(020)	5(140)	3(000)	4(000)	1
341	2	1(100)	3(100)	4(060)	5(160)	6(180)	2(000)	1
451	0	1(060)	4(020)	2(000)	3(000)	5(000)	6(000)	1
561	1	5(080)	3(020)	1(000)	2(000)	4(000)	6(000)	1
672	1	1(020)	3(050)	4(050)	2(040)	5(030)	6(060)	0
782	2	4(090)	3(070)	1(120)	5(250)	2(025)	6(080)	0
892	3	5(090)	6(015)	3(030)	2(150)	4(025)	1(120)	0

OK   CANCEL   RUN GA

Day 3 Continued Operations:

The following Figure shows the processing times and due dates interface for the modified FT06 test problem on day 3.

REFERENCES

1) Aldakhilallah, K. A., and Ramesh, R., (2001), "'Cyclic scheduling heuristics for a re-entrant job shop manufacturing environment", *International Journal of Production Research*, Vol. 39, Issue 12, pp. 2635 – 2657.

2) Anderson, E. J., and Nyirenda, J. C., (1990), "Two new rules to minimize tardiness in a job shop", *International Journal of Production Research*, Vol. 28, Issue 12, pp. 2277-2292.

3) Asano, M., and Ohta, H., (2002), "A Heuristic approach for Job shop scheduling to minimize total weighed tardiness", *Journal of Computers and Industrial Engineering,* Vol. 42, pp. 137-147.

4) Balas, E., (1998), "Optimized Crossover-Based Genetic Algorithms for the Maximum Cardinality and Maximum Weight Clique Problems", Journal *of Heuristics,* Vol. 4, pp.107-122.

5) Banzhaf, W., Nordin, P., Keller, R. E., Francone, F. D. (1998), <u>Genetic Programming - An Introduction - On the Automatic Evolution of Computer Programs and its Applications</u>, Morgan Kaufmann, San Francisco, CA.

6) Barman, S., (1997), "Simple priority rules combinations: an approach to improve both the flow time and tardiness", *International Journal of Production Research*, Vol. 35, Issue 10, pp. 2857–2870.

7) Barnes, J. W., and Chambers, J. B., (1995), "Solving the Job Shop Scheduling Problem Using Tabu Search", *IIE Transactions*, Vol. 27, pp. 257-263.

8) Barnes, J. W., and Laguna, M., (1993), "A Tabu Search Experience in Production Scheduling", *Annals of Operations Research*, Vol. 41, pp. 141-156.

9) Baykasoglu, A., Gocken, M., and Unutmaz, Z.D., (2008), "New approaches to due date assignment in job shops", *European Journal of Operational Research*, Vol. 187, Issue 1, pp. 31-45.

10) Baykasoglu, A., Ozbakir, L., and Dereli, T., (2002), "Multiple Dispatching Rule Based Heuristic for Multi-Objective Scheduling Of Job Shops Using Tabu Search", Proceedings of MIM 2002: 5th International Conference on Managing Innovations in Manufacturing, Milwaukee, WC, September 9-11, 2002.

11) Belton, V., and Elder, M.D., (1996), "Exploring a Multicriteria Approach to Production Scheduling", *Journal of the Operational Research Society,* Vol. 47, No. 1, pp.162-174.

12) Biskup, D., and Piewitt, W., (2000), "A note on an efficient algorithm for the single machine tardiness problem", *International Journal of Production Economics*, Vol. 66, pp. 287–292.

13) Blazewicz, J., Domschke, W., and Pesch, E., (1996), "The job shop scheduling problem: conventional and new solution techniques", *European Journal of Operational Research*, Vol. 93, pp. 01-33.

14) Blazewicz, J., Ecker, K.H., Pesch, E., Smith, G., and Weglarz, J., (2001), "<u>Scheduling Computer and Manufacturing Processes</u>", Springer, New York.

15) Blum, C., and Roli, A., (2003), "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", *ACM Computing Surveys*, Vol. 35, No. 3, pp. 268–308.

16) Bontridder, K. M., (2005), "Minimizing Total Weighted Tardiness in a Generalized Job Shop", *Journal of Scheduling*, Vol.8, Issue 6.

17) Boyd, E. A., and Burlingame, R., (1996), "A Parallel Algorithm for Solving Difficult Job-Shop Scheduling Problems", Operations Research Working Paper, Department of Industrial Engineering, Texas A&M University, College Station, Texas, USA.

18) Braun, H., (1993), "On Solving Traveling Salesman Problems by Genetic Algorithms", Parallel Problem Solving from Nature, Springler-Verlag, Berlin.

19) Brooks, G. H., and White, C. R., (1965), "An Algorithm for Finding Optimal or Near Optimal Solutions to the Production Scheduling Problem", *Journal Of Industrial Engineering*, Vol. 16, Issue 1, pp. 34-40.

20) Brucker, P., Jurisch, B., and Sievers, B., (1994), "A Branch and Bound Algorithm for the Job-Shop Scheduling Problem", *Discrete Applied Mathematics*, Vol. 49, pp. 109-127.

21) Calabrese, J., Henley, R., and Udayabhanu, V., (2001), "Benchmarking of dispatching rules for the job shop flow time problem", Proceedings of First International Conference on Logistics and Supply Chain Management, PSG College of Technology, India, pp.205–209.

22) Carlier, J., and Pinson, E., (1989), "An Algorithm for Solving the Job Shop Problem" *Management Science*, Vol. 35, Issue 2, pp. 164-176.

23) Chakroborty, P., and Mandal, A., (2005), "An asexual genetic algorithm for the general single vehicle routing problem", *Engineering Optimization,* Vol. 37, No. 1, pp. 1–27

24) Chambers, J.B., and Barnes, J.W., (1996), "New tabu search results for the job shop scheduling problem", Technical Report , Graduate Program in Operations Research and Industrial Engineering, University of Texas at Austin, Austin, USA.

25) Chatterjee, S., Carrera, C., and Lynch, L.A., (1996), "Genetic algorithms and traveling salesman problems", *European Journal of Operational Research*, Vol. 93, Issue 3, pp. 490-510

26) Cheng, R., Gen, M., and Tsujimura, Y., (1999), "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies", *Journal of Computers and Industrial Engineering*, Vol. 36, Issue 2, pp. 343-364.

27) Chu, C., Portmann, M. C., and Proth, J. M., (1992),"A Splitting-Up Approach to Simplify Job-Shop Scheduling Problems", *International Journal of Production Research,* Vol. 30, Issue 4, pp. 859-870.

28) Cicirello, V., and Smith, S., (2001), "Randomizing Dispatch Scheduling Policies", Proceedings of the 2001 AAAI Fall Symposium: Using Uncertainty Within Computation, November, 2001.

29) Coello, C. A., (2000), "An updated survey of GA-based multiobjective optimization techniques", *Journal of Association for Computing Machinery Computer Surveys,* Vol. 32, Issue 2, pp. 109-143.

30) Cohon, J. L., Marks, D. H, (1975),"A review and evaluation of multiobjective programming techniques", *Water Resources Research*, Vol. 11, Issue 2, pp. 208–219.

31) Corcoran, A. L., and Wainwright, R. L., (1994), "A parallel island model genetic algorithm for the multiprocessor scheduling problem", In Proceedings of the 1994 ACM Symposium on Applied Computing , Phoenix, Arizona, United States, March 06 - 08, 1994.

32) Corne, D. W., and Knowles, J. D., (2007), "Techniques for highly multiobjective optimisation: some nondominated points are better than others", Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England, July 07 - 11, 2007.

33) Cornforth, D.J., (2007), "An investigation into dynamic problem solving in a hybrid evolutionary market-based multi-agent system", Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 2007.

34) Davis, L., (1985), "Job-Shop Scheduling with Genetic Algorithm", Proceedings of the 1st International Conference on Genetic Algorithms and their Applications, Pittsburgh, PA, USA.

35) Della Croce, F., Tadei, R., and Volta, G., (1995), "A Genetic Algorithm for the Job Shop Problem", *Computers and Operations Research*, Vol. 22, Issue 1, pp. 15-24.

36) Dimopoulos, C., and Zalzala, A.M.S., (2000), "Recent developments in evolutionary computation for manufacturing optimization: Problems, solutions and comparisons", *IEEE Transactions on Evolutionary Computation*, Vol. 4, Issue 2, pp.93-113.

37) Esquivel, S.C., Ferrero, S.W., and Gallard, R.H., (2002), "Parameter Settings And Representations In Pareto-Based Optimization For Job Shop Scheduling", *Cybernetics And Systems*, Vol. 33, pp. 559-578.

38) Falkenauer, E., and Bouffouix, S., (1991)," A Genetic Algorithm for the Job-Shop", Proceedings of the IEEE International Conference on Robotics and Automation, Sacramento, California, USA, pp. 824-829.

39) Fisher, M. L., (1973a), "Optimal Solution of Scheduling Problems using Lagrange Multipliers: Part I", *Journal of Operations Research*, Vol. 21, pp. 1114-1127.

40) Fisher, M. L., (1973b), "Optimal Solution of Scheduling Problems using Lagrange Multipliers: Part II", Symposium on the Theory of Scheduling and its Applications, Springer, Berlin, Germany.

41) Fonseca, C. M., and Fleming, P. J., (1993), "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," Proceedings of the 5th International Conference of Genetic Algorithms, Urbana-Champaign, IL, USA, June 1993, pp. 416–423.

42) Franklin, C. L., (1969), "The current state of research in job shop scheduling", Proceedings of the Third Conference on Applications of Simulation, Los Angeles, CA, December 08 - 10, 1969, pp. 373-381.

43) French, S., (1982), "Sequencing and Scheduling - An Introduction to the Mathematics of the Job-Shop", John-Wiley and Sons, New York.

44) Furuhashi, T, Nakaoka, K. and Uchikawa, Y., (1994), "A New Approach to Genetic Based Machine Learning and an Efficient Finding of Fuzzy Rule", Proceedings of the 1994 IEEE/Nagoya-University World Wisepersons Workshop. LNAI, Vol. 1011, pp. 173-189, Springer- Verlag.

45) Garrido, A., Salido, M.A., Barber, F., and Lopez, M.A., (), "Heuristic Methods for Solving Job Shop Scheduling Problems",

46) Gen, M. and Cheng, R., (1999), "Genetic Algorithms and Engineering Optimization", John Wiley and Sons, New York.

47) Giffler, B., and Thompson, G.L., (1960), "Algorithms for solving production scheduling problems", *International Journal of Operations Research,* Vol. 8, pp. 487–503.

48) Glover, F., (1986), "Future Paths for Integer Programming and Links to Artificial Intelligence", *Journal of Computer and Operations Research*, Vol. 13, No. 5, pp. 533-549.

49) Glover, F., and Greenberg, H. J., (1989), "New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence", *European Journal of Operations Research*, Vol. 39, Issue 2, pp. 119-130.

50) Glover, F., and Kochenberger, G.A., (2003), "Handbook of Metaheuristics", Kluwer Publishers, Hingham, Massachusetts.

51) Gordon, V., Proth, J., and Chu, C., (2002),"A Survey of the State-of-the-Art of Common Due date Assignment and Scheduling Research", *European Journal of Operational Research*, vol.127, pp. 1-25.

52) Greenberg, H., (1968), "A Branch and Bound Solution to the General Scheduling Problem", *Journal of Operations Research*, Vol. 16, Issue 2, pp. 353-361.

53) Gupta, A.K., and Sivakumar, A.I., (2002), "Simulation based Multiobjective schedule Optimization in Semiconductor Manufacturing", In Proceedings of the Winter Simulation Conference 2002, San Diego, CA, December 8-11, 2002, pp. 1862-1870.

54) Hara, H., (1995), "Job-Shop Scheduling by Minimal Conflict Search", *Japanese Artificial Intelligence Society*, Vol. 10, Issue 1, pp. 88-93.

55) Hart, E., Ross, P., and Corne, D., (2005), "Evolutionary Scheduling: A Review", *Genetic Programming and Evolvable Machines*, Vol. 6, pp. 191-220.

56) Harvey, I., (1992), "The Microbial Genetic Algorithm", *Journal of Evolutionary Computation*, MIT Press 1996.

57) He, Z., Yang, T., and Deal, D.E., (1993), "A multiple-pass heuristic rule for job shop scheduling with due dates", *International Journal of Production Research*, Vol. 31, pp. 2677–2692.

58) Herroelen, W., De Reyck, B., and Demeulemeester, E., (1998), "Resource-Constrained Project Scheduling: A Survey of Recent developments", *Journal of Computers and Operations Research,* Vol. 25, No. 4, pp. 279-302.

59) Holland, J. H., (1975), Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI.

60) Holland, J. H., (1992), Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control and artificial intelligence, MIT Press 1992.

61) Holloway, C.A., Nelson, R. T., (1974) "Job shop scheduling with due dates and variable processing times", *Management Science*, Vol. 20, Issue 9.

62) Hoogeveen, H., "Multicriteria scheduling", *European Journal of Operational Research,* Vol.167, pp. 592–623.

63) Ignall, E., and Schrage, L., (1965), "Application of the Branch and Bound Technique to some Flow-Shop Scheduling Problems", Journal of *Operations Research*, Vol. 13, pp. 400-412.

64) Iima, H., Hara, T., Ichimi, N., and Sannomiya, N., (1999), "Autonomous Decentralized Scheduling Algorithm For A Job-Shop Scheduling Problem With Complicated Constraints", Proceedings Of The 1999 4th International Symposium On Autonomous Decentralized Systems, Tokyo, pp. 366-369.

65) Itoh, K., Huang, D., and Enkawa, T., (1993), "Twofold look-ahead search for multi-criterion job shop scheduling*", International Journal of Production Research*, Vol. 31, Issue 9, No. 9, pp. 2215–2234.

66) Jain, A. S., and Meeran, S., (1998), "A state-of-the-art review of job-shop scheduling techniques", Technical Report, Department of Applied Physics, Electronics and mechanical Engineering, University of Dundee, Scotland.

67) Jones, A., and Rabelo, L.C., (1998), "Survey of Job Shop Scheduling Techniques", PhD Dissertation, National Institute of Standards and Technology, Gaithersburg, MD, USA.

68) Jurisch, B., (1992), "Scheduling jobs in shops with multi-purpose machines", PhD thesis, University of Osnabrük, Germany.

69) Kaschel, J., Teich, T., and Kobernik, G. , and Meier, B., (1999), "Algorithms for the Job Shop Scheduling Problem − A Comparison of Different Methods", Technical University Chemnitz, Germany.

70) Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., (1983), "Optimization by Simulated Annealing", *Journal of Science*, Vol. 220, No. 4598, pp. 671 - 680.

71) Kruger, K., Shakhlevich, N. V., Sotskov, Y. N., and Werner, F., (1995), "A Heuristic Decomposition Algorithm for Scheduling Problems on Mixed Graphs", *Journal of the Operational Research Society*, Vol. 46, pp. 1481- 1497.

72) Kubiak, W., Lou, S.X.C., and Wang, Y., (1996), "Mean Flow Time Minimization in Reentrant Job Shops with a Hub", *Journal of Operations Research*, Vol. 44, No. 5. pp. 764-776.

73) Kuluc, S., and Kahraman, C., (2006), "Metaheuristic Techniques for Job Shop Scheduling Problem and a Fuzzy Ant Colony optimization Algorithm", Fuzzy Applications in Industrial Engineering, Springer, Berlin, Germany, pp. 401-435.

74) Kutanoglu, E., and Sabuncuoglu, I., (1999), "An Analysis of Heuristics in A Dynamic Job Shop With Weighted Tardiness Objectives", *International Journal of Production Research,* Vol. 37, No. 1, pp. 165 - 187.

75) Laarhoven, P.J.M.V., and Aarts, E.H.L., (1987), "Simulated Annealing; Theory and Practice", Kluwer Publishers, Hingham, Massachusetts.

76) Laguna, M., Barnes, J. W., and Glover, F. W., (1993), "Intelligent Scheduling with Tabu Search: An Application to Jobs with Linear Delay Penalties and Sequence-Dependent Setup Costs and Times", *Journal of Applied Intelligence*, Vol. 3, pp. 159-172.

77) Lin, S.C., Goodman, E.D., and Punch, W.F., (1997), "A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems", Proceedings of the 7th International Conference on Genetic Algorithm, San Francisco, CA, USA.

78) Liu, B., Hsu, W., and Chen, S., (1997),"Using general impressions to analyze discovered classification rules" Proceedings of 3rd International Conference on Knowledge Discovery & Data Mining, pp. 31-36.

79) Lorenco, H. R., (1995), "Job shop scheduling computational study of local search and optimization methods", *European Journal of Operational Research*, Vol. 83, pp. 347-364.

80) Lourenco, H.R., (1995), "Job-shop scheduling: Computational study of local search and large-step optimization methods", *European Journal of Operational Research*, Vol. 83, Issue 2, pp. 347-364.

81) Madureira, A., Ramos, C., and Silva, S.C., (2001), "A Genetic Approach for Dynamic Job-Shop Scheduling Problems", 4th Metaheuristics International Conference 2001, Porto, Portugal, July 16-20.

82) Manne, A. S., (1960),"On the Job-Shop Scheduling Problem", *Journal of Operations Research*, Vol. 8, pp. 219-223.

83) Masud, A.S.M., and Hwang, C.L., (1980), "An aggregate production planning model and application of three multiple objective decision methods", *International Journal of Production Research,* Vol. **18**, pp. 741–752.

84) Mattfeld, D.C., and Bierwirth, C., (2004), "An efficient genetic algorithm for job shop scheduling with tardiness objectives", *European Journal of Operational Research*, Vol. 127,Issue 3, pp. 616-630.

85) McMahon, G. B.,and Florian, M., (1975), "On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness", *Operations Research*, Vol. 23, Issue 3, pp. 475-482.

86) Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., (1953), "Equation of State Calculations by Fast Computing Machines", *The Journal of Chemical Physics*, Vol. 21, Issue 6, pp. 1087-1092.

87) Mitchell, M., (1996), An Introduction to Genetic Algorithms, MIT Press 1996.

88) Mitchell, M., (1998), An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA.

89) Mitchell, M., Forrest, S., (1994), "Genetic Algorithms and Artificial Life. Journal of Artificial Life", Vol. 3, pp. 267-289.

90) Mitchell, M., Holland, J. H., Forrest, S., (1994), Advances in Neural Information Processing Systems, Morgan Kaufmann Publishers, San Fransisco, CA, pp. 51-58.

91) Moon, I., and Lee, J., (2000), "Genetic Algorithm Application to the Job Shop Scheduling Problem with Alternative Routings", Technical report, Brain Korea 21 Logistics Team, Pusan National University, South Korea.

92) Nakano, R., and Yamada, T., (1991), "Conventional Genetic Algorithm for Job-Shop Problems", Proceedings of the 4th International Conference on Genetic Algorithms and their Applications, San Diego, USA, pp. 474-479.

93) Nasr, N., and. Elsayed, E. A.,(1990), "Job Shop Scheduling with Alternative Machines", *International Journal of Production Research*, Vol. 28, No. 9, pp. 1595-1609.

94) Nawa, N., Furuhashi, T., (1998), "Bacterial Evolutionary for Fuzzy System Design", Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics. IEEE Press 1998.

95) Nawa, N., Furuhashi, T., Hashiyama, T. and Uchikawa, Y., (1999), "A Study of the Discovery of Relevant Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm" IEEE Transactions on Industrial Electronics.

96) Nawa, N., Hashiyama, T. , Furuhashi, T. and Uchikawa, Y., (1997), "A Study on Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm with Adaptive Operator", Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 589-593. IEEE Press.

97) Nowicki, E., and Smutnicki, C., (1996), "A Fast Taboo Search Algorithm for the Job-Shop Problem", *Management Science*, Vol. 42, Issue 6, pp. 797-813.

98) Oates, M., Corne, D., Loader, R., (1999), "Skewed Crossover and the Dynamic Distributed Database Problem. In the Proceedings of the Fourth Int. Conference on Neural Networks and Genetic Algorithms, pp. 280-287, Springer- Verlag 1999.

99) Osyczka, Andrzej., (1985), "Multicriteria optimization for engineering design" Design Optimization, Academic Press, pp. 193-227.

100) Peres, S.D., and Paulli, J., (1997),"An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using Tabu search", *Annals of Operations Research*, Vol. 70*, pp. 281–306.

101) Pinedo, M., (2001), "Scheduling: Theory, Algorithms, and Systems", Prentice Hall, New York.

102) Pinedo, M., and Singer, M., (1999), "A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop" *Naval Research Logistics*, Vol, 46, Issue 1, pp. 1–17.

103) Ponnambalam, S. G., Ramkumar, V., and Jawahar, N., (2001), "A multiobjective genetic algorithm for job shop scheduling", *Journal of Production Planning and Control*, Vol. 12, No. 8, pp. 764-774.

104) Raheja, A.S., and Subramaniam, V., (2002), "Reactive recovery of job shop schedules--a review", *International Journal of Advanced Manufacturing Technology*, Vol. 19, Issue 10, pp. 756-763.

105) Raheja, A.S., and Subramaniam, V., (2003), "Reactive Schedule Repairing of Job-Shops", Technical Report, Innovation in Manufacturing Systems and Technology, National University of Singapore, MY.

106) Reeja, M. K., and Rajendran, C., (2000), "Dispatching rules for scheduling in assembly job shops – Part 1", *International Journal of Production Research*, Vol. 38, Issue 9, pp. 2051–2066.

107) Rosenberg, R.S., (1967), "Simulation of genetic populations with biochemical properties", PhD Dissertation, University of Michigan, Ann Arbor, MI, USA.

108) Sadeh, N., (1991), "Look-Ahead Techniques for Micro-Opportunistic Job Shop Scheduling", PhD Dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

109) Sadeh, N., and Nakakuki, Y., (1996), "Focused Simulated Annealing Search: An Application to Job Shop Scheduling", *Annals of Operations Research*, Vol. 60, pp. 77-103

110) Satake, T., Morikawa, K., Takaahashi, K., and Nakamura, N., (1999), "Simulated annealing approach for the minimizing the make span of the general job shop", *International Journal of Production Economics*, Vol. 60, Issue 61, pp. 515–522.

111) Schaffer, J. D., (1984), "Some experiments in machine learning using vector evaluated genetic algorithms*"*, PhD Dissertation, Vanderbilt University: Nashville, TN, USA.

112) Scrich C.R., Armentano V.A., Laguna, M., (2004), "Tardiness minimization in a flexible job shop: A tabu search approach", *Journal of Intelligent Manufacturing,* Vol.15, pp. 103-115.

113) Shafaei, R., and Brunn, P., (1999), "Workshop scheduling using practical (inaccurate) data. Part 2: An investigation of the robustness of the scheduling rules in a dynamic and stochastic environment", *International Journal of Production Research*, Vol. 37, Issue 18, pp. 105–4117.

114) Shakhlevich,N.V., Sotskov,Y.N., and Werner,F., (2000), "Complexity of Mixed Shop Scheduling Problems: A Survey", *European Journal of Operational Research*, Vol. 120, Issue 2, pp.343-351.

115) Simoes, A. and Costa, E., (1999), "Enhancing Transposition Performance", Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1434-1441, Piscataway, IEEE Press, New Jersey, USA.

116) Simoes, A. and Costa, E., (1999), "Transposition versus Crossover: An Empirical Study", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99), pp. 612-619, Morgan Kaufmann Orlando, Florida USA.

117) Simoes, A. and Costa, E., (1999), "Transposition: A Biologically Inspired Mechanism to Use with Genetic Algorithms. Proceedings of the Fourth Int. Conference on Neural Networks and Genetic Algorithms, pp. 178-186, Springer-Verlag.

118) Srinivas, N., and Deb, K., (1994), " Multiobjective optimization using nondominated sorting genetic algorithm", *Journal of Evolutionary Computation*, Vol. 2, Issue 3, pp. 221—248.

119) Stadler, W., (1984), "Multicriteria optimization in Mechanics: A survey", *Applied Mechanics Review,* Vol. 37, Issue 3, pp.277-286.

120) Subramaniam, V., and Raheja, A.S., (2003), "mAOR: A heuristic-based reactive repair mechanism for job shop schedules" *International Journal of Advanced Manufacturing technology*, Vol. 22, pp. 669–680.

121) Subramaniam, V., Raheja, A.S., and Reddy, K.R.B., (2005), "Reactive repair tool for job shop schedules", *International Journal of Production Research,* Vol. 43, pp. 1-23.

122) Sun, D. K., Batta, R., and Lin, L., (1995), "Effective Job-Shop Scheduling Through Active Chain Manipulation", *Computers & Operations Research,* Vol. 22, Issue 2, pp. 159-172.

123) Sun, D., and Batta, R., (1996), "Scheduling larger job shops: a decomposition approach", *International Journal of Production Research*, Vol. 34, Issue 7, pp. 2019 – 2033.

124) Suresh, R.K., and Mohanasundaram, K.M., (2005), "Pareto archived simulated annealing for job shop scheduling with multiple objectives", *International Journal of Advanced Manufacturing Technology*, Vol. 29, Issue 1, pp.184–196.

125) Szelke, E., and Kerr, R. M., (1994), "Foreword", IFIP Transactions B Applications in Technology, B-15, pp. 1.

126) T'kindt, V., Billaut, J.C., (2002)"Multicriteria Scheduling: Theory, Models and Algorithms", Springer, Berlin, Germany.

127) Valente, J.M.S., and Alves, R.A.F.S., (2007), "Heuristics for the Early/Tardy Scheduling Problem with Release Dates", *International Journal of Production Economics,* Vol. 106, Issue 1, pp. 261-274.

128) Vepsalainen, A. P. J., and Morton, T. E., (1987), "Priority Rules for Job Shops with Weighted Tardiness Costs", Journal of *Management Science*, Vol. 33, Issue 8, pp. 1035-1047.

129) Vilcot, G., Esswein, C., and Billaut, J.C., (2006), "Multi-objective genetic algorithm for the flexible job shop problem", 7th International Conference on Multi-Objective Programming and Goal Programming, Loire Valley , France, June 12–14, 2006.

130) Wang, T.Y., Wu K.B., (2000), "A revised simulated annealing algorithm for obtaining the minimum total tardiness in job shop scheduling problems", *International Journal of Systems Science*, Vol. 31, pp.537-542.

131) Willis, M., Hiden, H., Marenbach, P., McKay, B., and Montague, G. A., (1997), "Genetic programming: An introduction and survey of applications", Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications 1997, University of Strathclyde, Glasgow, UK, September 1-4.

132) Wu, L., and Wang, Y., (1998), "An Introduction to Simulated Annealing Algorithms for the Computation of Economic Equilibrium", *Journal of Computational Economics,* Vol. 12, No. 2, pp. 151-169.

133) Yamada, T., and Nakano, R., (1997), "Genetic Algorithms for Job-Shop Scheduling Problems", Proceedings for the Modern Heuristic for Decision Support, London, March 18-19, 1997, pp. 67-81.

134) Yang, X., and Gen, M., (1994), "Evolution program for Bicriteria transportation problem", Proceedings of the 16th international conference on computers and industrial engineering, Ashikaga, Japan, pp. 451-454.

135) Yoshitomi, Y., (2002), "A genetic algorithm approach to solving stochastic job-shop scheduling problems", *International Transaction in Operational Research*, Vol. 9, pp. 479– 495.

136) Z.X. Ong, J.C. Tay, C.K. Kwoh, (2005), "Applying the clonal selection principle to find flexible job-shop schedules", Artificial Immune Systems, pp. 442-455, Springer, Berlin, Germany.

VITA

Haritha Metta was born on June 7th, 1984 in Hyderabad, Andhra Pradesh, India. She completed her secondary school education from Hyderabad Public School, Hyderabad and finished her pre-college from Gowtham Junior College, Hyderabad. She received her bachelors' degree from Jawaharlal Nehru Technological University, Andhra Pradesh, India in May of 2005. She joined the University of Kentucky in the Fall of 2005 to pursue her Masters in Mechanical Engineering. During her course of Masters she served as a Teaching Assistant and Research Assistant.

Scholastic and Professional Honors

Harper Fellowship, Fall 2007 – Fall 2008
University of Kentucky, Lexington, KY

Kentucky Graduate Scholarship, Fall 2005 – To Present
University of Kentucky, Lexington, KY.

Papers and Conferences

1. **Haritha Metta**, Kirthi Bedida, Fazleena Badurdeen, Phil Marksberry, "From Maintenance to Sustainable Asset Management", Submitted to the Journal of Quality in Maintenance Engineering.

2. **Haritha Metta**, Kirthi Bedida, Fazleena Badurdeen, Phil Marksberry, "Total Productive Maintenance for a Sustainable Enterprise: Status and Future Directions", IMECE 07, Seattle, Washington, November 2007.

3. **Haritha Metta**, Arvind Goyal, Brandon Stump, Kundana Inala, Smitha Thuramalla, Fazleena Badurdeen, "A Simple Simulation Model to Demonstrate Mass Customization Strategies", 2007, Third World Conference on Mass Customization & Personalization, October 07-11, 2007, Boston, MA.

4. Chris Stovall, **Haritha Metta**, Mark Williams, Mike Effgen and Ibrahim S. Jawahir "Optimized Pallet Geometry for Sustainable Manufacturing of Porous Tungsten Products", Sustainable Manufacturing V: Global Symposium on Sustainable Product Development and Life Cycle Engineering, September 18-22, New York 2007.

5. **Haritha Metta**, Kirthi Bedida, "Application of Meta-Heuristic methods to solve complex manufacturing problems", Poster Presentation, UK GS conference, Lexington, KY, April 2007.

6. **Haritha Metta**, Harika Bose, "A Literature Review on Robotics and Automation", for Vector '04: A National Level Technical Symposium, Hyderabad, INDIA, May 2004.