The scheduling of manufacturing systems using Artificial Intelligence (AI) techniques in order to find optimal/near-optimal solutions.

THE SCHEDULING OF MANUFACTURING SYSTEMS USING ARTIFICIAL INTELLIGENCE (AI) TECHNIQUES IN ORDER TO FIND OPTIMAL/NEAR-OPTIMAL SOLUTIONS

SHAHID MAQSOOD

PhD

UNIVERSITY OF BRADFORD

2012

THE SCHEDULING OF MANUFACTURING SYSTEMS USING ARTIFICIAL INTELLIGENCE (AI) TECHNIQUES IN ORDER TO FIND OPTIMAL/NEAR-OPTIMAL SOLUTIONS

Keywords: Job Shop Scheduling Problem (JSSP), Genetic Algorithm (GA), Heuristics, Optimisation, Benchmark Problems

ABSTRACT

This thesis aims to review and analyze the scheduling problem in general and Job Shop Scheduling Problem (JSSP) in particular and the solution techniques applied to these problems. The JSSP is the most general and popular hard combinational optimization problem in manufacturing systems. For the past sixty years, an enormous amount of research has been carried out to solve these problems. The literature review showed the inherent shortcomings of solutions to scheduling problems. This has directed researchers to develop hybrid approaches, as no single technique for scheduling has yet been successful in providing optimal solutions to these difficult problems, with much potential for improvements in the existing techniques.

The hybrid approach complements and compensates for the limitations of each individual solution technique for better performance and improves results in solving both static and dynamic production scheduling environments. Over the past years, hybrid approaches have generally outperformed simple Genetic Algorithms (GAs). Therefore, two novel priority heuristic rules are developed: Index Based Heuristic and Hybrid Heuristic. These rules are applied to benchmark JSSP and compared with popular traditional rules. The results show that these new heuristic rules have outperformed the traditional heuristic rules over a wide range of benchmark JSSPs. Furthermore, a hybrid GA is developed as an alternate scheduling approach. The hybrid GA uses the novel heuristic rules in its key steps. The hybrid GA is applied to benchmark JSSPs. The hybrid GA is also tested on benchmark flow shop scheduling problems and industrial case studies. The hybrid GA successfully found solutions to JSSPs and is not problem dependent. The hybrid GA performance across the case studies has proved that the developed scheduling model can be applied to any real-world scheduling problem for achieving optimal or near-optimal solutions. This shows the effectiveness of the hybrid GA in real-world scheduling problems.

In conclusion, all the research objectives are achieved. Finaly, the future work for the developed heuristic rules and the hybrid GA are discussed and recommendations are made on the basis of the results.

ACKNOWLEDGEMENTS

In the name of Allah, Most Gracious, Most Merciful

Praise be to Allah, Lord of the Universe, that I am able to complete this work. Peace and prayer be upon His last Prophet and Messenger, Muhammad SAW.

With my deepest gratitude and appreciation, I sincerely thank my Supervisors, Dr. M. K. Khan (Associate Dean) and Professor Alastair Wood (Dean) in the School of Engineering, Design and Technology, University of Bradford. I am particularly indebted to Dr. Khan for his endless motivation, cooperation, patient guidance, practical suggestions and, most importantly, his precious time for making this thesis possible.

A debt of gratitude to the experts in manufacturing systems and my colleagues Professor Dr. Iftikhar Hussain and Professor Dr. Sahar Noor at Department of Industrial Engineering, KPK University of Engineering and Technology (UET), Peshawar, Pakistan for their support and cooperation during the GA model development.

I would also like to thank the Board of Trustees, Endowment Fund Project, KPK University of Engineering and Technology (UET), Peshawar and Higher Education Commission (HEC), Pakistan for their financial support for the research.

There is a long list of friends who have been very supportive and encouraging. I am thankful to all of them.

Finally, this thesis is dedicated to my parents, wife, children, family members and my teachers for all their support, patience, encouragement, good wishes and prayers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ANN	Artificial Neural Networks
CM	Cellular Manufacturing
CR	Critical Ratio
ES	Expert Systems
EDD	Earliest Due Date
FIFO	First In First Out
FL	Fuzzy Logic
FSSP	Flow Shop Scheduling Problem
GA	Genetic Algorithm
GAP	Relative deviation from optimum
HGA	Hybrid Genetic Algorithm
HybH	Hybrid Heuristic Rule
IBH	Index Based Heuristic Rule
JSSP	Job Shop Scheduling Problem
KB	Knowledge Base
MP	Management Priority
MS	Minimum Slack
MSE	Mean Square Error
OSSP	Open Shop Scheduling Problem
PDR	Production Dispatching Rules
Process Plan	Sequences of machines for operations of parts
PT	Processing Times for each job
SA	Simulated Annealing
SASM	Shauful Alam Steel Mills

CHAPTER 1

INTRODUCTION

1.1 Introduction

In current manufacturing environments, low unit cost and high quality products no longer solely define an efficient manufacturing system (Wu et al., 2000; Maqsood et al., 2011). To maintain market share, a manufacturing system must be responsive (Jain et al., 1999). Manufacturing systems with characteristics such as fluctuating demand, product varieties and priorities, imbalanced capacity, job re-entry into machines, alternative machines with unequal capacity, and shifting bottlenecks make scheduling a very difficult task (Chen, 2009). These conflicting requirements demand efficient, effective, and accurate scheduling that is complex in all but the simplest of production environments. As a result, there is a great need for effective scheduling algorithms and heuristics to find feasible solutions to such complexities (Jain and Meeran, 1999; Chen, 2009).

Scheduling consists of allocating and sequencing activities that need to be performed within a set of limited available resources (Low et al., 2009). In a successful manufacturing system, several key functions are embodied within manufacturing. The scheduling activity is used to optimize the utilization of resources and has become an essential contributor to manufacturing systems. The contemporary business environment can be characterized by expanding the global competition and customer individualism leading to a high variety of products made in relatively low volumes (Tariq, 2008; Low and Yeh, 2009; Mohamed et al., 2011). It has been estimated that more than 75% of manufacturing occurs in batches of less than 50 items (Askin et al., 1993). Therefore, recently manufacturing systems have been kept under constant pressure by the unpredictability in demand and the ever-decreasing product life-cycle, and are finding it increasingly challenging to meet these demands. These are the main challenges that low-volume manufacturing sectors are facing. The job shop manufacturing environment suits the aforementioned challenges and is widely used to provide immediate benefits.

For the past 40 years, researchers have applied different techniques, particularly Artificial Intelligence (AI) based techniques, to manufacturing scheduling problems due to their abilities to resolve the complexities involved. Jain and Meeran (1999) carried out a detailed literature review of the solution techniques that are used to solve JSSPs, and recently Maqsood et al. (2010) carried out a detailed review of Artificial Intelligence (AI) techniques used for manufacturing scheduling. These reviews concluded that the discipline of scheduling remains open to significant research and development.

In this chapter, the research problem and its scope is defined, together with the objectives of the research and the proposed systematic approach for achieving the objectives. Sub-sections of the approach are elaborated in the final section of this chapter.

1.2 The Research Problem

The classic $n \times m$ Job Shop Scheduling Problem (JSSP) is to schedule production times for n different jobs on m different machines (Askin and Standridge, 1993). The JSSP concerns the determination of the operation sequences on the resources in order that the Makespan is minimized, i.e. the time required to complete all jobs (Gen et al., 2008). JSSP consists of several assumptions (Cheng et al., 1996):

- At time 0, a set of n jobs is available;

- Each machine processes only one job at a time;
- Each job processes on one machine at a time and the job does not visit the same machine twice;
- The processing time of each operation is known;
- There are no precedence constraints amongst the operations of different jobs;
- Operations are non-preemptive, i.e. a running operation is executed until completion;
- Neither release times nor due dates are specified.

A JSSP is the most general and popular hard combinational optimization problem in manufacturing systems (Park et al., 2003; Pan et al., 2009; Lei, 2010; Yusof et al., 2010). This is because of its large solution space, which is very difficult to handle. For example, if n different jobs are to be processed on m different machines, then there are $(n!)^m$ alternatives amongst which an optimal solution for a certain measure of performance exists. For example, a very simple problem of 20 jobs and 20 machines will give 5.27×10^{367} possible alternative solutions. From amongst these alternative solutions an optimal and feasible solution is to be determined. With a high performance computer that could evaluate one alternative in one microsecond, It could take more than 1000 years to find the optimal solution with exact approaches, (Hitomi, 1996; Morshed, 2006). The computational requirements of analysis and classification of instances as “hard” or “easy” is known as complexity theory (Garey and Johnson, 1979). According to Garey and Johnson (1979), JSSP belongs to class of NP decision problems that can be solved in polynomial time by a non-deterministic computer. NP stands for non-deterministic polynomial. Historically the NP term was introduced in certain computational devices called non-deterministic Turing machines (Turing, 1936). NP means that it is not possible to solve an arbitrary instance in polynomial time unless P=NP , where P is a sub-class of NP and consists of sets of problems that can be solved deterministically by a polynomial time algorithm (polynomial time is a synonym for "tractable", "feasible", "efficient", or "fast”). NP problems can be NP-complete and NP-hard. The NP-complete problem belongs to set of NP for which no efficient solution algorithm has been found. According to Blazewicz et al. (1996), if there is a polynomial algorithm for any NP-complete problem then there are polynomial algorithms for NP-complete problems. The NP-hard problem is class of NP problems that are at least as hard as the hardest problems in NP. Most of the special cases of JSSP are NP-hard, which makes the JSSP one of the most stubborn members of NP (Zhou, 2001; Yamada, 1992).

In JSSP, each job comprises a set of operations. The operation order on machines is pre-specified, and each operation is characterized by a required machine and a fixed processing time (Jain and Meeran, 1999; Gen et al., 2008). However, there remains a lot of potential for improvement in existing techniques.

According to Noor (2007), the inherent shortcomings of solutions to scheduling problems has directed researchers to develop hybrid approaches as no single AI tool for scheduling has yet been successful in providing an optimal solution. The hybrid approach complements the merits of, and compensates for the limitations of, each individual AI technique towards better performance and improved results in solving both static and dynamic production scheduling environments. Over the years, hybrid approaches have generally outperformed single techniques such as the simple Genetic Algorithm (GA). It is anticipated that future AI hybrid approaches will solve real-world dynamic scheduling problems and will provide a reliable and efficient tool for solving scheduling problems.

Realizing the fact that real-world scheduling problems are mostly dynamic and multi-objective in nature, a framework for a job shop scheduling system is needed for providing an optimal or near-optimal solution and to achieve certain objectives.

1.3 Research Objectives

The main objective of this research is to develop a hybrid scheduling methodology using Genetic Algorithms (GAs) for solving JSSPs in order to find an optimal or near optimal solution for the selected performance criteria (makespan, flow times, earliness, tardiness).

More explicitly, the objectives are as follows:

a. A detailed review of scheduling problems and solution approaches in order to ascertain the contemporary knowledge and information relating to scheduling, with the aim of acquiring knowledge in this area for designing conceptual and actual simulation models for production scheduling.
b. Study of existing scheduling solution techniques, such as the traditional heuristic approaches, Branch and Bound (B&B), Genetic Algorithm (GA), Artificial Neural Networks (ANN), Knowledge Based Systems or Expert Systems, Fuzzy Logic (FL), Simulated Annealing (SA), Approximation Based Techniques, Mathematical methods, etc., to ascertain recent knowledge and information related to scheduling, and to find suitable techniques after comparing their strengths and weaknesses. This literature study will identify the area of research to focus upon.
c. To develop an intelligent search algorithm based upon new heuristic approaches and GA:

a) Develop new heuristic rules for the initial solution generation, check their performance, and compare results with traditional heuristic rules for selected scheduling problems;
b) Incorporate newly developed heuristic rules within the GA and design the Hybrid Genetic Algorithm (HGA). The GA representation, operators and parameters will be selected based on literature review;
c) Validate and test the performance of the new heuristics and HGA by applying these techniques to benchmark JSSPs and comparing the results and algorithm performance with existing solution methods;
d) Apply the developed HGA to industrial case studies available in the literature to gauge its strength and to check its performance on real-world scheduling problems;
e) Identify future work.

1.4 Conceptual Approach to the Problem

To achieve the research objectives a conceptual approach is adopted as shown in Figure 1.1.

This research is divided into three stages. In the first stage a detailed literature review is carried out regarding scheduling problems, approaches to their solutions, and the current trends in scheduling theories. The second stage forms the main part of this research in which novel heuristic rules and hybrid GAs approaches are developed. The developed heuristics and HGA are tested on several benchmark problems and industrial case studies in order to check their performance and to gauge their strengths and weaknesses. In the third stage a detailed analysis of the results of the computational experiments is carried out.

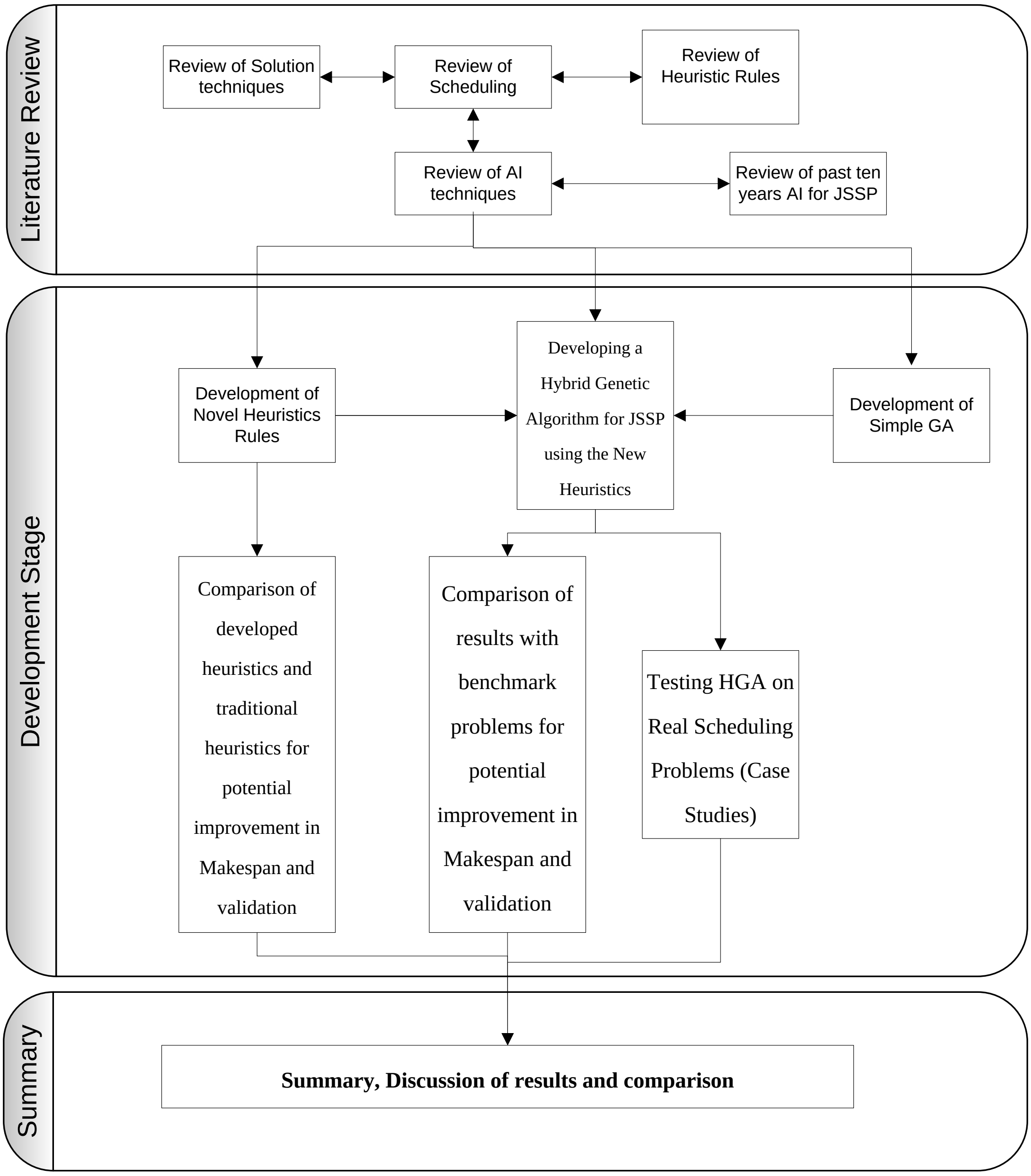


Figure 1. 1: Conceptual approach for the proposed research

The classic $n \times m$ *minimum-makespan* Job Shop Scheduling Problem (JSSP) is a hard combinatorial problem. The performance of a JSSP depends upon the significance of selecting the best heuristic methods and meta-heuristic techniques with few or no assumptions about the problem and which can search very large candidate solution spaces. The heuristic method has provided quick solutions and active schedules for scheduling problems during the past 60 years. These heuristics have been used in combination with meta-heuristics in order to achieve improved Makespan solutions. From the literature, it is evident that the better the initial solution from heuristic rules, or any other method, the better the final solutions from meta-heuristics techniques. Therefore, there is a need for novel heuristic rules that can perform effectively across all sizes of scheduling problem, and, novel heuristics will be proposed in this research with their results being compared to traditional heuristic rules in order to evaluate their performance. After the development of the heuristic rules, they will be combined with a meta-heuristic tool such as GA. The evaluation process in the case of a GA, for the JSSP, is a key step that determines the fitness of the objective function. The developed novel heuristics will be used in the key step of the GA, i.e., for evaluation and the initial solution set.

1.5 Organization of the Thesis

This thesis consists of seven chapters. Chapter 1 covers an introduction to the research problem, research objectives, and their justification. Chapter 2 focuses upon an introduction to various manufacturing environments, a literature review of solution techniques, and consideration of different scheduling criteria and notations used in scheduling theory. Chapter 3 provides a literature review of AI techniques applied to solve JSSPs, including an updated review of the hybrid approaches. Chapter 4 provides a development and design of two novel heuristic rules for scheduling problems: The Index Based Heuristic (IBH) and the Hybrid Heuristics Rule (HybH). The proposed heuristic rules are applied to several benchmark JSSPs and industrial case studies from the literature in order to check the validity and effectiveness of the proposed heuristics. Chapter 5 highlights a proposed Hybrid Genetic Algorithm (HGA) with the aim of achieving optimal (or near-optimal) solutions for the benchmark JSSPs. The chapter presents a detailed description of a genetic algorithm used to encode the job shop schedule (different genetic parameters and parameter analysis (sensitivity analysis) for a wide range of benchmark JSSPs). Chapter 6 presents results for the job shop and flow shop scheduling problems, for the two novel heuristic rules, i.e. HybH and IBH (developed in Chapter 4) and for the HGA (developed in Chapter 5). The chapter also presents HGA results from a computational test-bed consisting of benchmark JSSPs of various sizes, and presents a discussion of the results from the proposed heuristics and HGA for industrial case studies. Chapter 7 details conclusions and recommendations for future work, both for the proposed heuristics and HGA.

1.6 Summary

This chapter has provided a brief background to scheduling problems and the research objectives. It also covered the main contributions of this research to the area of scheduling, which are the development of the novel heuristics and HGA for JSSPs and FSSPs and their application to real test cases.

CHAPTER 2

INTRODUCTION TO MANUFACTURING SCHEDULING

2.1 Introduction

According to Pinedo et al. (1998) scheduling deals with the allocation of scarce resources to tasks over time. It is a decision making process with the aim of optimizing one or more objectives such as Makespan, due-dates, and completion time. A resource may be the machines in a workshop, the work center in a factory, the CPU of a computer, airport gates and runways, etc. The task may be a production process, boarding, execution of different computer programs, landing or take-off at an airport, with a certain priority level, start time, finish time, etc.

This scheduling is an important element that has a major impact upon the efficiency of manufacturing and production systems since system performance depends upon optimal (or good) schedules. Companies must therefore have an efficient scheduling framework at their disposal that can provide the production system with a quick and efficient schedule (Bai, 1998).

Scheduling problems are often complicated by a large number of constraints such as time restrictions (deadlines, precedence etc.) and resource envelopes (Lopez et al., 2008). For example, there may be precedence constraints connecting activities that specify those activities that must precede other activities, and by what delay and/or by how much allowed overlap. Resource constraints may be unavailability for a specific interval of time due to planned maintenance. These constraints and complex inter-relationships can make an exact or an optimal solution of a large scheduling problem very difficult to obtain. These issues have arisen in a large number of scheduling models.

Past researchers carried out significant amounts of investigation into polynomial time algorithms for deterministic scheduling problems with the assumption that there are a finite numbers of jobs with known processing times, and with one or more objective functions. However, many scheduling problems are NP-hard, which do not have polynomial time algorithms. The difficulty level of an NP-hard scheduling optimization problem is similar to combinatorial optimization and stochastic modelling. In stochastic models it is assumed that jobs are finite, and that there are no known job data such as processing time, start time, due date, etc.; only their distributions are known in advance. These models are single objective optimization problems. Researchers in the past have focused on the borderline between polynomial time solvable problems and NP-hard problems (Pinedo et al., 1998). Real time scheduling frameworks or models also face problems in implementation due to input data reliability issues.

2.2 Scheduling in Manufacturing Systems

In manufacturing environments, released orders normally have to be translated into jobs with associated due dates. These jobs often have to be processed on one or more machines in a Work Centre (WC), in a given sequence, for a certain amount of processing time. The processing of jobs may take longer in queues mainly due to the three Ms (Man, Machine and Materials). Lesser-skilled labour may result in longer than expected processing time. Machine breakdowns, or a late supply of raw or semi finished items from vendors or other work centers, may cause delays in completion time.

The shop floor is not the only part of the organization that impacts the scheduling process. It is also affected by the production planning process that handles the medium-term to long-term planning for the entire organization (Pinedo et al., 1998).

The scheduling function has to interact with other decision-making functions. A decision made at higher planning levels may impact the scheduling process directly. Figure 2.1 shows the information flow diagram for a generic manufacturing environment and the role of manufacturing scheduling in the system. In cases where a facility does not have a scheduling system, the MRP system may be used for the production planning purposes (Pinedo et al., 1998).

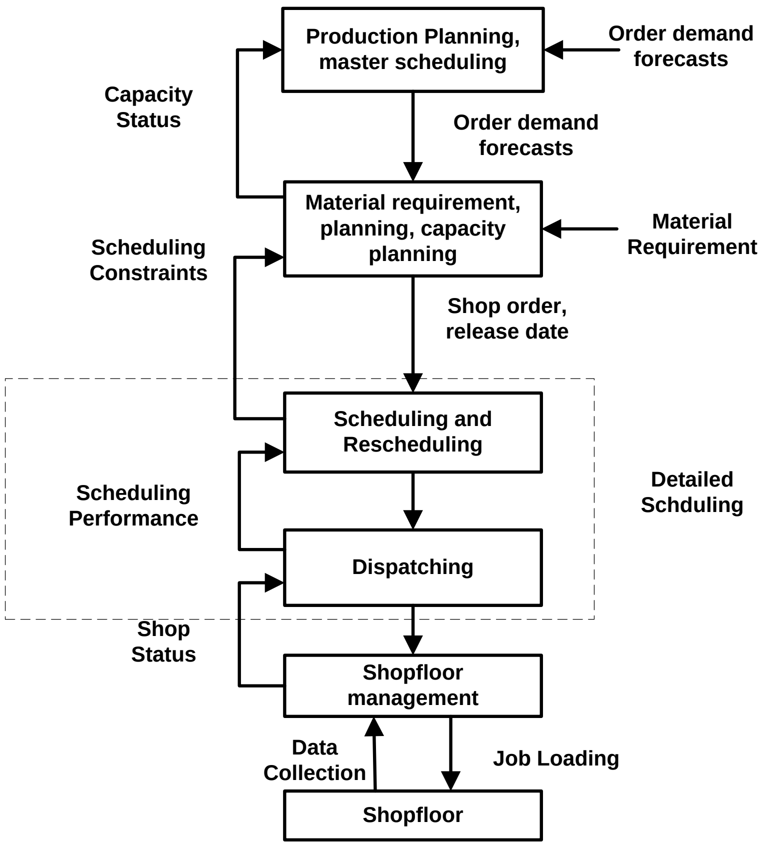


Figure 2. 1: Information flow diagram of a Manufacturing System (Pinedo et al., 1998)

In the following sections of this chapter, brief introductions are presented to various manufacturing scheduling environments, their notations, several classes of manufacturing schedules, and complexity of scheduling problems. The manufacturing environments range from small, complex, and custom job shops to high speed, low product, variety transfer lines; from discrete parts manufacturing to continuous process flows. Although several key functions are embodied within manufacturing, the scheduling activity has become an essential contributor to a successful manufacturing system.

2.3 Manufacturing Scheduling Environment

In defining a scheduling problem, constraints on jobs that are determined principally by the flow pattern of the jobs on machines and the scheduling objective must be specified. In this context, some well-known scheduling environment definitions are as follows:

2.3.1 Job Shop Manufacturing

In job shop manufacturing, there are m machines. For a finite number of jobs, each job has a predefined route to follow. Each job visits each machine once. The main aim of the job shop is to achieve a higher degree of flexibility so that products having a wide range of variation in size and shape can be produced in small lot sizes and in a single facility (Tariq, 2008). The distinguishing feature of the job shop is the manufacturing of products that may have different processing sequences and variations in processing times. Operations are performed sequentially on a single lot of parts that travel either in batches or together through the entire shop. There are no shop floor inventories that are not identified with a single activity. Job shop manufacturing is highly complicated and does not repeat in any simple way. The main dictating force in the selection of machines is the variety of products and smaller lot sizes. This is the reason that in job shop manufacturing, general-purpose machines are mainly utilized as they can perform a variety of operations. The grouping of machines in a job shop environment is carried out on the basis of functions, e.g. lathe machines are placed in one Work Centre (WC), milling machines in another and so on. In Figure 2.2 the environment will be a job shop if there is a single machine in each WC.

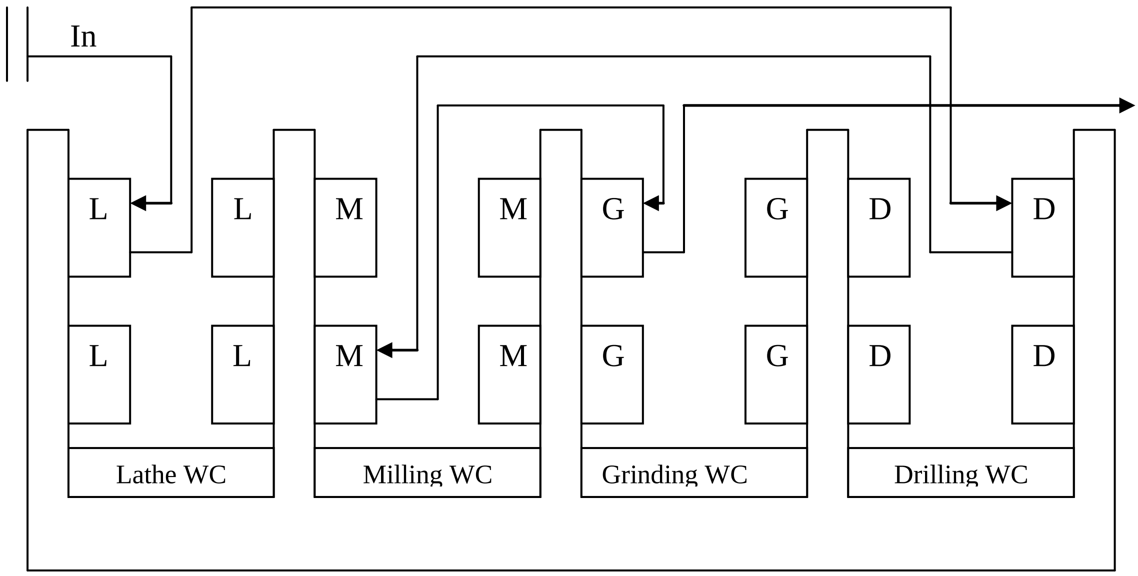


Figure 2. 2: Flexible Job Shop Layout

2.3.1.1 Flexible Job Shop Manufacturing

In flexible job shop manufacturing, instead of m machines, there are c WCs with a number of identical machines in parallel as shown in Figure 2.2. Each job has its own route and has to be processed on a single machine in each WC.

2.3.2 Flow Shop Manufacturing

In flow shop manufacturing, there are m machines in the series. Each job from a finite number of n jobs has to be processed on each one of the m machines. All jobs follow the same sequence in a series of m machines, i.e. machine 1 (Lathe), then machine 2 (Grinding), etc. For the simplest case, each job consists of the same set of activities to be performed sequentially on the same set of machines in multiple sets of machines, as shown in Figure 2.3.

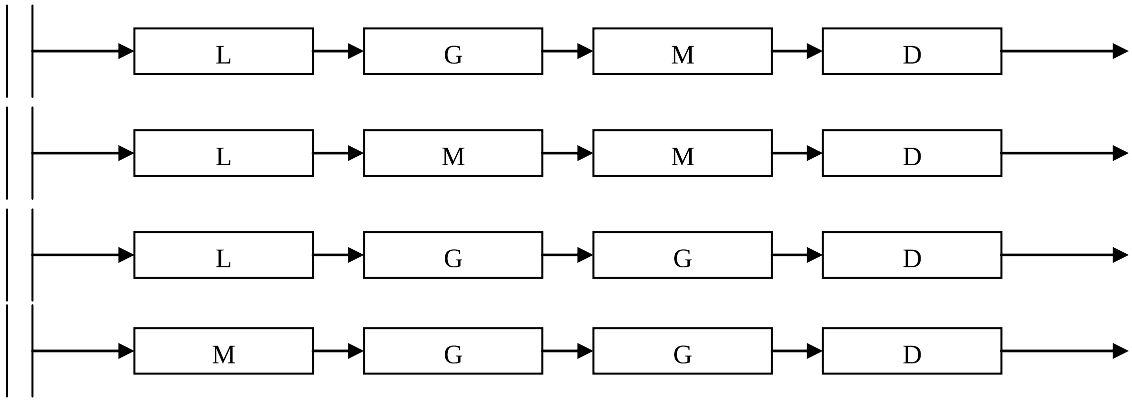


Figure 2. 3: Flow Shop

In flow shop manufacturing, all queues are usually assumed to operate on the basis of the First In First Out (FIFO) rule, i.e. the first job in the queue will be processed first, always followed by the second, then the third, and so on.

2.3.2.1 Flexible Flow Shop

The flexible flow shop manufacturing is a generalized form of flow shop manufacturing. Instead of m machines in the series there are c WCs in series. Each WC consists of a number of identical machines in parallel. Each job has to be processed on each WC. At every stage, a job requires processing on only one machine in a WC. A typical flexible flow shop manufacturing layout is shown in Figure 2.4.

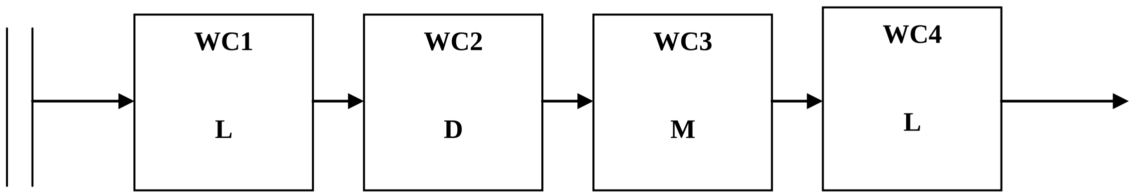


Figure 2. 4: Flexible Flow Shop

2.3.3 Open Shop Manufacturing

The open shop is similar to the job shop with the exception that there are no precedence constraints between the operations of each job. The work in process inventories, or nearly finished products, are also high in order to provide jobs to high priority customers.

2.3.4 Summary

Figure 2.5 illustrates the relationship between the previously mentioned manufacturing machine shop environments. These scheduling environments can be solved for various objectives.

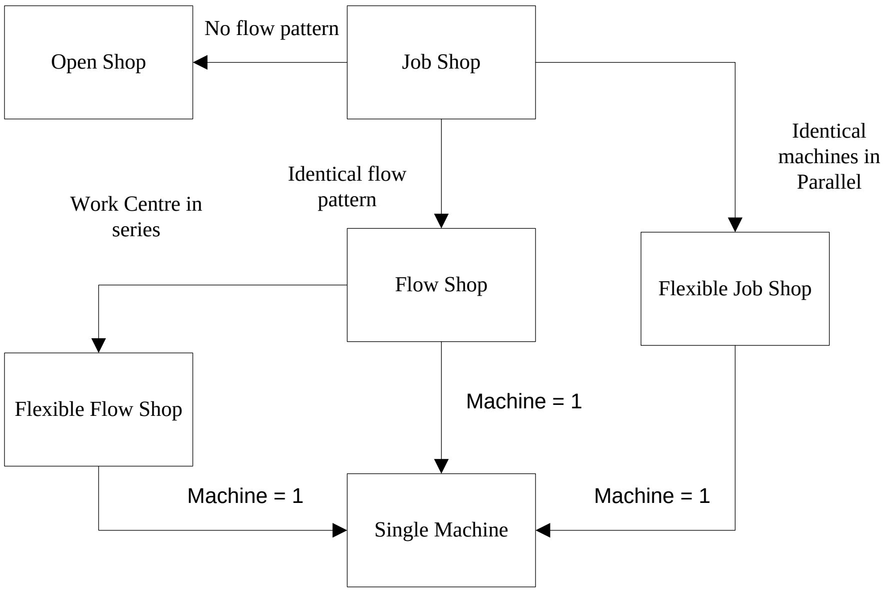


Figure 2. 5: Relationship between different machine shop environments

In a job shop, each job has its own individual flow pattern or predefined constrained route through which the job must pass. In a flow shop, however, each job has an identical flow pattern, whereas an open shop has no specific flow pattern. The flexible job shop and flexible flow shop environments have parallel identical machines and parallel work centres, respectively.

2.4 Notation for Scheduling Problems

Conway et al. (1967) provided a classification scheme for scheduling problems based upon descriptors A/B/C/D, which has since been followed by a number of researchers (Maccarthy et al., 1993). The meaning of each letter is described in Table 2.1.

Table 2.1 Four letter classification schemes

S. No.	Letter	Meaning
1	A	Any positive integer N, represents number of jobs
2	B	Any positive integer M, represents a number of machines
3	C	Represents flow pattern such as: *M*: Single machine, *J* : Job Shop, *F* : Flow Shop, *P* : Permutation Flow Shop *O* : Open Shop
4	D	Represents Scheduling criteria to be optimized (discussed in detail in Section 2.6) such as: C_{max}: Minimization of Makespan F_{max}: Minimization of Maximum Flow Time

For example, $n/m/J/C_{max}$ is a Job Shop Scheduling Problem (JSSP) with n jobs to be processed on m machines, which attempts to minimize makespan. Mccarthy and Liu (1993) state that the four field notation has been used widely by researchers and they suggested several modifications for the C descriptor:

$$C \in \{k-parallel, r_j, Str, Prec, prmt, unit, eq, depend, setup\}$$

Table 2.2 shows the meaning of each descriptor.

Table 2.2: C Letter Descriptor's Meanings

S. No.	Descriptor	Meaning
1	$k - paralle$	*k machines in parallel*
2	r_j	*Jobs with ready time*
3	Str	*Strings jobs*
4	$Prec$	*Precedence constraints*
5	$prmt$	*Pre-emption is allowed*
6	$unit$	*Unit processing time*
7	eq	*Equal processing time for all jobs*
8	$depend$	*Dependent jobs*
9	$setup$	*Sequence-dependent setup times*

According to Graham et al. (1979) all scheduling problems are described by a triple $\alpha \mid \beta \mid \gamma$. α describes the machine shop environment and contains a single entry, β provides details of processing constraints (may have no entry or multiple entries), and γ describes the objective function to be optimized (may be single or multiple). The number of jobs is usually denoted by n and the number of machines by m. Subscript j refers to a job and subscript i refers to a machine. If a job requires a number of processing steps or operations, the ordered pair $(i\,, j)$ refers to the processing step (operation of job j on machine i). For example, p_{ij} represents the processing time of job j on machine i. If the processing time of j is independent of the machines, then i is omitted. Release date (r_j) of job j is also known as the ready date or job arrival time to the system. d_j represents the due date of job j or the promised date with the customers. w_j, the weight of job j, represents its importance over other jobs.

Table 2.3 lists a summary of the most common scheduling environments specified by α in the manufacturing scheduling problem ($\alpha \mid \beta \mid \gamma$).

Table 2.3: Manufacturing scheduling environments (Pinedo et al., 1998)

	Type	A	Characteristics
1	Single Machines	1	Continuous flow, Single Machines, simplest machine environment
2	Identical Machines in parallel	P_m	Discrete or continuous, linear or complex, grouping
3	Flow Shop	F_m	Discrete or continuous, linear flow, jobs all highly similar, grouping and lotting important
4	Job Shop	J_m	Discrete, complex flow, unique jobs, no multi-use parts
5	Open Shop	O_m	Discrete, complex flow, some repetitive jobs and/or multi-use parts
6	Batch Shop	B_m	Discrete or continuous, less complex flow, many repetitive jobs and multi-use parts, grouping and lotting important
7	Manufacturing Cell	MC_c	Discrete, automated grouped version of open job shop or batch shop

2.5 Job Shop Scheduling Model

According to Blazewicz et al. (1996) scheduling problems can be broadly defined as "the problems of the allocation of resources over time to perform a set of tasks". The literature of manufacturing scheduling is full of very diverse scheduling problems (French, 1982; Sidney, 1983; Pinedo et al., 1998; Brucker, 2007). The Job Shop Scheduling Problem (JSSP) concerns the determination of the operation sequences on the resources so that the Makespan is minimized, i.e. the time required to complete all jobs (Gen et al., 2008). JSSP consists of several assumptions as follows (Cheng et al., 1996):

- Each machine processes only one job at a time;
- Each job processes on one machine at a time and the job does not visit the same machine twice;
- The processing time of each operation is known;
- There are no precedence constraints among the operations of different jobs;
- Operations are non-preemptive, i.e. a running operation is executed until completion;
- Neither release times nor due dates are specified.

The JSSP is considered to be one of the most difficult problems to handle due to its large solution space. For example, if n different jobs are to be processed on m different machines, then there are $(n!)^m$ alternatives amongst which an optimal solution for a certain measure of performance exists and theoretically can be found in a finite number of computational iterations. However, it is practically difficult because of the combinatorial increase of the problem size. This is why the JSSP is considered to be a member of a large class of intractable numerical problems known as NP-hard (NP stands for non-deterministic polynomial) problem and is difficult to solve optimally. For example, a very simple problem of 20 jobs and 20 machines will give 5.27×10^{367} numbers of alternatives. It will therefore need over 1000 years to find its optimal solution using a high-performance computer evaluating one alternative per microsecond (Hitomi, 1996; Morshed, 2006). Furthermore, each job is composed of a set of operations, the operation order on machines is pre-specified, and each operation is characterized by required machine and fixed processing times (Jain and Meeran, 1999; Gen et al., 2008).

Roy and Sussmann (1964) proposed the current form of JSSP and were first to propose the disjunctive graph representation. Balas (1970) was the first to apply an enumerative approach to the disjunctive graph. Since then many researchers have discussed mathematical models and tried various strategies for solving this problem (Adams et al., 1988; Blazewicz et al., 1996; Cheng et al., 1996; Jain et al., 1998; Park et al., 2000; Noor and Khan, 2007; Gen et al., 2008). Grabot et al. (1994) applied heuristic rules to JSSP. Jain and Meeran (1998) applied neural networks to JSSPs and Yang et al. (2000) used neural networks combined with an heuristic approach to solve these problems. Moreover some researchers, especially Cheng et al. (1999), using hybrid Genetic Algorithms (GAs), also obtained optimal Makespan solutions for a set of JSSPs.

2.5.1 Mathematical Formulation of JSSP

Notation

Indices

i, l: index of jobs, $i, l = 1,2,\cdots,n$

j, h: index of machines, $j, h = 1,2,\cdots,m$

k: index of operations, $k = 1,2,\cdots,m$

Parameters

n: Total number of jobs

m: Total number of machines

C_{max}: Makespan

M_j: the j^{th} machine

J_i: the i^{th} job

where $i = 1,2,\cdots,n$

O_{ikj} : the k^{th} operation of job J_i operated on machine M_j

P_{ikj}: Processing time of operation O_{ikj}

Decision Variables

t_{ikj}: Completion time of operation O_{ikj} on machine M_j for each job J_i

The JSSP are treating is to minimize the Makespan, so the problem could be described as an n-job m-machine JSSP by simple equations as follows:

$$min\ C_{max} = max_{ikj}\{t_{ikj}\} \quad \text{Equation (2.1)}$$

$$\text{s. t. } t_{i,k-1,h} + P_{ikj} \leq t_{ij}, \forall i,k,h,j \quad \text{Equation (2.2)}$$

$$t_{ikj} \geq 0, \forall i,k,j \quad \text{Equation (2.3)}$$

The objective in Eq. (2.1) is to minimize the Makespan. The constraint in Eq. (2.2) is the operation precedence constraint; the $(k-1)^{th}$ operation of job i should be processed before the k^{th} operation of the same job. The time chart for this model is illustrated in Figure 2.6.

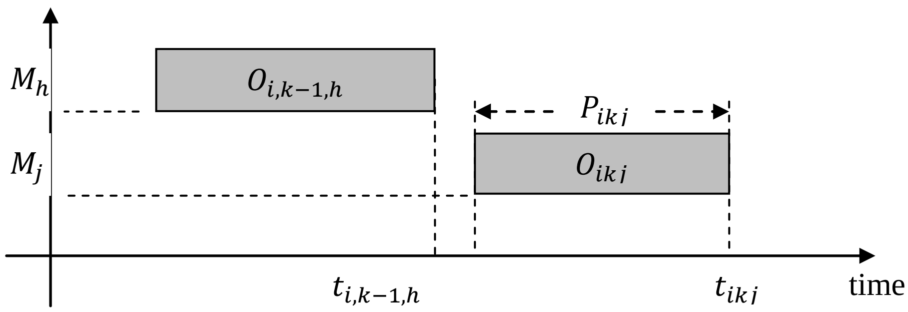


Figure 2. 6: Time Charts for Constraint

2.5.2 Justification for choosing Makespan as Objective Function

The well known Travelling Salesman Problem (a salesman travels to a number of cities) schedules the route with the objective of minimum travel distance. This is the same problem as a production scheduling problem with the objective of minimizing the Makespan.

Makespan minimization is considered to be the main driving objective function due to the fact that this criterion was the first objective considered by researchers since the post World War II industrial revolution. The problem was normally single machine or parallel machine at that time. Mathematically the C_{max} problem was easy to formulate. Consequently, it has been the principal criterion for academic research as it is able to capture the fundamental computational difficulty that exists implicitly in determining an optimal schedule. Nevertheless, this criterion is also widely used in industry because it provides a good deal of flexibility. A solution for C_{max} is likely to perform well on average with respect to the criteria of total completion time, total tardiness, total flow time, and maximum lateness (Liaw, 2000; Kis, 2003; Morshed, 2006; Zhang et al., 2009).

2.6 Scheduling Criteria

Scheduling criteria, the performance measure indicator, is defined as the goodness of (a set of) scheduling rules. Schedules are generally evaluated by aggregate quantities, involving information about a number of jobs, resulting in one-dimensional performance measures (Momim, 1999). The broad objectives of Job Shop Scheduling (White, 1987) are:

- Minimize Work-In-Process (WIP) inventory;
- Maximize utilization of resources;
- Maximize service to customers;

These objectives are usually in conflict. For example, minimization of WIP can increase capacity but can reduce utilization. Similarly, minimization of inventory will lead to under utilization of resources and an unsatisfactory service to customers. To find a satisfactory compromise on objectives in any given situation, the following criteria depending upon processing times, due dates, utilization, and inventory are commonly used for scheduling (White, 1987; Halshal, 1995):

For the j^{th} Job, define the following measures:

C_j = Completion time of Job j

d_j = Due date of Job j

O_j = Operation of Job j

r_j = Release/ ready date of Job j

C_j = Completion time of Job j

a_j = Allowence time of Job j $= d_j - r_j$

L_j = Lateness of Job j $= C_j - d_j$

T_j = Tardiness of Job j = E_j = Earliness of Job j $= \max\{-L_j, 0\}$

N = Number of Job completed

nT = Number of Tardy Jobs

F_j = Flow Time of Job j

The objective to be minimized will always be a function of the total completion time of a job and which depends upon the schedule (Pinedo et al., 1998). Moller (1966), listed 27 different objectives and Baker (1974), classified scheduling problems for a single objective function. The completion time for job j on machine i is denoted as $\boldsymbol{C}_{ij}$. Maccarthy and Liu (1993), referring to Baker's (Baker, 1974), classification, listed three types of decision making goals that have dominated the research in scheduling, and indicated commonly used measures for scheduling performance that are associated with them:

2.6.1 Efficient utilization of resources:

i) Maximum Completion Time (Makespan)

$$\mathbf{C}_{\max} = \min(\{0, Cj\}) \qquad \text{Equation (2.4)}$$

in which

$$\mathbf{C}_{j} = \text{Time of the last job released and has left the shop}$$

2.6.2 Rapid Response to demand

i) Flow Time: Total time taken by a job on the shop floor.

$$F = \sum_{j=1}^{N} (c_j - r_j) \qquad \text{Equation (2.5)}$$

ii) Mean Flow Time

$$\overline{F} = \frac{\sum_{j=1}^{N} (c_j - r_j)}{N} \qquad \text{Equation (2.6)}$$

2.6.3 Close Conformance to prescribed deadline

i) Mean Earliness

$$\overline{E} = \frac{\sum_{j=1}^{N} \max\{0, -L_j\}}{N} \qquad \text{Equation (2.7)}$$

ii) Mean Tardiness

$$\overline{T} = \frac{\sum_{j=1}^{N} \max\{0, L_j\}}{N} \qquad \text{Equation (2.8)}$$

iii) Mean Lateness

$$\overline{L} = \frac{\sum_{j=1}^{N} L_j}{N} \qquad \text{Equation (2.9)}$$

iv) Mean Absolute Lateness

$$\bar{L} = |\bar{L}| = \frac{\sum_{j=1}^{N} |L_j|}{N}$$ Equation (2.10)

v) Percent Tardiness

$$\%T = \frac{nT}{N} \times 100\%$$ Equation (2.11)

vi) Maximum Tardiness

$$T_{max} = \max\{\max(o, L_j)\}$$ Equation (2.12)

where $0 < j < N + 1$

vii) Maximum Earliness

$$E_{max} = \max\{\max(o, L_j)\}$$ Equation (2.13)

in which $0 < j < N + 1$

In most scheduling systems, one of the above scheduling criteria is either minimized or maximized, but in the real world a trade-off between these criteria is desired for an optimal output of the system. The problem becomes more complex by increasing the number of objectives and consequently an effective trade-off between them becomes difficult. Therefore, an optimal solution in real time applications is unlikely to be achieved.

2.7 Complexity of JSSP

The JSSP has already been confirmed amongst the worst members of the class of NP-complete problems in manufacturing systems (Jones et al., 1998; Vela et al., 2010). Most variants of the deterministic JSSP, except for a few formulations with the number of machines or jobs limited to 1 or 2, are known to be NP-hard (Brucker, 1988; Brucker, 2007). In particular, JSSPs with the number of machines *m* fixed (*m*

≥ 2) using the C_{max} performance criteria are NP-hard in the strong sense (Garey et al., 1978; Gonzalez et al., 1978; Gen et al., 2008).

For JSSP with deteriorating jobs, i.e. jobs whose processing times are an increasing function of their starting time results in NP hardness. Mosheiov (2002) presents NP hardness for flow shop and open shop with three or more machines and for job shops with two or more machines. Recently, Thornblad et al. (2011) presented a correction in Mosheiov's (Mosheiov, 2002) theorem 2 and proved that flow shop is NP-hard even for three machines.

Gen, Lin et al. (2008) also referred to French's (French, 1982) theorem 11.6 and Garey and Johnson's (Garey and Johnson, 1978) theorem 1 and stated that strong NP-hardness of a problem implies that it is impossible to create a search heuristic which guarantees to find a solution for which the relative error ε is bounded by

$$\frac{performance\ measure\ of\ found\ solution}{performance\ measure\ of\ optimum\ solution} \leq\ 1 + \varepsilon$$

and which runs in polynomial time both in the problem size and 1/ε. This result shows that efficient approximation algorithms with guaranteed performances should not be expected for these problems unless $P = NP$. Therefore, most research focused on finding (near) optimal schedules has been turned towards implicit enumeration algorithms (B&B techniques), local improvement methods (shifting bottleneck), and heuristic search methods such as genetic algorithms, tabu search and simulated annealing.

The JSSPs are not only NP-hard but are also very difficult to solve heuristically. For example, the Fisher and Thompson's FT-10 (10-job x 10-machines) problem (Fisher et al., 1963) remained open for 25 years until Adam et al. (1988) published an optimal solution.

2.8 Benchmark Problems

In the field of evolutionary computation, different algorithms are used to compare using large sets of data, especially when the test involves function optimization (Gordon et al., 1993). However, comparing two algorithms with all possible functions, the performance of any two will be the same (on average) (Oltean, 2004). Therefore, there is a need for benchmark problems that are perfect test sets, where all the functions are present, and allowing conclusions to be obtained from the performance of algorithms.

For JSSP the benchmark problems are developed by various researchers (Fisher and Thompson (1963) - FT; Carlier (1978)- CAR; Lawrence (1984) - LA; Adams et al., (1988) - ABZ; Applegate and Cook (1991) - ORB; Storer et al., (1992) - SWV; Yamada and Nakano (Yamada et al., 1992) – YN and Taillard (1993) - TD. The FT problems received the greatest analysis of all these problems (Morshed, 2006).

In these benchmark problems (See Appendix A and Appendix B) the precedence order and processing times for operations are generated randomly. The latter is drawn from a discrete uniform distribution (except for the ORB instances) and the objective in each problem is to minimize the Makespan (Jain and Meeran, 1999).

2.9 Solution Representation

A common charting technique, Gantt Chart, the first revolutionary technique to represent scheduling solutions, was named after Henry Gantt (Gantt, 1919). This method has been used since the early 19th century and has traditionally been the most popular method of solution representation. Blazewicz et al. (1996), indicate that the disjunctive graph model, $G\ \{N, A, E\}$ (Roy and Sussmann, 1964) is now more prevalent .

2.10 Conclusion

In this chapter a brief introduction to different manufacturing environments in manufacturing systems is presented, followed by an introduction, mathematical model, scheduling criteria, and complexity issues of JSSP. The sources and types of benchmark problems are presented at the end of the chapter. In Chapter 3, details of the literature review and analysis are discussed.

CHAPTER 3

LITERATURE REVIEW

3.1 Introduction

In this chapter a detailed literature review of scheduling problems is carried out, in order to ascertain the contemporary knowledge and information relating to scheduling. The main emphasis in this chapter is to review solution approaches to scheduling problems, and identify gaps that require further research. The effort is therefore devoted to the review of literature in the scheduling area with the aim of acquiring knowledge for designing conceptual and actual simulation models for manufacturing scheduling problems.

3.2 Solution Approaches to Manufacturing Scheduling Problems

Study of scheduling theory began in the early 50s and Johnson's (1954) article is acknowledged as a pioneering work (Maccarthy and Liu, 1993). In this section, various approaches reported in the literature to solve the scheduling problem since Johnson's first efficient algorithm are reviewed. In the literature these approaches have been categorized in various ways. Gonzalez (1978), and Momin (1999), have categorized these approaches as Mathematical, Priority Rules, Heuristics and Artificial Intelligence (AI). Jain et al. (1999), broadly categorized them as an approximation and optimization approaches, that have been further divided into a number of branches as shown in Figure 3.1.

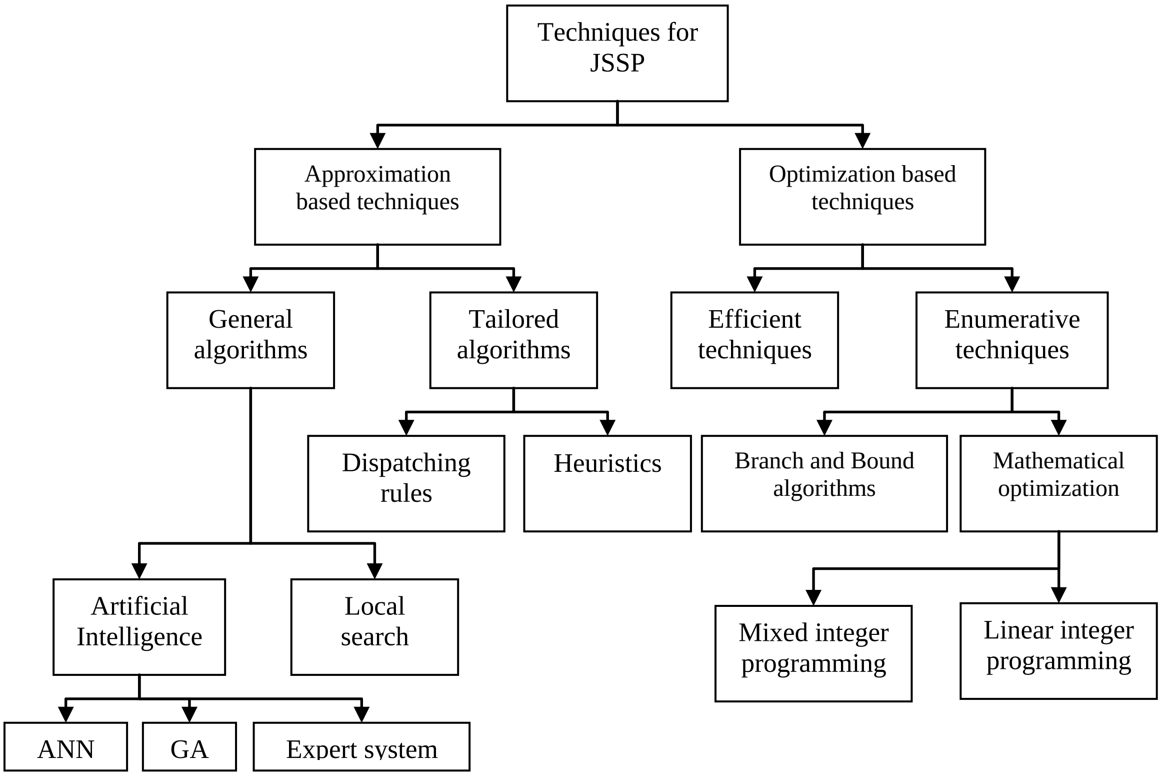


Figure 3. 1: Solution approaches to JSSP (Jain, 1998)

Optimization-based techniques are further classified as efficient techniques and enumerative techniques. The enumerative approach has two further subclasses: B&B algorithms and mathematical optimization (mixed and linear integer programming) based algorithms. Approximation techniques, on the other hand, are initially classified as general algorithms and tailored algorithms. Tailored algorithms are either dispatching rules or heuristic based algorithms, whereas general algorithms are classified as AI-based techniques (ANN, GA and Expert Systems) and local search-based algorithms. A literature review of optimization and approximation approaches is given below.

3.3 Reviews of Optimization methods

A method using an optimization criterion is exact if it guarantees optimality of the solutions found. Exact procedures are computationally expensive and, with an increase in the problem size, the computation time for finding a solution increases exponentially.

3.3.1 Efficient Algorithms

Johnson's article (Johnson, 1954) in the area of scheduling theory is acknowledged as pioneering, presenting an efficient optimal algorithm for $n/2/F/Cmax$. He generalized the algorithm and applied it to $n/3/F/Cmax$ scheduling problem. According to Conway et al. (1967), this early work had a great influence on subsequent research in the area of scheduling theory. Later Jackson (1956) and Smith (1956), developed various optimal rules for single machine problems. According to Maccarthy and Liu (1993), these early works formed the basis for much of the development of classical scheduling theory. Giglio et al. (1964), applied Johnson's method to a six-jobs, three-machine flow shop problem. In the early 1980s, Hefetz et al. (1982), developed an efficient method for n Jobs and 2 machines, where all operations are of unit processing time. Brucker (1988), developed an efficient algorithm for two jobs and m machines with the shortest processing time. Kubiak et al. (Kubiak et al., 1995), developed an efficient algorithm with an objective of minimizing the Makespan in two machines with respect to a succinct encoding of the problem instances. This was an improved form of the proposed earlier algorithms for the problem of Hefetz et al., (1982), Timkovskiy (1985) and Brucker (1988). Recently, Baptiste et al. (2004) switched the focus to minimization of total completion time and presented a shortest path optimization algorithm for scheduling jobs with release dates. They also conjectured that there always exist schedules minimizing both maximum completion time and total completion time for jobs with release dates. This is true in case of non pre-emptive schedules (Coffman, 1972) and pre-emptive schedules (Coffman et al., 2003). More recently Coffman et al. (2012) presented an efficient algorithm based on the work of Baptiste and Timkovskiy (2004) and resolved the conjecture discussed. They proved that an ideal schedule does not exist in general when pre-emptions are allowed. On the other hand, when pre-emptions are not allowed, then ideal schedules do exist for general precedence constraints.

In conclusion, despite the progress made by the recent methods described above, efficient methods could not be found for JSSP instances where $m > 3$ and $n > 3$ (Morshed, 2006). French (1982), predicts that no efficient algorithms will ever be developed for the majority of scheduling problems. This is the reason that researchers have turned their focus to mathematical formulations and enumerative approaches.

3.3.2 Mathematical Formulations

Mathematical programming has been extensively applied to the JSSP. These problems are formulated using linear programming, integer programming (Balas, 1965; Balas, 1967), mixed-integer programming (Balas, 1967; Balas, 1970), and dynamic programming (Srinivas, 1971).

The problem of solving a system of linear inequalities can be classified as a linear equation. Kantorovich (1940), developed the earliest linear programming in 1939 for finding optimal solutions. If some or all of the unknown variables are required to be integers, then the problem is called an Integer Programming (IP) or integer linear programming. (ILP) problem. If only some of the unknown variables are required to be integers, then the problem is called a mixed integer programming (MIP) problem. These are generally also NP-hard. Balas' work was focused on the configuration of the integer and mixed integer programming using computational power. Dantzig (1963) and Minoux (1986), described the simplex method and integer programming, two classes of solution technique based upon linear programming. In the early 90's Blazewicz et al. (1991) provided a survey of scheduling formulations.

Various researchers (Bowman, 1959; Giffler et al., 1960; Manne, 1960; French, 1982; Blazewicz et al., 1991) are of the view that integer programming formulations of scheduling problems are computationally infeasible and yet to achieve a breakthrough.

Due to these facts and to the combinatorial nature of the JSSP, a group of researchers began to decompose the scheduling problem into a number of sub-problems, proposing a number of techniques to solve them. Recently, the larger and hard scheduling problems have been subjected to recent advances such as parallel computing facilities in order to achieved solution quickly. Some new solution approaches are also applied to these problems. However, researchers still face difficulties in the formulation of material flow constraints as mathematical inequalities.

More recently, Akcora et al. (2005), discussed Integer Programming (IP) for job shop scheduling for varying reward structures where IP is used to optimally minimize the penalty reward based upon the completion time of the task. Pan and Chen (2005) reported on mixed Binary Integer Programming (BIP) formulations for the re-entrant (multiple visits to the machine groups) JSSP. In order to improve the solution speed of the BIP formulations, two-layer division procedures have been developed and incorporated in the BIP model of four new formulations.

3.3.3 Decomposition strategies

Davis and Jones (1988), proposed a methodology based upon the decomposition of mathematical programming, that used both 60s Benders type decomposition [republished (Benders, 2005)] and Dantzig and Wolfe's (Dantzig et al., 1960) type decompositions. The methodology was part of a closed-loop, real-time, two-level hierarchical shop floor control system (Jones and Rabelo, 1998). The top-level scheduler, i.e. the supernal, specified the earliest start time and the latest finish time for each job. The lower level scheduling modules, i.e. the infimums, would refine these limit times for each job by detailed sequencing of all operations. A multi-criteria objective function was specified that included tardiness, throughput, and process utilization costs. The decomposition was achieved by first reordering the constraints of the original problem to generate a block angular form, then transforming that block angular form into a hierarchical tree structure (Jones and Rabelo, 1998).

3.3.4 Enumerative Techniques and Lagrangian Relaxation

An exact optimization is a procedure whereby applying a mathematical analytical method determines a global optimum of the decision problem (Morshed, 2006). Two popular solution techniques for IP problems are Branch and Bound (B&B) and Lagrangian Relaxation (LR). Lagrangian Relaxation (LR), which has been used for more than 30 years, is also used to solve JSSP (Nowicki et al., 1996). IP problems can be solved by LR by omitting specific integer valued constraints and adding the corresponding costs (due to these omissions and/or relaxations) to the objective function (Jones and Rabelo, 1998). However, for large scheduling problem, both techniques the B&B and LR are computationally very expensive. However, the main focus of various researchers' for enumerative approaches to the JSSP is a B&B technique due to the fact that mathematical approaches are inadequate for the JSSP. B&B is one of the most well-known enumerative techniques (Garey et al., 1979; Baptiste and Timkovsky, 2004). Summarizing Morton and Pentico (1993), "*The* *basic idea of branching is to conceptualise the problem as a decision tree. Each decision choice point - a node - corresponds to a partial solution. For each node, there grow a number of new branches, one for each possible decision. This branching process continues until leaf nodes, that cannot branch any further, are reached. These leaf nodes are solutions to the scheduling problem*". Although efficient bounding and pruning procedures have been developed to speed up the search, this is still a very computationally intensive procedure for solving large scheduling problems. The first B&B was applied to JSSP using the disjunctive graph model by (Balas, 1970). This method only considers critical operations. Florian et al. (2003), also presented a B&B algorithm for *minimum-makespan* JSSP reached a value of 972 for the 10 × 10 (10 machines and 10 jobs) benchmark problem. Moreover, they were the first to solve the Fisher and Thompson 20 × 5 benchmark to optimality. Lageweg et al. (2012), were first to use the one-machine lower bound, hence extending the previously used lower bounds. They generated all active schedules branching over the conflict set in Giffler and Thompson's algorithm (1960). A priority rule at each node of the search tree delivers an upper bound. There is no report on the 10x10 benchmark problem. Barker and McMahon (1985), associated with each node in their enumeration tree sub-problem whose solutions are a subset to the solution set of the original problem; a complete schedule; a critical block in the schedule which is used to determine the descendant subproblems; and a lower bound on the value of the solutions of the subproblem. The lower bound is a single-machine lower bound as computed in McMahon and Florian (1975). Barker and McMahon (1985), have not achieved the optimal results for FT10 (10 x 10) and FT20 (20 x 5) benchmark problems. The reached Makespan value of 960 and 1303 for FT10 and FT20 respectively. Brucker et al. (1996), calculated different lower bounds: one machine relaxation and two jobs relaxation. They were able to improve Carlier and Pinson's (Demirkol et al., 1997) B&B algorithm and accelerate substantially and easily found an optimal schedule for the 10 × 10 problem. However, they were unable to find an optimal solution for the 20 x 5 problem within a reasonable amount of time. Perregaard and Clausen (2010), applied parallel B&B to JSSP and achieved an optimal solution to the 10 × 10 and 20 x 5 problems much less than one minute. According to Jain and Meeran (1999), in their comparative study of B&B techniques indicate that Martin's (Martin, 1996) time based oriented representation of the decision variant of JSSP performed better than other B&B techniques. Pan et al. (2006), applied B&B to a single machine sequencing problem. They based their algorithm on Carlier's (1982) B&B algorithm. The result indicated overall improvement through their algorithm. However, it was unable to find optimal solutions to larger problems. Recently, Nababan et al., (Nababan et al., 2008), applied the B&B method using the disjunctive programming approach minimum-makespan JSSPs. They tested their algorithm against selected benchmark JSSPs (FT, LA, ABZ, ORB, YN and SWV) of different size and hardness such as 50 x 15 and 50 x 20 problems in order to gauge the strength and weakness of the technology. The algorithm failed to achieve optimum results for most of the problems, but they did achieved near-optimal results comparatively in lesser computational time.

In conclusion, B&B methods are able to produce good solutions but cannot guarantee optimal solutions. Specifically, in larger problems the method suffers from memory overflow and becomes computationally expensive. To address this issue, parallel techniques can be applied. The parallel technique results for larger problems are still very disappointing.

Therefore, researchers have shifted their focus towards approximation-based techniques and have developed many techniques in recent years in order to achieve quality solutions in lesser time. In the following section a detailed review of these techniques is carried out.

3.4 Approximation Based Approaches

A method using an optimization criterion is exact if it guarantees optimality of the solutions found, otherwise it will be called approximation or *heuristic* when it empirically provides ‘good’ solutions (Maccarthy and Liu, 1993). The main advantage of these heuristic techniques is the ease with which they can be implemented in practice.

In this analysis, the two main categories of approximation technique Tailaord Algorithms (Priority Dispatch Rules (PDR) and Heuristics) and Artificial Intelligence (AI) techniques (See Figure 3.1). These techniques have been discussed in the past six decades and made their contributions in the field of scheduling. The aim of the review is to acquire knowledge and identify the gap in research. The findings from the review are also presented at the end of this section.

3.4.1 Conventional Heuristics for JSP

The JSSP is one of the hardest combinatorial optimization problems to tackle. Since JSSP is a very important everyday practical problem, it is therefore natural to look for approximation methods that produce a feasible schedule in useful time. Gen, Lin et al. (2008), have classified heuristic procedures for a JSSP into two classes:

- One-pass heuristic
- Multi-pass heuristic

In a one-pass heuristic the decision as to which job is to be loaded on a machine and when the machine will be free, is normally done with the help of PDR (Holthaus et al., 1997). An approach to the JSSP is that the main problem can be broken down into a number of sub-problems. Sub-problems are scheduled separately. There are many rules for choosing an operation from a specified subset to be scheduled next. Such methods are easy to implement and substantially reduce the computational requirements, and are very popular techniques (Morshed, 2006). These methods may not produce guaranteed optimal solutions but definitely present a feasible solution evaluated through a particular performance factor. In addition, one-pass heuristic may be used repeatedly to build more sophisticated multi-pass heuristic in order to obtain better schedules at some extra computational cost (Gen et al., 2008).

3.4.2 Heuristics Rules

In scheduling literature, the term such as scheduling rules, heuristic rules, dispatching rules, or priority rules are often used synonymously. In the past six decades there has been a substantial growth in the field of sequencing and scheduling research. The heuristic scheduling rules deal with the complexities of manufacturing under global competition are currently much sought after. These heuristics prioritize all jobs that are waiting to be processed on a resource. It is increasingly recognized that all the strengths of traditional operations research, knowledge based systems, heuristic rules, AI and sophisticated user interfaces will be necessary to build the needed system, which can fulfil the future needs. Such successful integration of approaches has not really occurred yet, despite the fact that many systems has been built by researchers that attempt some integration.

According to Panwalkar and Iskander (1976), scheduling research can be divided into two main categories: theoretical research dealing with optimizing procedure limited to the static problems and experimental research dealing with scheduling rules in static and dynamic cases. Pinedo (1998), called this experimentation research in scheduling as scheduling in practice. He categories heuristic in general purpose procedure rules along with SA, TS and GAs.

In this section a number of general purpose procedures (existing and new) that are useful in dealing with scheduling problems in practice is presented. These procedures can be easily implemented with relative ease in industrial scheduling systems. All the procedures described are heuristics that do not guarantee an optimal solution; instead they aim to find reasonably good solutions in a relatively short time. These heuristics are fairly generic and can be adopted easily to a large variety of scheduling.

3.4.2.1 Heuristics Rules and their Classification

The heuristic rules research has been active for several decades and many different rules have been studied in the literature. Giffler and Thompson (1960), have laid foundation for heuristic rules. These rules are now probably the most frequently applied heuristics for solving scheduling problems in practice because of their ease of implementation and their low time complexity (Storer et al., 1992; Gen et al., 2008). These rules can be classified in various ways. For example, according to Jackson (1957) a distinction can be made between static and dynamic rules. Static rules are not time dependent or in which priority value does not change as a function of the passage of time. They are just a function of the job and/or of the machine data, for instance, Weighted Shortest Processing Time (WSPT). Dynamic rules are time dependent. One example of a dynamic rule is the Minimum Slack (MS) first rule that orders jobs according to max $(d_j - p_j - t, 0)$, which is time dependent. This implies that at some point in time job *j* may have a higher priority than job k and at some later point in time jobs *j* and *k* may have the same priority.

A second way of classifying rules is according to the information they are based upon. For example Conway and Maxwell (1962) classification. The describe “local” heuristic rule which uses only information related to either the queue where the job is waiting or to the machine where the job is queued. Most of the traditional heuristic rules such as FIFO, SPT can be used as local rules. A global rule may use information regarding other machines, such as the processing time of the job on the next machine on its route. An example of a global rule is the LAPT rule for the two machine open shop. Moore and Wilson (1967), have classified a few dispatching rules by a two dimensional way of showing whether a specific rule is static or dynamic and whether it is local or global.

Panwalker and Iskander (1976), categorized scheduling rules as simple, combined simple, weighted priority rules and heuristic scheduling rule. The simple priority rules are usually based on information related to a specific job such as its due date, processing time, remaining number of operations, etc. Sub-classification is based on information related to (i) processing times, (ii) due dates, (iii) number of operations, (iv) costs, (v) setup times, (vi) arrival times (and random), (vii) slack (based on processing times and due dates), (viii) machines (machine-oriented rules), and (ix) miscellaneous information. The combination of simple priority rules in many cases works by dividing a queue into two or more rules apply to the same queue under different circumstances. The weighted priority indexes combine simple and combined priority rules with different weights. The heuristic scheduling rules involve a more complex consideration such as anticipated machine loading, the effect of alternate routing, scheduling alternate operation, etc. These rules are usually used in conjunction with the former three priority rule groups. In some cases a heuristic rule may involve nonmathematical aspects of human intelligence, such as inserting a job in an idle time slot by visual inspection of a schedule.

3.4.2.2 Priority Rule Based Algorithms

The algorithms of Giffler and Thompson can be considered as the common basis of all priority rule based algorithm. Giffler and Thompson (1960), have proposed two algorithms to generate schedule: active schedule and non-delay schedule generation procedures.

3.4.2.2.1 Active Schedule

Generation procedures operate with a set of schedulable operations (operations unscheduled yet with immediately scheduled predecessors) determined from constraints or precedence structure. The number of stages for a one-pass procedure is equal to the number of operations $m \times n$. At each stage, one operation is selected to add into partial schedule (in progress schedule) and the conflicts among operations are solved by priority heuristic rules. Following the notations of Baker (1974), Then

PS_t= is a partial schedule containing t scheduled operations,

S_t = is the set of schedulable operations at stage t, corresponding to a given PS_t,

σ_i = is the earliest time at which operation $i \in S_t$ could be started,

φ_i = is the earliest time at which operation $i \in S_t$ could be completed.

For a given active partial schedule, the potential start time σ_i is determined by the completion time of the direct predecessor of operation i and the latest completion time on the machine required by operation i. The larger of these two quantities is σ_i.

The potential finishing time φ_i is simply $\sigma_i + t_i$, where t_i is the processing time of operation i. The procedure to generate an active schedule works as follows:

Procedure Priority Dispatching Heuristic (active schedule generation)

Input: JSSP data

Output: a complete schedule

Step 1: Let $t = 0$ and begin with PS_t as the null partial schedule.

Initially S_t includes all operations with no predecessors.

Step 2: Determine $\varphi_t^* = min_{i \in S_t} \{\varphi i\}$ and the machine $m *$ on which

φ_t^* could be realized.

Step 3: For each operation $i \in S_t$ that requires machine $m *$ and for which

$\sigma_i < \varphi_t^*$, calculate a priority index according to a specific priority rule. Find the operations with the smallest index and add this operation to PS_t as early as possible, thus creating a new partial schedule PS_{t+1}.

Step 4: For PS_{t+1}, update the data set as follows:

i). Remove operations i from S_t

ii). Form S_{t+1} by adding the direct successor of operation j to S_t

iii). Increment t by one

Step 5: Return to step 2 until a complete schedule is generated.

3.4.2.2.2 Non-Delay Schedule

A non-delay schedules can be generated by replacing the earliest finish time with the earliest start time in step 2 and step 3 of the above algorithm as shown below:

Procedure Priority Dispatching Heuristic (non-delay schedule generation)

Input: JSSP data

Output: a complete schedule

Step 1: Let $t = 0$ and begin with PS_t as the null partial schedule.

Initially S_t includes all operations with no predecessors.

Step 2: Determine $\sigma_t^* = min_{i \in S_t} \{\sigma i\}$ and the machine $m*$ on which

σ_t^* could be realized.

Step 3: For each operation $i \in S_t$ that requires machine $m*$ and for which

$\sigma_i < \sigma_t^*$, calculate a priority index according to a specific priority rule. Find the operations with the smallest index and add this operation to PS_t as early as possible, thus creating a new partial schedule PS_{t+1}.

Step 4: For PS_{t+1}, update the data set as follows:

i). Remove operations i from S_t

ii). Form S_{t+1} by adding the direct successor of operation j to S_t

iii). Increment t by one

Step 5: Return to step 2 until a complete schedule is generated.

In theory and practice both of these scheduling types are applied to JSSPs (See Section 3.4.2).

The recent comparative study by Chang et al. (1996) evaluates the performance of 42 PDRs using a linear programming model. Their analysis indicates hat the shortest processing time (SPT) related rules consistently perform well while he longest processing time (LPT) based rules consistently perform badly.

In some cases a heuristic rule Morshed (2006), reports that in the analyses based on the Relative Deviation (RD) from optimum, the traditional heuristics achieve results extremely quickly but are of very poor quality (the RD from the optimum schedule can be as great as 74%), and in general, the solution quality degrades as the problem's dimensionality increases. However, still these rules are most popular and commonly used in scheduling in practice. The researchers also came up with different hybrid approaches including combination of different heuristic rules and addressed these problems (Maqsood et al., 2011) because no single rule shows clear dominance in order to find the best solution from a combination of different heuristics. However, more computing time is required by a combination of heuristic rules in comparison with the their simple rules. In order to overcome achieve overcome the deficiencies of the conventional heuristics two novel heuristic rules are proposed: Index Based Heuristic (IBH) and a Hybrid Heuristic (HybH). The design and development phase of theses new heuristic rules is discussed in the following section.

3.4.2.3 Heuristic Rules in Scheduling

In scheduling theory, priority rules are the most frequently applied heuristics for solving scheduling problems. Due to their ease of implementation and low time complexity these rules are also common in practice. The early work of Giffler and Thompson (1960), is considered to be the common basis of all priority rule based heuristics (Storer et al., 1992; Gen et al., 2008). They proposed two algorithms to generate active and non-delay schedule. In non-delay schedule no machine remains idle if a job is available for processing and inactive schedule no operation can be started earlier without delaying another job. Non-delay schedules is a subset of active schedules. Giffler and Thompson (1960), proposed at a tree-structured generation procedure approach. The nodes in the tree correspond to partial schedules, the arcs represent the possible choices and the leaves of the tree are the set of enumerated schedules. For a given partial schedule, the algorithm identifies all processing conflicts (i.e. Operations competing for the same machine), and at each stage these conflicts are resolved through an enumeration procedure. In contrast, heuristics resolve these conflicts with priority dispatching rules, i.e., specify a priority rule for selecting one operation among the conflicting operations (Gen et al., 2008).

Table 3. 1: Well known conventional dispatching rules used to solve JSSP

Rule	Description
SPT	Select the operation with the Shortest Processing Time (SPT)
LPT	Select an operation with Longest Processing Time (LPT)
LRT	Select the operation belonging to the job with the (Longest Remaining Processing Time) longest remaining processing time
SRT	Select the operation belonging to the job with the (Shortest Remaining Processing Time) shortest remaining processing time
LRM	Select the operation belonging to the job with the (LRT excludes the operation longest remaining processing time excluding the under consideration) operation under consideration
EDD	A schedule is developed on the basis of Earlier Due Dates (EDD) of a job. A schedule starts with a job having the EDD in the first position followed by the job having the EDD amongst the remaining unscheduled jobs. The schedule ends when all the jobs are scheduled.
FIFO	First in, first out (The operation that arrived the earliest is processed first or served first)
CR	The job sequencing priority is the ratio between the time remaining in the work remaining and is known as the critical ratio. Therefore, a job having the lowest critical ratio is sequenced first and vice-versa. Being a dynamic rule, it is mostly used in practice (Baker et al., 1960; Noor, 2007). The objectives that may be achieved by implementing this rule are the minimization of lateness and tardiness.
MP	According to this rule, jobs are sequenced according to the priority list provided by the management. That priority may be according to the importance level of a client with the management. According to Momin (1999), priority related to jobs is set in advance and provided as an input in the beginning of a schedule.

Over the years, many researchers (Baker and Dzielinski, 1960; Adam et al., 1980; Anderson et al., 1990; Holthaus et al., 1997; Dominic et al., 2004; Chiang et al., 2007) proposed heuristic dispatching rules for scheduling. Panwalkar et al. (1977), have carried out a comprehensive survey of scheduling heuristics. They presented reviewed and classified 113 dispatching rules. Chang et al. (1996), evaluates the performance of 42 dispatching rules using a linear programming. They found that the shortest processing time (SPT) rule consistently perform well and the longest processing time (LPT) consistently performs badly. In Table 3.1 most prominent heuristics used in the literature to solve real life scheduling problems are briefly described.

Rachamadugu et al. (1990), studied the performance SPT, FIFO, First in System (FIS), Least Work Remaining (LWR) and Fewest Remaining Operations (FRS) on important criteria such as the mean flow time and work-in-process inventory for FMS environment. Their comparative results indicate LWR performs better than the other rules. Grabot and Geneste, (1994), studied dispatching rules and proposed a hybrid rule by combining SPT and slack time rule. The combined rule has performed better than the single rule for all objectives. Holthaus et al. (1997), present two new dispatching rules for JSSP. These rules combine the process time and work content in the queue for the next operation on a candidate's job, by making use of additive and alternative approaches. After an extensive simulation study they concluded that It has been found that no single rule is effective in minimizing all measures of performance and recommended hybrid process time based rules in future work. Canbolat et al. (2004), combined SPT and CR rules for JSSP and used fuzzy logic in combination with these rules and named it fuzzy priority rule. They considered generalized JSSPs with 15 machines and 50 jobs, whose operation numbers are between 3 and 6. The comparative results for mean flow time, mean tardiness, WIP, total production value indicated that the hybrid methodology performed better than traditional rules. Dominic, Kaliyamoorthy et al. (2004), attempts to provide efficient dispatching rules for dynamic JSSP by combining (MWKR - FIFO and TWKR - SPT) different dispatching rules. Their results also show that combined rules perform well under most conditions. Jayamohan et al. (2004), proposed five dispatching rules for JSSP with the aim to optimize different weights or penalties for different jobs. They tested the performance of their rules using the one-way ANOVA and Duncan's multiple range tests. One the basis of statistical analysis of the absolute values, they found that PT + PW(WF + WT) performed well for minimization of the weighted mean flowtime and weighted mean tardiness, weighted flowtime and weighted tardiness of jobs followed by the W(PT + PW + ODD) rule. For the maximum and standard deviation of weighted flowtime and weighted tardiness of jobs, the WSLACK, WODD, WCOVERT and WATC rules performed well. However, the choice of a dispatching rule is influenced by shop parameters such as due-date setting and utilization levels, and hence, the shop floor manager can evaluate. Chiang and Fu (2004), proposed a dispatching rule, Enhanced Critical Ratio (ECR), which uses 'group information' to prioritize jobs. Which is a combine concepts of SPT and EDD and LRT. They applied the rule to JSSP with an objetive of minimizing the tardy rate. However, this combined principle is only applicable to non-tardy jobs in ECR. Later, Chiang and Fu (2007), proposed an extension to their work (Chiang et al., 2004). This extended ECR rule is applicable to all jobs by introducing a due date extension procedure. Demirkol et al. (1998) JSSP benchmark cases were used for result comparison with 18 existing rules. The experimental result showed the advantage of the proposed rule on multiple criteria in different shop conditions, especially when the tardy rate and the mean tardiness are the major concerns.

Kawai et al., (2005), proposed new dispatching rules based on SPT, LPT, MWKR (Most WorK Remaining), LOPN (Least OPeration Numbers) and SLACK (shortest due date). They applied three combined rules to 13 benchmark JSSPs. The first rule is the rule that combines two simple dispatch rules which are often adopted in actual production systems. These proposed rules results comparison with any single dispatch rule shows improved result.

Restrepo et al. (2008), proposed two fuzzy based dispatching strategies: fuzzy-job and fuzzy-machine is proposed and their performance is compared to two well known dispatching rules such as SPT and WEED (Weighted Earliest Due Date). On the basis of results from a total of thirty batches, they claimed that proposed fuzzy-based methodologies especially fuzzy-job shows a superior performance compared to the traditional dispatching rules considered.

According to Morshed (2006), PDRs achieve results extremely quickly, however, the quality (deviations from optimum can be as great as 74%) is very poor. Despite the deficiencies of dispatching rules, these are still most commonly applied techniques to the scheduling problems in practice, as they may provide good solutions in less time to complex problems in real-time.

One thing that can be concluded from the above discussion is that every rule is suitable for a certain condition and can achieve a certain objective, but when it comes to practice, there are a number of other related objectives too, that have to be taken into consideration. Combining these different rules or utilizing different information about jobs, most effective approaches for scheduling can be developed which can fulfil multiple objectives. Hence there is a need for development of new heuristics which overcomes the deficiencies of traditional heuristics and more importantly perform well across different size of problems.

In chapter 4, two new heuristic rules are proposed for JSSP and are compared with traditional heuristics. Literature shows that the dispatching rules are suitable as an initial solution techniques therefore the proposed techniques are also for the initial solution generation of JSSP in the proposed Hybrid Genetic Algorithm (HGA) in chapter 5.

3.4.3 Review of Artificial Intelligence

Intelligence is the ability to learn, understand, solve problems and to make decisions (Negnevitsky, 2002). Before reaching this level, human thought process started with data, information and knowledge. Data are unprocessed facts and when assembled together, it becomes information. The interpretation of that information will then become knowledge, and adding experience to it will make the person intelligent. Artificial Intelligence (AI) is a field of knowledge in computer application development that attempt to imitate the human behavior in completing tasks . AI emerged as a computer science discipline in the mid 1950s and has since produced a number of powerful tools, many of which are of practical use in engineering in order to solve difficult problems normally associated with human intelligence (Pham et al., 1999). AI covers a range of applications. Some researchers have coupled Information Technology (IT) with AI for achieving a quick response to the market (Cheng et al., 1998).

Table 3. 2: AI Function and Techniques for Manufacturing (Teti et al., 1997)

Artificial Intelligence in Manufacturing		
AI Function	**AI Techniques**	**Manufacturing Sectors**
Advice	Genetic Algorithm	Design
Control	Neural Network	Production
Learning	Fuzzy logic	Planning
Knowledge	Neuro-Fuzzy	Scheduling System
Reasoning	Simulated annealing	Control
Goal Keeping	Expert System	Assembly
Communication	Knowledge Based System	Monitoring
Decision Making	Hybrid System	Inspection
Pattern Recognition	Multi Agent	Maintenance
Self-Improvement		
Self-Maintenance		
Self-Organized		

In AI, JSSP are represented analogously instead of mapping information to a number of subjective arithmetic computations (Morshed, 2006). Noronha and Sarnia (1991) presented a survey of several existing AI techniques and applied some approaches to planning and scheduling systems. Tati et al. (1997) categorized AI into its functions, techniques and manufacturing sectors as shown in Table 3.2. From Table 3.2 it is clearly evident that AI has been applied to a wide range of problems in the manufacturing environment including scheduling.

This section describes how AI techniques have been applied to JSSP, to ascertain the recent knowledge and information relating to scheduling, with the aim of acquiring knowledge in this area for the future design of a conceptual and actual scheduling model for manufacturing environment.

In the past ten years Al techniques, such as Knowledge Base System (KBS), Neural Networks (NN), Fuzzy Logic (FL), and Genetic Algorithms (GAs) have been extensively applied to scheduling problems. Three main methodologies (FL, NN and GA) are reviewed (basic and detailed) and analyzed in the following section. Many other Al techniques, such as Expert Systems (ES) / Knowledge Based Systems (KBS), Simulated Annealing (SA), Case Based Reasoning (CBR), Frame Based Systems (FBS), are also being applied to scheduling problems. However, the effect of these techniques applied to JSSP is limited.

3.4.3.1 Expert Systems (ES) /Knowledge Based System (KBS)

KBS or ES is a branch of AI. These are computer programs that require a specialized expertise without the assistance of a common sense knowledge for the day-to-day operations of all areas of industries (Rich et al., 1991). According to Awad (1995) "knowledge is understanding gained through experience or study which is the accumulation of facts, procedural rules, or heuristics". It requires familiarization in dealing with something in order for a person to perform a task. Andersson (2008) added that the definition of knowledge has been debated long before it was used in engineering back in the era of the ancient Greek Plato as *"justified true belief"*. According to Andersson, knowledge can be arranged in the hierarchical form of a pyramid model as shown in Figure 3.2.

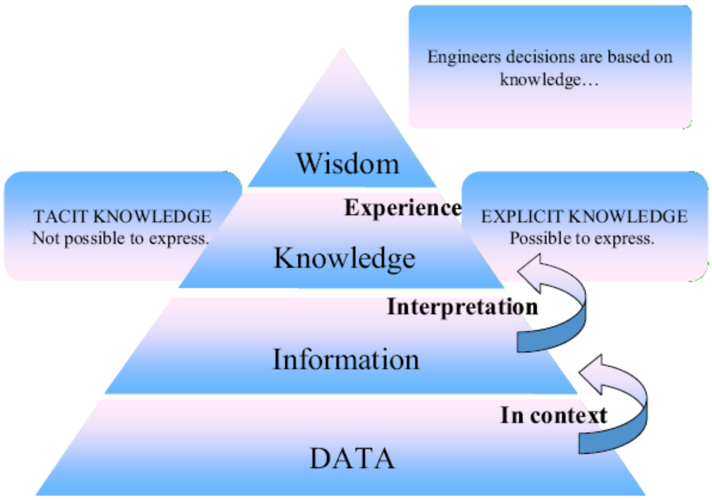


Figure 3. 2: Pyramid model, the hierarchy of data, information, knowledge and wisdom (Andersson, 2008)

Data are unprocessed facts, are static and have no meaning (for example, "the building is 800 meter heigh"). This data becomes informed after a certain assembling of facts process such as "it took two years to build the building". This information when interpreted by a person will become knowledge and adding experience to it, the person will have wisdom. As for the above example, it is not profitable to have this kind of building if the company wants to rent the building in one year time. In addition, tacit knowledge or explicit knowledge is a kind of knowledge that is difficult to express (Lintern, 2006) and is normally known to an individual, such as the painting skills. On the other hand, explicit knowledge is the knowledge that you are able to express, such as in manuals and procedures. Therefore, it is important to transform tacit knowledge into explicit knowledge in order to accommodate the transfer of knowledge, especially in building an expert system.

According to Mohamed and Khan (2011), an ES makes a decision in a narrow domain on the basis of input information (factual and domain knowledge) in a similar way to the human experts who decide on the basis of experience, observation, knowledge and reasoning. Table 3.3 gives a brief comparison of human experts, ES and conventional programs.

Table 3. 3: Comparison of Human Experts, ES and other Computing Programs [Hussain 1998]

Human Experts	**Expert System**	**Conventional programs**
Knowledge exists in a compiled form.	Knowledge and compilation are separated.	Knowledge exists in the control structure for compilation.
Use knowledge in the form of a rule of thumb or heuristics in the given domain for a solution of the given problem.	Knowledge, in the form of rules is processed for reasoning in the given domain for s solution of the problem.	Data is processed through a series of well-defined algorithms to solve general numerical problem.
Capable of explaining the reasoning	Limited explanation of how a particular rule was fired and why a particular data was used.	Is unable to explain anything.
Can use inexact reasoning and deal with incomplete and fuzzy data.	The limited capabilities of inexact reasoning and can deal with uncertain and fuzzy data	Cannot deal with incomplete data and/or reasoning.
The quality of a solution can be better by learning from experience over a period of time, but the process is not cost effective and is inefficient.	The knowledge base can be easily broadened by the addition of new rules with the passage of time.	The quality of a solution can be improved over a period of time by rewriting the code and knowledge and the data, which becomes difficult.
Work with a parallel thinking mechanism and thus the conclusion is more realistic.	Cannot work with a parallel thinking Mechanism and thus solutions are not real world solutions.	Cannot think.

O'Kane (2000) used ES to provide decision making and control across FMS and recommended shift of emphasis from predictive scheduling to reactive scheduling. In a dynamic scheduling environment, Priore et al. (2001) used ES in the decision making stage i.e. select the most appropriate dispatching rule at each moment in time. They also reviewed man-machine learning-based scheduling approaches. Metaxiotis et al. (2002) used ES for to select the most appropriate algorithm from a library of many candidate algorithms for scheduling. Benavides and Prado (2002) used ES for detailed scheduling problems. They used ES as a support tool for obtaining static system and optimize solutions of complex problems particularly when the developed system operates together with MRP II. Varela et al. (2003) presented the ES evolutionary strategy with bottleneck for scheduling problems where they introduce specific knowledge in the initial solution. Soyuer et al. (2007) developed ES for a scheduling system that is realistic and applicable to real life situations. They considered Job specification, machine competence, due date, earliest completion and minimum setup time factors. The algorithm achieves applicability and optimality of the solution. Recently, in job shop environment, the reduction of the standard deviation generated through the routing of production orders in manufacturing resources, Zattar et al. (2008) presented an ES which obtained this objective through the suggestion of the best machining route for each job order in a simulation, based upon historic simulation data (base of facts) and a set of rules.

3.4.3.2 Fuzzy Logic (FL)

Fuzzy Logic (FL) was introduced in 1930 by Jan Lukasiewicz, a Polish logician who studied the mathematical representation of fuzziness based on such terms as tall, old and hot (Negnevitsky, 2002). He introduced the extended range of truth values version of logic to all real numbers in the interval between 0 and 1, contrary to the classical version which operates with only two values, 1 (true) and 0 (false), and used a number to represent a possibility that a given statement was true or false. This work led to an inexact reasoning technique, often called the possibility theory. Lukasiewicz's main contribution was the presentation of a simple fuzzy set (in his paper appendix), which outlined the operation of the fuzzy set operations [re-published by Pogorzelski, (1965)].

Black (1937), argued that a continuum implies degrees. “Imagine”, he said, “a line of countless chairs.” At one end is a Chippendale chair (a famous furniture designer in the mid 18th century). Next to it is a near-Chippendale, in fact indistinguishable from the first item. Succeeding ‘chairs’ are less and less chair-like, until the line ends with a log. When does a chair become a log? The concept chair does not permit us to draw a clear line distinguishing a chair from not a chair. Black also stated that if a continuum is discrete, a number could be allocated to each element. This number would indicate a degree. But the question is, degree of what? Black used the number to show the percentage of people who would call an element in a line of ‘chairs’ a chair; in other words, he accepted vagueness as a matter of probability (Negnevitsky, 2002).

Zadeh (1965), rediscovered, identified, explored, and promoted fuzziness. Apart from a formal mathematical logic, he introduced a concept of applying natural language terms. This new logic of representation and manipulating fuzzy terms was called FL (Negnevitsky, 2002). According to Zedan (1965), “*Fuzzy Logic is determined as a set of mathematical principles for knowledge representation based on degrees of membership rather than that on crisp membership of classical binary logic*”.

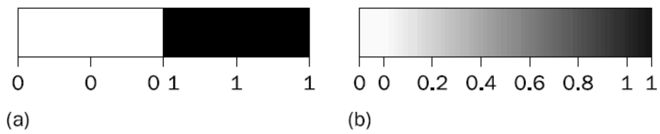


Figure 3. 3: Range of values in (a) Boolean and (b) Fuzzy Logic

Unlike two-valued Boolean logic, fuzzy logic is multi-valued, which deals with the degrees of membership and uses the continuum of logical values between 0 (completely false) and 1 (completely true). Instead of just black and white, it employs the spectrum of colors, accepting that things can be partly true and partly false at the same time, as shown in Figure 3.3.

Recently, Fuzzy scheduling models have attracted an increased interest among the scheduling research community (Petrovic et al., 2008) (Słowiński et al., 2000; Petrovic et al., 2008). Inexact scheduling parameters have been represented as fuzzy numbers and operations on them have involved fuzzy arithmetic. Parameters that are most often represented as fuzzy numbers are processing times and due dates (Ishibuchi et al., 1994; Ishii et al., 1995; Kuroda et al., 1996). However, there are models that deal with the Fuzzy job precedence relation and breakdown parameters of scheduling by employing fuzzy sets (Luh et al., 1994). Fuzzy sets have also been used to represent flexible constraints, the violation of which has to be minimized. Most often, the models included flexible temporary constraints, where the best schedule requested the least relaxation of release dates or due date constraints (Fargier, 1996). The discussed cases for scheduling were simple one-machine cases, while few authors in literature attempted complex cases such as job shop.

Sakawa and Kubota (2000) considered the fuzzy nature of the data in real-world scheduling problems with fuzzy processing time and fuzzy due-date. They formulated a multi-objective JSSP for six jobs, six machines and ten jobs, ten machines. This formulation was on the basis of the agreement index of fuzzy due-date and fuzzy completion time and these objectives not only maximize the minimum agreement index but also maximize the average agreement index and minimize the maximum fuzzy completion time. Chan et al. (2003) presented a real-time fuzzy expert system to scheduling parts in a flexible manufacturing system (FMS). They applied FL to improve the system performance by considering multiple performance measures in scheduling rules by focusing on characteristics of the system's status, instead of parts, to assign priorities to the parts waiting to be processed. Canbolat and Gundogar (2004) presented a fuzzy priority rule (FPR) for JSSP. They used fuzzy logic to combine SPT, CR priority rules, and next machine's load (NML) in order to satisfy all objectives. They used fuzzy logic to calculate a priority value by considering SPT, CR, and NML. The FPR select job with the highest priority value of process. Chang et al. (2006) presents a fuzzy extension of the economic lot-size scheduling problem (ELSP) for fuzzy demands of their work. Also, a genetic algorithm governed by the fuzzy total cost function and fuzzy feasibility constraints is designed and assists the ELSP in search for the optimal or near-optimal solution of the binary variables.

Some recent successful implementation to operational or shop scheduling problems, which got satisfactory results and feasible solutions, follow, Petrovic, Fayad *et al*. (2008) used fuzzy rule based system logic for determining the lot-sizes of jobs in a real-world JSSP in the presence of uncertainty, using the following premise variables: size of the job, the static slack of the job, the workload on the shop floor and the priority of the job. The determined lots' sizes were input to a fuzzy multi-objective genetic algorithm for the JSSP. They modelled the imprecise jobs' processing times and due dates using fuzzy sets, with objectives as average weighted tardiness of jobs, the number of tardy jobs, the total setup time, the total idle time of machines and the total flow time of jobs for quality measure of the generated schedules. Mehrabad et al. (2009) successfully applied FL to a single machine scheduling problem. They aim to improve it to a real-world application. They defined processing times and due dates of jobs are defined as fuzzy numbers and considered two objectives: average tardiness and number of tardy jobs for minimization. Lai and Wu (2009) obtained feasible results for flow shop scheduling problems with fuzzy processing times. They present a computational procedure to obtain the approximated membership function of the fuzzy completion time. With a ranking concept among fuzzy numbers, an objective to minimize the fuzzy makespan and total weighted fuzzy completion, the best schedule was searched; the processing times were assumed as fuzzy numbers. Zhang and Wu (2010) obtained a near-optimal solution using decomposition based hybrid optimization algorithm for a large-scale JSSP, with the objective of minimizing the total weighted tardiness. They constructed a fuzzy inference system to calculate the jobs' bottleneck characteristic values that are used to guide the process of sub-problem solving in an immune mechanism in order to promote the optimum efficiency. Li et al. (2010) consider a single machine due date assignment scheduling problem with uncertain processing times and general precedence constraint among the jobs. They assumed processing times of the jobs as fuzzy numbers and presented the precedence constraint is a tree or a collection of trees (Petrovic et al., 2008).

Scheduling problem with uncertain processing times and general precedence constraint among the jobs. They assumed processing times of the jobs as fuzzy numbers and presented the precedence constraint is a tree or a collection of trees.

3.4.3.3 Simulated Annealing

In the early 1980s, Simulating Annealing (SA) was one of the popular algorithms used to tackle the combinational optimization problems. The ideas that form the basis of SA was Metropolis et al.'s (1953), an early work. He developed an algorithm to simulate the cooling process of metals in a heat bath known as annealing. In annealing or cooling process of heated metals, the atoms align in an ordered manner and form a crystal, which is the state of minimum energy in the system or the global minimum. This method was independently described by Kirkpatrick, Gelatt et al. (1983). To adopt this analogy, SA uses temperature as the control parameter that is decreased by iterations until it gets close to zero (Teti and Kumara, 1997). Specifically, SA is a stochastic local search technique based upon principles of physics.

SA is a random search technique i.e. starts with initial random population and proceed until optimization is achieved. By comparison, both SA and GA start with an initial random population and proceed until optimization is achieved. SA simulates the metal cooling and freezing processes, whereas GA is based on the genetic processes. According to Bureerat and Limtragool (2008) the searching procedure of GA starts with an initial solution (known as a parent), which would be mutated during the process, leading to a set of children. Only the best offspring would then become a candidate for challenging its own parent. For minimization purposes, the parent would be replaced by the offspring if it had a lower objective value than that of the parent. The offspring, however, even though having a higher objective function value than that of its parent could still challenge its parent, provided that the Boltzmann probability accepted it. The best solutions and the parent are initially the same but they can be different during the optimization process. The iteration stops when the system is frozen or has reached the crystallized state.

The SA application is also being studied to solve the manufacturing cell formation problems (Wu et al., 2008). In cellular manufacturing, machines are allocated to process one or more part components so that each cell is operated independently to minimize part movement. However, the problem with this type of manufacturing is to achieve the timing optimization (Wu et al., 2008). The allocation of machines in the cellular manufacturing normally depends on the parts' assignments. Therefore, the SA approach uses the strategy of searching for better neighborhood solutions to improve the current solution aiming to optimize quality. A neighborhood solution is defined as the possible movements among the cells.

Vanlaarhooven et al. (1992) describe an approximation algorithm (based on SA) for the problem of finding the minimum makespan in a JSSP. The generalization involves the acceptance of cost-increasing transitions with a nonzero probability to avoid getting stuck in local minima. The algorithm asymptotically converges in probability to a globally minimal solution. They compared the computational experiments and concluded that SA found shorter Makespan solutions than Adams, Balas et al. (1998), shifting bottleneck procedure at a higher computational cost. Yamada, Rosen et al. (1994), applied SA to JSSP. They used permutation procedure to generate new schedules from existing schedules. The SA probabilistically chooses, accepts or rejects the new schedule, allowing importance sampling search over the JSSP space. Their experimental results show that SA can find near optimal schedules and often outperforms previous SA adjacent swapping approach. Sadeh et al. (1996), applied SA to JSSP with tardiness and inventory costs. The algorithm shows significant increase in schedule quality reduced scheduled cost by 28% over the combination of thirty-nine traditional dispatching rules and release policies, though at the expense of intense computational efforts. The SA for some well-known analytical results on the convergence of simulated annealing (SA) do not hold on the application to the JSSP. To overcome this issue Kolonko (1999), proposed a new SAGen approach that uses a small population of SA runs in a GA framework, and allow reheating in SA through an adaptive temperature control. They compared their results with Vanlaarhoven, Aarts et al. (1992), and found improvement in most of the considered JSSP benchmark problems. Ponnambalam et al. (1999) applied SA to JSSP with an objective of minimization of Makespan. Later Ponnambalam et al. (1999) SA (adjacent swapping: pairwise exchange, insertion, and random insertion based) to JSSP with an objective of minimization of Makespan. They compared their results and claimed that SA often gives better results with random insertion perturbation scheme (RIPS). Cruz-Chavez et al. (2004) proposed SAR (SA with restart) for the JSSP, which restarts with a new value every time the previous algorithm finishes with the condition that the initial value of the makespan of the schedule would not surpass a previously established upper bound. They considered FT10 and LA40 problem from The experimentation and compared their result with literature. The author claimed that for both the problem, the SAR is starting with the best schedules that do not surpass a UB, improves the solution considerably. Steinhofel et al. (2003) present a solution to the JSSP using a local search algorithm based on a simulated annealing method with the aim to optimize the C_{max} criterion. They found that a non-uniform sampling SA performed better than a uniform sampling SA when tested on JSSPs.

They attempted to experimentally analyze the energy landscape in order to avoid the local minima and find the optimum solution. They used what they called a non-uniform neighborhood', in which more than one swap was allowed to create neighborhoods. They found that the non-uniform neighborhood' performs better in finding the optimum than the uniform neighborhood' in which only one swap is allowed to find the next neighbor. The authors report some moderate success in improving the known upper bounds. However, their algorithm did not provide any new optimal solution. Varadharjan et al. (2005) have discussed the Multi-Objective SA (MOSA) algorithm for flow shop scheduling problems to minimize the makespan and the total flow time. Two varieties of the proposed algorithm, called MOSA-I and MOSA-II, with different parameter settings with respect to temperature and epoch length, are considered in the performance evaluation of algorithms. Bozejko et al. (2009) presented two parallel SA (adjacent swap based) for the JSSP with the sum of job completion times criterion.

Most of the SA literature is a hybrid approach by using SA with some other techniques. Recently, Zhang and Wu (2010) presented a hybrid simulated annealing algorithm based on a novel immune mechanism for JSSP and Jamili et al. (2011) also presented a hybrid approach based on SA and particle swarm optimization. They conclude that the hybrid algorithms are more efficient and produce better results compared to the conventional SA.

3.4.3.4 Artificial Neural Networks

An ANN is a reasoning based computational model of human brain (Teti and Kumara, 1997; Negnevitsky, 2002). The ANN consists of a number of very simple and highly interconnected processors called neurons, operate in parallel and are analogous to the biological neurons in the brain (Noor, 2007). The neurons are connected by weighted links that pass signals from one neuron to the other. A neuron produces only one signal as an output, although it receives more than one input signal. A neuron which receives signals from its input links, computes a new activation level and sends it as an output signal which can be either a final solution or an input to another neuron (Negnevitsky, 2002). Figure 3.4 represents connections of a typical ANN.

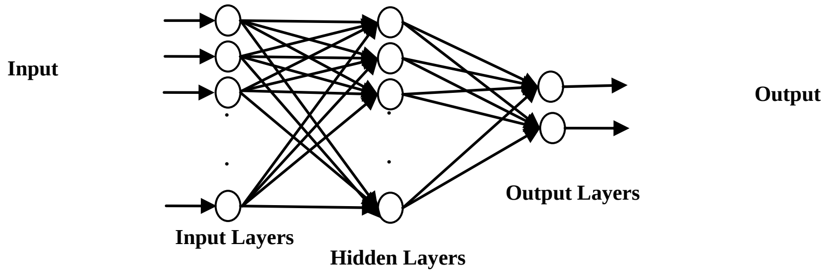


Figure 3. 4: Architecture of a typical artificial neural network

McCulloch et al. (1943 &1990) proposed a very simple idea that is still the basis for most artificial neural networks. The neuron computes the weighted sum of the input signals and compares the result with a threshold value θ. If the net input is less than the threshold, the neuron output is -1, but if the net input is greater than or equal to the threshold, the neuron becomes activated and its output attains a value of $+1$.

ANNs are preferred over time consuming simulation approaches in some cases of manufacturing systems design. Manufacturing scheduling systems are not completely exposed to the ANN learning, capturing and predicting complex relationships between input and output variables' qualities (Akyol et al., 2008).

Negnevitsky (2002) described main ANN architectures in his book: Error correcting networks or Multilayer neural networks (Back-Propagation and Forward Propagation), Hopfield network, Bidirectional Meeran associative memory network and self-organising NN. Wang and Brunn (Wang et al., 1995) also provided analysis and review of these methods for JSSP. Jain and Meeran (Jain and Meeran, 1998) presented investigation and review of Back Error Propagation (BEP). They also introduced modified BEP Neural Networks (BEPNN) and applied it to JSSPs. Recently, Akyol et al. (2007) also presented an extensive review of these techniques and their characteristics.

Yang and Wang (2000) developed the first Constraint-Adaptive Neural Network (CSANN). They applied CSANN to JSSP. The author claimed that CSANN performed well when the expected makespan is suitably chosen. And when the specification of the expected makespan is too loose, the feasible solution searched may be not good enough, and when too tight or shorter than the optimum, the feasible solution cannot be obtained. Yang and Wang (2001) extended their work (Yang and Wang, 2000) of CSANN. They applied a combine CSANN with a new heuristic to JSSP for obtaining a non-delay schedule. Chen and Huang (2001) proposed a competitive network (fuzzy Hopfield neural network clustering technique) for JSSP. They referred and considered their previous work and considered the same problem (Cheng et al., 1999). Simulation results illustrate that imposing the fuzzy Hopfield neural network onto the proposed energy function provides an appropriate approach to solving JSSP. However, this method is not good for larger problems. Sabuncuoglu and Touhami (2002) examined robustness of using ANN as a simulation BPNs based meta-model to estimate manufacturing system performances. The JSSP model was simulated and the effects of various performance factors were studied. Their study shows that the meta-models were successful in discriminating between dispatching policies in this same context and the success of meta-modelling with NN depends on the combination of the system characteristics and the error assessment criteria, as well as the purpose of simulation applications. Akyol (2004), and Solimanpur et al. (2004), successfully applied NN to FSSP with minimization of Makespan as an objective function.

Shugang et al. (2005) applied hybrid Neuro-Fuzzy approach to JSSP with minimizing total weighted quadratic tardiness of all jobs as an objective. They used GA to train the hybrid network (EANN).

Zhao et al. (2005) applied hybrid ANN-GA approach to JSSP with minimization of Makespan as a criterion. They developed a Constraint Neural Network (CNN) to represent processing restriction resulting in higher speed and efficient algorithm compared to previous hybrid systems. To improve the accuracy of the fluctuation smoothing rules Chen (2009) proposed a hybrid fuzzy c-means (FCM)–BPNs for JSSP. They modified the well-known fluctuation smoothing rules with some innovative treatments. Their algorithm outperforms some previous existing approaches in reducing the average cycle time and cycle time variation at the same time.

Recently Yang, Wang et al. (2010), (Yang et al., 2010) applied CSANN-II, an extension of their original CSANN algorithm discussed earlier in this section, to JSSP. In CSANN-II, the topology corresponding to the resource constraints is simplified according to the online resource constraint satisfaction situation when it is running via a simple sorting algorithm. Consequently, CSANN-II's computational time per schedule is reduced over the original CSANN model. The algorithm outperforms three classical heuristic algorithms, which are widely used as the fundamental tools for advanced JSSP systems (Hart et al., 2005).

In essence, as the majority of NN methods are combined with other methods. The ANN alone is mainly successful in small problem sizes (Akyol and Bayhan, 2007; Maqsood et al., 2010). However, for optimization of larger problems and even in hybrid approaches NN usually not performed very well and but instead outperformed by other techniques. Consequently, NNs are not considered to be competitive with the best heuristics for larger optimization problems.

3.4.3.5 Evolutionary Computation

In operations research, Evolutionary Computation (EC) is a subfield of AI (more particularly computational intelligence) that involves combinatorial optimization problems. EC deals with Genetic Algorithms (GA), Evolution Strategies and Genetic Programming (GP). This approach is based on the computational models of natural selection and genetics. The following steps are followed in evolutionary computations (Goldberg, 1989; Negnevitsky, 2002).

- Create a population of individuals
- Evaluate their fitness
- Generate new population by applying genetic operators
- Repeat this process a number of times

In the following section, a general discussion about how the GA works and its application in the field of manufacturing scheduling is discussed.

3.4.3.5.1 Genetic Algorithm

Genetic Algorithms are a class of stochastic search algorithms based on biological evolution (Negnevitsky, 2002). A GA is inspired by Darwin's theory of evolution (Yeh et al., 2007) and refers to a model introduced and investigated by Holland (1975) and his students (e.g. DeJong, 1975), and later, other researchers (Goldberg, 1989; Davis, 1991) adapted these algorithms for designing solution methods for optimization problems. GAs are still one of the most popular optimization tools and are capable of being applied to an extremely wide range of problems. Jong (1993) in a paper "*GAs are NOT Function Optimizers*", tried to prove that GAs are potentially far more than just a robust method for estimating a series of unknown parameters within a model of a physical system. In literature, some researchers used GAs from an experimental perspective and some focused on GAs as an optimization tool. Recently, GAs have been preferred over traditional methods for optimization problems due to their proven capabilities of solving many large problems. Coley (2005) has listed a range of practical optimization problems in his book *"An Introduction to GAs for Scientists and Engineers"*, to which GAs have been successfully applied. A typical GA may consist of the following:

a) A population of solution 'guesses' to the problem. Rather than starting from a single point (or guess), GAs are initialized with a population of guesses. The population is normally random and spread throughout the search space. The initial guesses (or chromosomes) are held as binary encodings (or strings) of the true variables, although an increasing number of GAs use "real valued" (base-10) encodings.

b) A procedure to calculate the goodness or badness of individual solutions within the population. This is known as a selection procedure. For the selection of chromosomes, many methods are used such as the roulette wheel selection, tournament selection, rank selection and steady state selection.

c) A way of mixing fragments of the better solutions to form a better new solution.

d) A mutation operator to avoid permanent loss of diversity within the solutions.

As discussed earlier, GAs use a stochastic search method; the fitness of a population may remain stable for a number of generations (or iteration) before a superior chromosome appears. In such cases, the use of conventional terminating criteria is problematic (Negnevitsky, 2002). Therefore, it is common practice to terminate a GA after a specified number of iterations. After termination, the chromosomes are examined for the best chromosome in the population and the GA is restarted if no satisfactory solution is found.

3.4.3.5.2 Encoding Problem

One basic feature of genetic algorithms is that it works on the coding space (chromosomes) and on the solution space (Evaluation) as shown in Fig. 3.5. Natural selection is the link between chromosomes and the performance of their decoded solutions (Cheng et al., 1996). In Holland's work, encoding is carried out using binary strings, since then various non-string encoding techniques have been created for JSSP, to which classical GA was difficult to apply directly (Ying and Liao 2004).

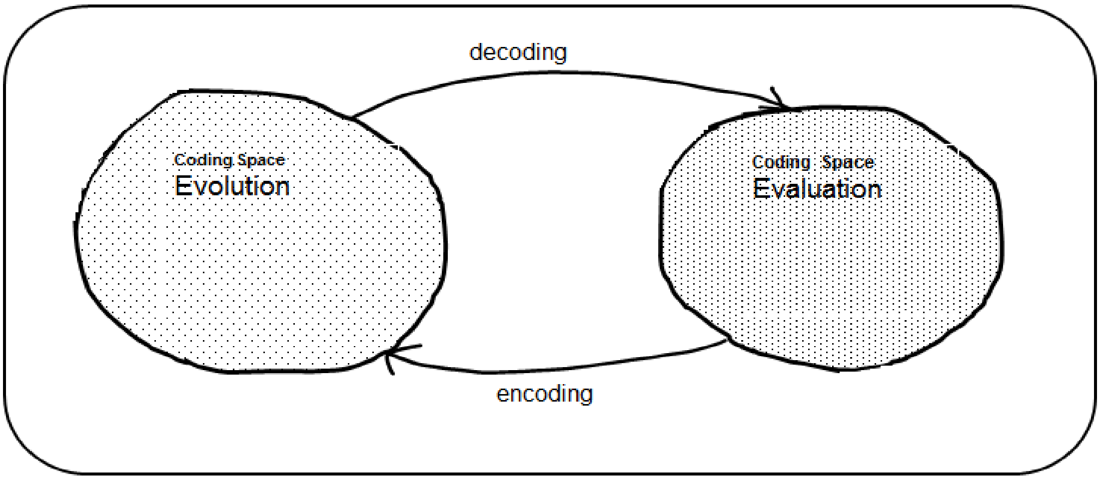


Figure 3. 5: Coding spaces and solution spaces (Chang et al., 1996)

Encoding of solutions to chromosomes is considered to be a key feature in GAs. There are three critical issues emerged concerned with the encoding and decoding between chromosomes and solutions in non-string coding approach as follows:

- The feasibility of a chromosome
- The legality of a chromosome

- The uniqueness of mapping.

The feasibility means that whether or not a solution decoded from a chromosome lies in the feasible region of a given problem. The legality means that whether or not a chromosome represents a solution to a given problem as shown in Figure 3.6.

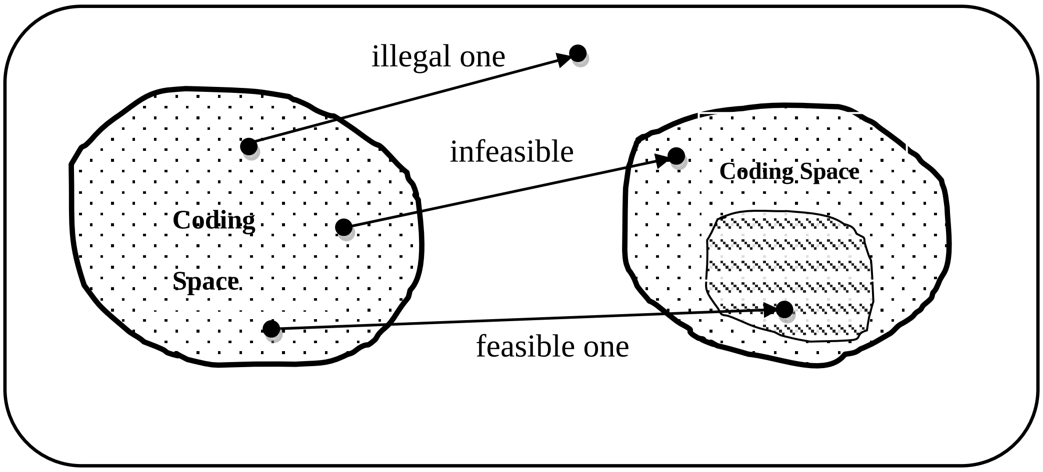


Figure 3. 6: Feasibility and Legality (Chang et al., 1996)

The feasible region can be represented as a system of equalities or inequalities (linear or non-linear). In JSSP, the infeasibility of chromosomes is due to the precedence constraints and for which there is no better representation with a system of inequalities. Therefore, it also makes difficult to apply the penalty approach, which are not easily applied to handle such kind of constraints (Cheng et al., 1996). In JSSPs, the optimum typically occurs at the boundary between feasible and infeasible area. The penalty approach forces genetic search to approach to optimum from both feasible and infeasible regions.

The problem-specific encoding techniques normally causes the illegality of chromosomes, and such encodings usually yield illegal offspring. Hence, the chromosome cannot be decoded to a solution or evaluated. The penalty techniques are also inapplicable in such situation. Orvosh and Davis (1994) have shown that it is relatively easy to repair an infeasible or illegal chromosome and these repair strategies which converts an illegal chromosome to a legal one, surpass other strategies such as rejecting strategy or penalizing strategy.

3.4.3.5.3 Genetic Representation of JSSP

In development of GA for JSSP, representation of solutions together with problem specific genetic operations are the key steps. Cheng et al. (1996), have classified representation scheme for JSSP and are still in used. Recently, Gen, Lin et al.(2008), discussed these nine schemes as shown in Table 3.4.

These representations can be classified into the following two basic encoding approaches: Direct approach and Indirect approach. In the direct approach, a schedule (the solution of JSSP) is encoded into a chromosome and GAs are used to evolve those chromosomes to find a better schedule. In the indirect approach, such as priority rule-based representation, a sequence of dispatching rules for job assignment, but not a schedule, is encoded into a chromosome and GAs are used to evolve those chromosomes to find a better sequence of dispatching rules. A schedule is than constructed through the sequence of dispatching rules. Table 3.4, shows these schemes and research work related to each scheme in since their use. Moreover, new representation schemes and hybrid schemes, which are not listed in the table, are also discussed in this section.

Table 3. 4: Classification of representation (Cheng et al., 1996) and recent research in JSSP

Approach	**Representation Strategy**	**Literature**
Direct	Operation-based	(Fang et al., 1993), (Gen et al., 1994), (Liaw, 2000), (Wang et al., 2001), (Zhou et al., 2001; Zhou et al., 2004), (Park et al., 2003),(Pezzella et al., 2008),
	Job-based	(Bierwirth, 1995), (Bierwirth et al., 1996), (Ono et al., 1996), (Shi, 1997), (Braune et al., 2005), (Chang et al., 2006), (Noor and Khan, 2007), (Tariq, 2008), (Pan and Huang, 2009), (Tseng et al., 2009),(Maqsood et al., 2011)
	Job pair relation-based	(Yamada et al., 1991), (Pesch, 1993)
	Completion time-based	(Yamada and Nakano, 1992)
	Random keys	(Bean, 1994), (Norman et al., 1996) (Rothlauf et al., 2002), Goncalves et al. (2005), (Snyder et al., 2006), (Samanlioglu et al., 2008), (Chaudhry et al., 2008), (Kachitvichyanukul et al., 2011)
Indirect	Preference list-based	(Davis, 1985), (Falkenauer et al., 1991), (Croce et al., 1995), (Kobayashi et al., 1995)
	Priority rule-based	(Dorndorf et al., 1995), (Wu and Zhao, 2000), (Gao et al., 2007), (Gao et al., 2009)
	Disjunctive graph-based	(Tamaki et al., 1992)
	Machine-based	(Dorndorf and Pesch, 1995)

Biegel et al. (1990) successfully applied GAs to the JSSP (n jobs, 2 machines and n jobs, m machines) with a specific goal is to increase the throughput. They used string type representation. Their research indicates that GAs could be the appropriate tool to bring JSSP into a manageable arena. They identified future areas in the field and GAs limitations. The earliest direct approach, binary encoding based representation of precedence relationships of operations on the same machine was presented by Yamada and Nakano (1991). The tested their algorithm against FT06, FT10 and FT20 benchmark JSSPs. Later on they presented an improved version (Yamada and Nakano, 1992) of their work based on Giffler and Thompson's (Giffler and Thompson, 1960) algorithm.They improved their work (Yamada and Nakano, 1992). They successfully found optimal for FT06 and FT10. However, FT20 results were 1184 compared to optimum result of 1165 at that time found by McMahon (1975). Fang et al. (1993) used operations-based representation of their Evolving Heuristic Choice (EHC). They applied EHC to open shop problem for minimization of Makespan, which performed better than Tabu Search on benchmark problems. Pesch (1993) combined a local search heuristics with GA (with job-pair based representation) to control sub-problem selections in his decomposition approach. The sub-problems are solved by a constraint propagation approach, which finds good solutions by fixing arc directions to FT problems.

The operation-based representation encodes a schedule as a sequence of operations and each gene stands for one operation. There are two possible ways to name each operation. One natural way is to use natural numbers to name each operation, like the permutation representation for TSP. Unfortunately because of the existence of the precedence constraints, not all the permutations of natural numbers define feasible schedules (Gen et al., 2008). Gen et al. (1994) proposed an alternative: they name all operations for a job with the same symbol and then interpret them according to the order of occurrence in the sequence for a given chromosome. They used GA in combination with B&B methods for JSSP and successfully test (achieved optimum) their algorithm against FT06, FT10 and FT20 benchmark problem. Their FT20 result was 1175 with a relative deviation of 0.0085. However, the solution in GA approach was the best at that time.

Bierwirth (1995) introduced job-based representation technique - mathematically known as "permutation with repetition" which was used to sequence the tasks of a JSSP on a number of machines related to the technological machine order of jobs. This single chromosome representation produces operation sequences with no illegality issues. As a consequence of the representation scheme the new crossover operator preserving the absolute order of a permutation. He applied his GA to FT, LA and ABZ problems. The results were encouraging and found near optimal for all problems and optimum for FT06 and LA30 problem. Later Bierwirth et al. (1996) analysis of three crossover operators and the behaviour was similar to the well known Order-Crossover for simple permutation schemes. They algorithm this time found more optimal solution of benchmark problems.

Ono (1996) propose a GA for JSSP and used a job sequence matrix. This Job based Order Crossover (JOX), preserve characteristics, the order of each job on all machines between parents and their children, and take account of the dependency among machines. The JOX's offspring are not always feasible, therefore they propose a technique to transform them into active schedules by using the Giffler and Thompson method (Giffler and Thompson, 1960). Recently (Tariq, 2008) has also used the same approach to JSSP. Shi (1997) presented a crossover technique which randomly divided an arbitrarily chosen mate into two subsets. This resulted offspring from this job based representation overcame the the problem of infeasibility in genetic generation. The applied to FT10 and FT20 benchmark problem and found optimum results. Recently, Noor (2007) applied using job-based developed by (Braun et al., 2004) in for GA and applied to JSSP.

Bean (1994) first introduced random key representation for GAs. With this technique, genetic operations can produce feasible offspring without creating additional overhead for a wide variety of sequencing and optimization problems. Later, in their work (Norman and Bean, 1996) they successfully generalized the approach to the JSSP. Goncalves et al. (2005) used a random key alphabet and an evolutionary strategy identical to the one proposed by Bean (1994) in their HGA for the JSSP. The schedules are constructed using a priority rules in which the priorities are defined by the genetic algorithm. Schedules are constructed using a procedure that generates parameterized active schedules. After a schedule is obtained, a local search heuristic is applied to improve the solution. The approach is tested on a set of standard instances taken from the literature and compared with other approaches.

The first indirect approach, preference list-based representation was originally proposed by Davis (Davis, 1985) for a kind of scheduling problem. Falkenauer and Bouffoix (1991), used it for JSSP with release times and due dates. Croce et al. (1995) used this representation and applied to classical JSSP. They argued that the deduction procedure only generates non-delay schedules, and cannot guarantee it that the optimal solution is encoded. They gave a rather complex lookahead evaluation procedure to help the deduction procedure get an active schedule. However, Chen et al. (1996) referred to Baker (Baker, 1974) and claimed that the argument is not true, because when generating a non-delay schedule, the critical machine must first be identified. A critical machine is one which can start earlier and then select an operation which can be processed earliest on the critical machine. Kobayashi et al. (1995) also adopted a similar kind of representation in their work. The difference with the above method is that they use Gifller and Thompson's heuristic to decode a chromosome into a schedule.

Dorndorf and Pesch (1995) proposed a priority rule-based GA. They encoded the chromosome as a sequence of dispatching rules for job assignment and a schedule is constructed with a priority dispatching heuristic based on the sequence of dispatching rules. GAs were used to evolve those chromosomes to find out a better sequence of dispatching rules. Dispatching rules are most frequently applied to solve JSSP.

Tamaki and Nishikawa (1992) proposed a disjunctive graph-based representation in their algorithm. Its resemblance is with job pair-based representation. Dorndorf and Pesch (1995) also proposed a machine-based genetic algorithm, where a chromosome is encoded as a sequence of machines and a schedule is constructed with Adam's (Adams et al., 1988) shifting bottleneck heuristic based on the sequence. They proposed a genetic strategy instead of enumerative tree search for JSSP. The GA was used to evolve a chromosome from a list of chromosome in machine order.

Matta (2009) used Liaw's (Liaw, 2000) approach for the basic open shop scheduling problem i.e. a chromosome was represented as a string of operations rather than as a string of jobs. That is, each gene stands for one operation and each operation is listed in the order in which they are scheduled. Matta (2009) used operations-based chromosome representation with the added dimension of a stage element or defined a gene as a two-dimensional array with the first element indicating the operation and the second element indicating the stage on which the operation is to be scheduled.

Gao et al. (2007) used Priority-based representation for GA and applied their algorithm to FJJSP three objectives: min Makespan, min maximal machine workload and min total workload. Later in another work Gao et al. (2009) applied their algorithm to a multi-objective scheduling problem.

Chaudhry and Drake (2008) used the permutation representation to a solution scheduling problem, i.e., a list of jobs is itself taken as a chromosome. For example, if in a flow shop scenario there are five jobs A-B-C-D-E, one chromosome according to permutation representation can be ABCED, while another could be DECAB. On the other hand, in a job-shop scenario if there are two jobs, each having three operations then the chromosome representation keeping in view the technological constraints (i.e., operation 2 of job 1 cannot be done unless operation 1 is finished and so on). Pan and Huang (2009) proposed HGA for No-wait JSSP with the objective of minimizing total completion time. A Job-based representation is used and genetic operation is defined by cutting out a section of genes of a chromosome and treated as a sub-problem, and is then transformed into an asymmetric travelling salesman problem (ATSP) and solved by Johnson et al.'s (Johnson et al., 2002) Nearest Neighbor (NN) algorithm and Patching (PA) .

Zhang and Wu (2010) successfully implemented a decomposition based hybrid optimization algorithm is presented for large-scale job shop scheduling problems in which the total weighted tardiness must be minimized. They used SA and GA to solve JSSP. They also used a fuzzy inference system to calculate the jobs' bottleneck characteristic values which depict the characteristic information in different optimization stages.

More recently Kachitvichyanukul and Sitthitham (2011) presented a two-stage GA (2S-GA) for multi-objective JSSP. The 2S-GA is proposed with three criteria: min Makespan, min Total Weighted Earliness, and min Total Weighted Tardiness. In Stage 1 they applied parallel GA to find the best solution of each individual objective function with migration among populations. While in Stage 2 it combines the populations and the evolution is based on Steady-State GA using the weighted aggregating objective function. The random keys representation is applied to the problem and the schedules produced using a permutation with m-repetitions of job numbers. 2S-GA performance was tested against benchmark instances. It shows that it has outperformed some of published traditional GA approaches.

Table 3. 5: comparative analysis of GA for FT06, FT10 and FT20 problems

Papers (GA)	**FT06** Optimum = 55	**FT10** Optimum = 930	**FT20** Optimum = 1165
Nakano (1991)	55	965	1215
Yamada (1992)	55	930	1184
Pesch (1993) - 2J-GA	55	937	1193
Pesch (1993) - JC-GA	55	937	1175
Pesch (1993)	55	937	1165
Storer/Wu/Park (1993)	55	954	1180
Gen et al., (1994)	55	962	1175
Mattfeld et al. (1994)	55	930	1135
Birewirth (1995)	55	936	1181
Dorondorf/Pesch (1995) – P-GA	55	960	1249
Dorondorf/Pesch (1995) – SB-GA	55	938	1178
Mattfeld (1996)	55	930	1165
Norman & Bean (1997)	55	937	1165
Shi (1997)	55	930	1165
Cai et al., (2000)	55	930	1165
Wang and Zheng (2001) GA	55	997	1247
Park et al. (2003) PGA	55	936	1178
Goncalves et al. (2005) GA-PDR	55	930	1165
Baune et al. (2005)	55	930	1165
Morshed (2006) – GA-TS	55	930	1165
Noor and Khan (2007)	55	930	1165
Tariq (2008)	55	930	1165

The GAs have been applied to many benchmark problems. FT problems are common among almost all researchers. Therefore an overview of results from GAs applied to *minimum-makespan* JSSPs with different representation for the past years is presented in Table 3.5. 1st Column shows the author, 2nd 3rd and 4th Column shows the result obtained by GAs for FT06, FT10 and FT20 benchmark JSSPs.

3.5 Literature Review Conclusion

The primary objective of this research is to study scheduling problems and their solution approaches in order to identify the area where useful concepts and techniques can be developed and their implementation can significantly contribute to performance improvement of these scheduling problems. The review indicates that computational intelligent techniques dominated literature on scheduling and considerable developments have been made in the recent years. However, these developments faced the inherent difficulty and still there is no heuristic with a guaranteed performance across all sizes of problem specially large problems. This is the reason that scheduling problems are considered to be the hardest optimization problems and there is a need of new approximation techniques which can guarantee that the approach produces optimum results.

The study of solution approaches applied to JSSP over the past few decades shows that GAs are dominating due to their search capabilities among all approaches. However, GA requires fine tuning in order to yield optimum result. Therefore, researchers shifted their focus mainly on hybrid approaches mainly combining GA with other AI techniques or introducing heuristics to main GA loops such as local search heuristics. The initial solution in GA or HGA can significantly effect the JSSP solutions. The fitter the initial solution the faster the GA will converge with a better solution. It is therefore recommended that a new heuristic rule-based systems must be developed which can provide stable results across the problem sizes and can be incorporated with AI techniques such as GA. Such heuristic rules and hybrid approaches would not only be applicable to JSSP but also to solve other complex combinatorial and real life problems.

In the following chapter two new heuristics developed for scheduling problems are discussed and in chapter 6 their comparative analysis with other conventional heuristics is carried out.

CHAPTER 4

DEVELOPMENT OF NOVEL HEURISTIC RULES

4.1 Introduction

In scheduling literature, the terms such as scheduling rules, heuristic rules, dispatching rules, or priority rules are often used synonymously (Panwalkar and Iskander, 1977). In the past six decades, there has been a substantial growth in the field of sequencing and scheduling research. The heuristic scheduling rules that deal with the complexities of manufacturing under global competition are currently much sought after. These heuristics prioritize all jobs that are waiting to be processed on a resource. It is increasingly recognized that all the strengths of traditional operations' research, knowledge based systems, heuristic rules, Artificial Intelligence (AI) and sophisticated user interfaces will be necessary to build the needed system which can fulfil the future needs. Such successful integration of approaches has not really occurred yet, despite the fact that many systems have been built by researchers that attempt some integration.

According to Panwalkar and Iskander (1976), scheduling research can be divided into two main categories: theoretical research dealing with optimizing procedures limited to the static problems and experimental research dealing with scheduling rules in static and dynamic cases. Pinedo (1998), called this experimentation research in scheduling as scheduling in practice. He categorizes heuristic in general purpose procedure rules along with simulated annealing, tabu search, and Genetic Algorithm (GA).

This chapter describes a number of general purpose procedures (existing and new) that are useful in dealing with scheduling problems in practice is presented. These procedures can be easily implemented with relative ease in industrial scheduling systems. All the procedures described are heuristics that do not always guarantee an optimal solution; instead they aim to find reasonably good solutions in a relatively short time. These heuristics are fairly generic and can be adopted easily to a large variety of scheduling.

The first section of this chapter gives a generalized heuristic, heuristic classifications and an overview of some selected traditional heuristic rules. In the second section, a *procedure is developed and presented* which is used to developed new heuristics and can be used in future for development of more heuristic rules. The procedure is based on human intelligience which analyzes and synthesizes existing rules followed by experiments on these rules with a view to improvement through techniques such as swap and delay. In the third section, the outcome of the analysis and the need for the new heuristic has been discussed. The final section, *present two novel heuristics* that have been developed, based on the literature review and experimental study conducted during this research.

The proposed heuristic rules are applied to selected benchmark JSSPs of different hardness in order to check their validity and effectiveness. The development of Hybrid Genetic Algorithm (HGA) based on the novel heuristic rules are discussed in Chapter 5 and Chapter 6 then covers the detailed performance analysis of the heuristic rules and HGA developed in Chapter 4 and Chapter 6 respectively.

4.2 Development of New Heuristic Rules

Every quality optimization model has certain common characteristics. They are developed in such a way that they can be applied as criteria of efficiency for any existing or new optimization problems. The quality of optimization model is normally evaluated through its ability to fulfil the following common requirements:

- Validity
- Reliability
- Ease of testing
- Ability to interpret and compare
- Cost effectiveness

These features and a certain level of generalization of the rules and procedures must be fulfilled for the particular optimization problem to be sufficiently. In such generalizations, one should always keep in mind the need to foresee as many limitations as possible and the need to ensure the ability to upgrade as well as the flexibility of the model itself.

Figure 4.1 shows the research and development stages which led to the development of the two novel heuristics. As can be seen in the figure, the development process is divided into three main phases (i) need identification (ii) analysis and synthesis and (iii) new heuristic rules. The need identification is further divided into literature review of scheduling problems and their solution techniques (see Chapter 3 for detail). The analysis and synthesis are divided into study of existing heuristic rules, modification in existing rules using human intelligence, manual procedures, result comparison of the existing rules and modified rules. The new heuristic rules discussion is also divided into the procedure and validations of the two new heuristic rules.

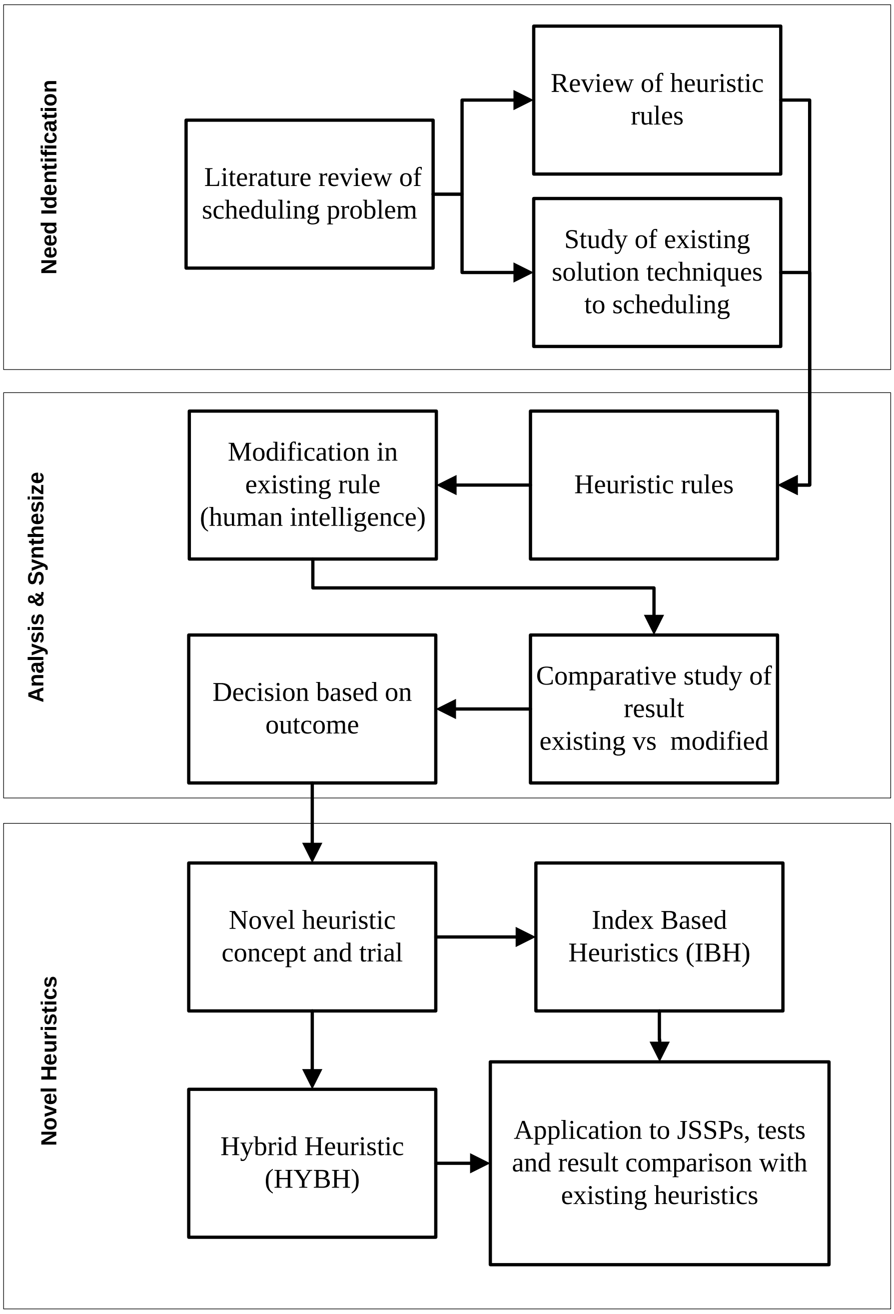


Figure 4. 1: Phases of development of new heuristics

From the literature review, it has been concluded that AI tools have been the most extensively applied to scheduling problems. Mostly, the hybrid scheduling optimization models perform better than single tools. It was also noted that most GAs are combined either with another AI tool such as Artificial Neural Networks (ANN), Fuzzy Logic (FL), Simulated Annealing (SA) or with heuristic rules.

One of the research project objectives is to create a new heuristic based algorithm based on the conclusion of the literature review. Rather than using the existing AI tools and heuristic rules, which researchers have extensively applied to JSSPs, it was decided to develop new heuristic rules, which could perform better than the current rules. Therefore, a substantial amount of literature material was gathered and reviewed related to scheduling techniques applied to JSSP (see Chapter 3).

The processing time based heuristic rules applied to JSSP were mainly focused on the selection of objective function i.e. Makespan, from which it was identified that there is still need of some models which could be reliable, fast, effective and can easily be implemented to scheduling problems of any hardness and size.

As discussed the heuristics are normally intended to minimize the inventory and/or tardiness costs and therefore it is natural that they have a direct proportionality to the time period of flow time and tardiness of jobs respectively. They are also applied to JSSPs for minimization of total completion time and better machine utilization. On the basis of their nature of operation the heuristic rules can be classified into five categories:

(i) Processing time based rules e.g SPT
(ii) Due date based rules e.g EDD
(iii) Simple rules based on shopfloor condition in certain machine environments e.g Shortest Queue (SQ) first rule, which is a time dependent or dynamic rule
(iv) Combined rule i.e. combination of any rule in (i), (ii) and (iii).

The general processing time based rules perform better under tight load condition, whilst due date based rules perform better under light load condition (Conway, 1964; Chang et al., 1996; Rajendran et al., 1999). The choice of heuristic depends upon which criterion is to be met, for example, Makespan, mean flowtime or tardiness.

The typical processing time based SPT rule has often been used as a benchmark since this rule was developed and has been ranked on the top (Chang et al., 1996). These rules have been used by many practitioners and researchers (Conway, 1964; Panwalkar and Iskander, 1977; Pinedo et al., 1998; Rajendran and Holthaus, 1999; Zhou et al., 2001; Weckman et al., 2008) in their review and research for minimization of Makespan, whilst the due date or slack-related rules performed well in tardiness measures (Pinedo et al., 1998). The combined processing time and due date based rules are also used in minimization of Makespan and tardiness e.g. Minimum Slack (MS) rule, Critical Ratio (CR) rule etc. Ramasesh (1990), has presented an excellent report on some of these dynamic and popular heuristics.

In deterministic scheduling problems, it is common practice for researchers to assume that all jobs are available at the beginning of a scheduling period. Therefore, it is natural that many optimizations and heuristic algorithms have been developed for minimization of Makespan or total flow time or both. Many of the studies have therefore considered processing time based rules such as SPT, LPT, FIFO (Rajendran and Holthaus, 1999). In this research of deterministic scheduling problems, the optimization objective function selected is Makespan (for detail see Chapter 2). These heuristic rules are useful in finding optimal or near optimal schedules with a single objective. Therefore, the research analysis is narrowed down to processing time based and due date rules (with all releases and due dates zero) rules as they have proven to provide better solution comparatively and have been listed as key factors by Chang et al., (1996) in their performance review of 42 heuristic rules. Table 4.1 lists some of the better known heuristic rules or traditional rules applied to JSSPs.

Table 4. 1: List of some better known heuristic rules

S No.	Heuristic Rule
1	FIFO – First In First Out
2	LPT – Largest Processing Times
3	SPT – Shortest Processing Time
4	CR – Critical Ration
5	EDD – Earlist Due Date
6	MS – Minimum Slack
7	WSPT – Weighted Shortest Processing Time

In literature, the Makespan comparisons or performance comparisons are made using the Relative Deviation (RD) measure or the Mean Relative Error (MRE), also known as the percent GAP (% GAP). The measure % GAP is the deviation of the Makespan value obtained by a particular heuristic from the optimum or the global Makespan. It represents a measure of the quality of the best global Makespan. The % GAP for a particular heuristic is calculated from the best-known global or Lower Bound (LB) (or optimum Makespan) and the Makespan obtained from particular algorithm using the following relative deviation formula:

$$\% \, GAP = \left\lfloor \frac{Makespan\ found - Makespan\ Optimum}{Makespan\ Optimum} \right\rfloor \times 100 \qquad \text{Equation 4.1}$$

4.3 Analysis and Synthesis of the heuristics rules

During the development phase, an extensive literature review resulted in knowledge of existing heuristic rules and the procedures. This knowledge helped in avoiding any repetition of the work already covered in the literature. Figure 4.2 shows the next development phase or analysis phase after need identification phase. In the analysis phase, solutions obtained from existing rules in the form of Gantt charts are considered. Then gaps or idle machine times and the poor jobs, which are mostly affecting the Makespan of the problems are identified. The poor jobs might be the last assigned job on a machine or the job with large waiting time on a machine.

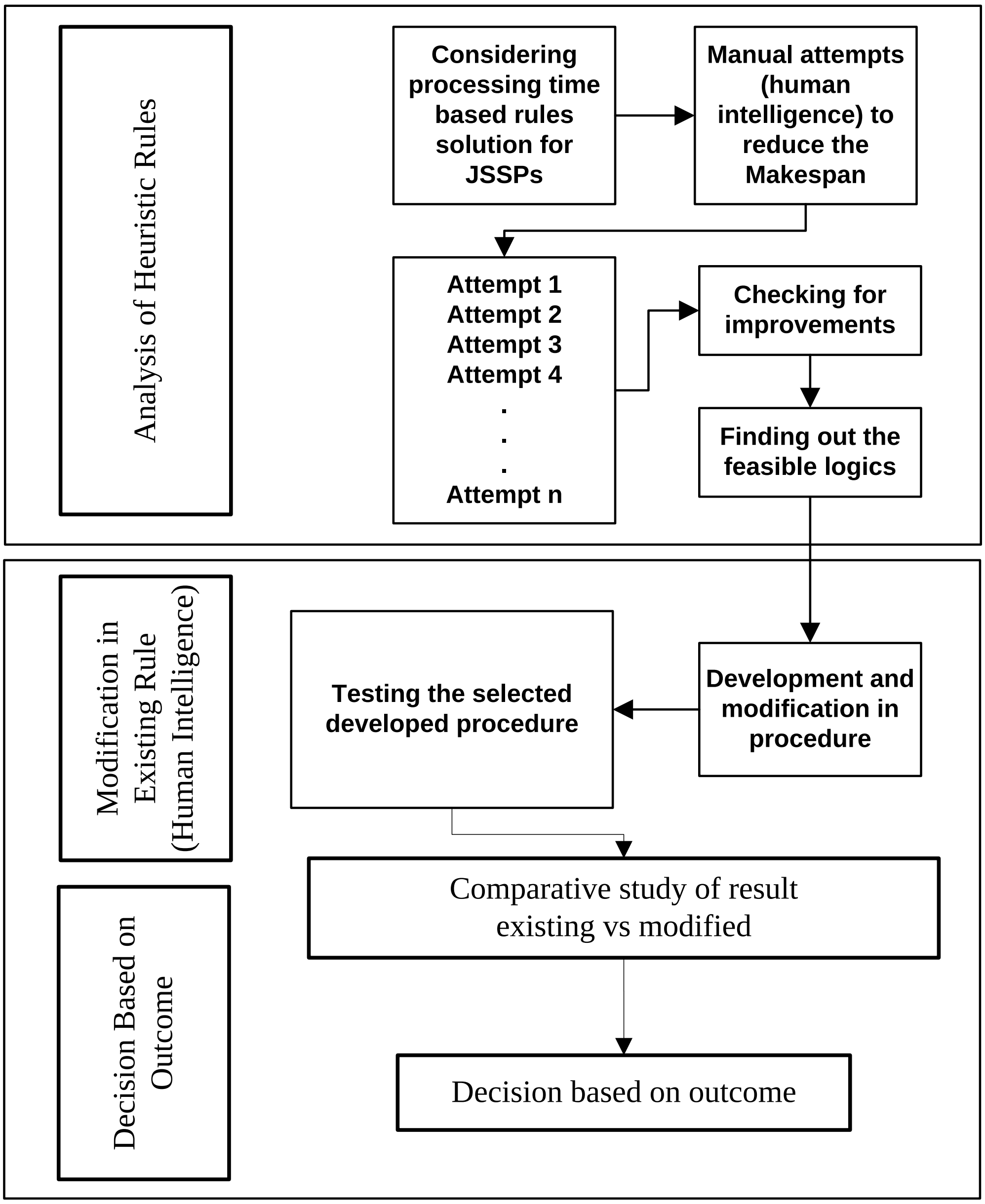


Figure 4. 2: Analysis and synthesis of stage of heuristic rules

Once the identification procedure ends, some manual procedures or additional steps are applied to the solutions of the existing rules. These procedures naturally alter the solutions of the problems such as Makespan, tardiness or machine utilization. The impact of these procedures and results are recorded during the process. These results from new procedure are then compared and best procedures are selected on the basis of the comparison. This selection process is called synthesis.

Manual procedures are carried out using drawing sheets, colour pencils and solutions in the shape of Gantt charts. In literature, these kinds of procedures or techniques are applied to solutions and are known as a human intelligence procedures. Each new technique or additional step applied to the existing solution is called an *attempt* in this research. These attempts are different in number depending upon the problem size. In small problems, the possibility of changes in either job's priority order or delaying a job for another is limited as compared to a larger problem. This argument is supported with detail examples in the next subsections. The manual procedure results are recorded and a comparison with the original solution for any improvement in the objective.

An important step in the analysis phase is determining which procedure is logical and can be formulated or put into the form of an algorithm. All the attempts are then assessed and formulated. Some attempts were found not logical, which were either discarded or further modified in order to transform it to a logical procedure. In the following subsections, the analysis phase is explained with an example problem. The Shortest Processing Time (SPT) rule is applied to this problem, which is considered the most commonly used rule for JSSP in literature and is found to be very effective in Makespan minimization of average measures (Chang et al., 1996).

4.3.1 Analysis of Example Problem

A simple three jobs and three machine example problem with best known optimum Makespan of 63 is selected initially for experimentation, as shown in Table 4.2. As discussed the main objective function is the minimization of the Makespan.

Therefore, the analysis carried out during development stages are focused on how to minimize the Makespan. In the example problem, each job consists of three operations and the machines' constraints are given. For example J_1, must be processed on Machines M_1, M_3 and M_2 for 16, 21 and 12 units of time respectively. As mentioned, the analysis was carried out manually with the help of drawing sheets, rulers and colour pencils initially in order to study the effect of each step and modification by considering additional steps or techniques through human intelligence, with a view to develop a new heuristic rule.

Table 4. 2: Example process plan

Process Plan						
Jobs	**O_1**		**O_2**		**O_3**	
	M	**PT**	**M**	**PT**	**M**	**PT**
J_1	1	16	3	21	2	12
J_2	1	15	2	20	3	9
J_3	2	8	3	18	1	22

Using the SPT rule the example problem is evaluated and the final schedule is shown in Figure 4.3. The X-axis of the figure shows time and the Y-axis shows different machines. The first, second and third subscript of Operation O, represents job number, operation number and machine number respectively. For example, O_{132} mean that Machine 2 loaded with Job 1 for Operation 3. This operation starts at time 84 and finish at time 97, which is also the Makespan of the problem.

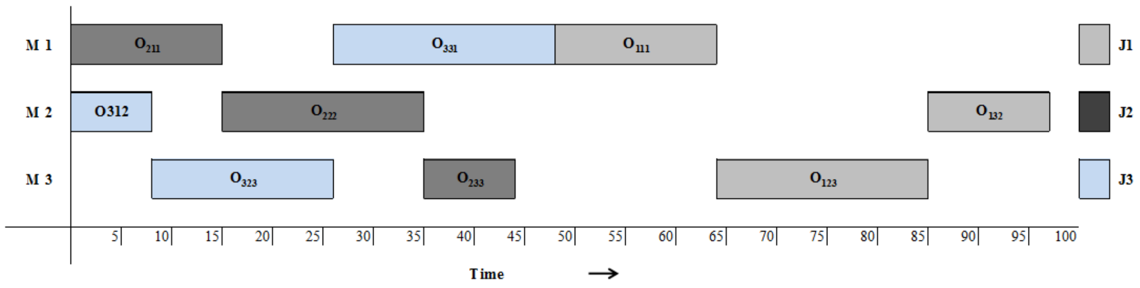


Figure 4. 3: Machine Gantt chart for example problem using SPT

By viewing the schedule or machine Gantt chart generated from SPT, visible gaps or idle machine intervals can be seen between the operations of batch jobs on each machine. For example, on Machine M_2 a gap can be seen in between Operations O_{222} and O_{123}. This gap is actually representing an idle time of the machine or the time when a machine is not in use. An ideal heuristic must generate a gapless schedule i.e. each machine's utilization is 100%. Obviously, the Makespan will be equal to an optimal or lower bound. In order to achieve such a schedule, the sequencing of operations should be changed on one or all machines in order to look for all possible schedules. This process of 'playing' with operations on a machine is termed as *experimentation* in this research and each change or application of new or modified rule is called an *attempt* in this research.

4.3.1.1 Attempt 01

The first attempt is made by selecting J_1 because its third operation (O_{123}) finish time is the Makespan time and the idle time in Machine M_2 between O_{222} and O_{123} is the largest on the chart as shown in Figure 4.4. In order to reduce the idle time and improve the starting time of O_{123} its first Operation O_{111} is inserted in the gap between O_{211} and $O_{331,}$ resulting in changing of O_{111} starting at 48 to 15 time units, as shown in Figure 4.4.

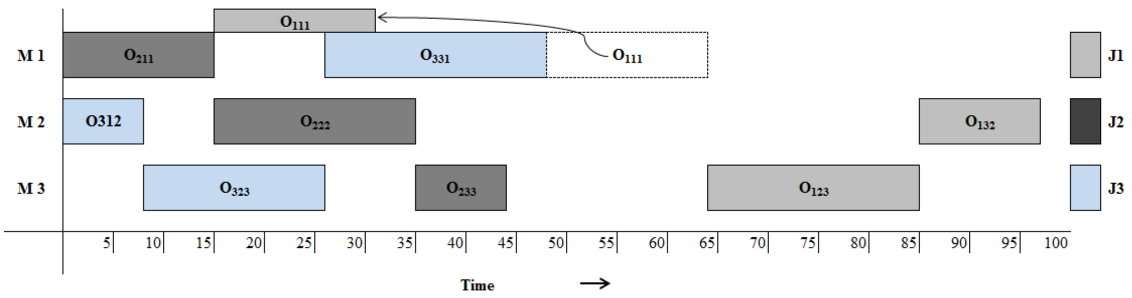


Figure 4. 4: Attempt 01 – Changing Job 1's Operation 1 (O_{111}) position on Machine 1

This kind of move always causes complexities in the procedures. These complexities can be one or more than one depending on the size of problems. In this particular problem, the move causes O_{111} to overlap with O_{331}. This problem can be sorted in many ways such as introducing a technique called delay. The delay technique will delay O_{331} a number of time units equal to the overlapped time units. For example in this problem, O_{331} is delayed by 5 time units as shown in Figure 4.5.

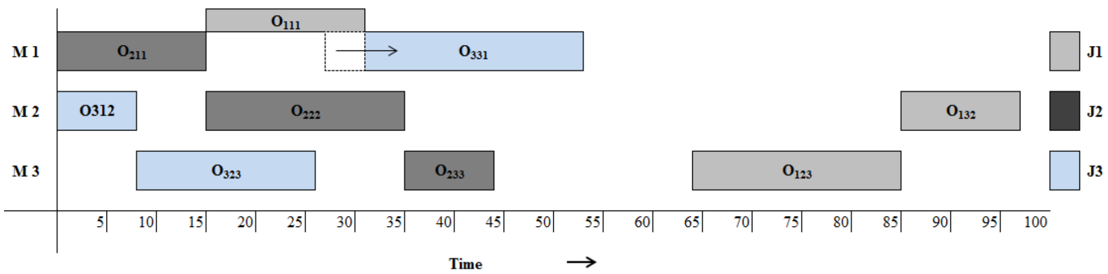


Figure 4. 5: Delay technique applied to (O_{331}) the problem

In such move, it is very important to check the legality and feasibility issues i.e. it should not violate the precedence constraints on any machine. In this case, the delay of O_{331} and movement of O_{111} to an early time position or starting time position does not affect the precedence constraints. Therefore, this experiment step is legal and can yield a feasible schedule. However, the O_{132} or O_{123} cannot be inserted in the void in between O_{312} and O_{222} or O_{323} and O_{233} respectively, because it will violate the precedence constraints as shown in Figures 4.6 and 4.7.

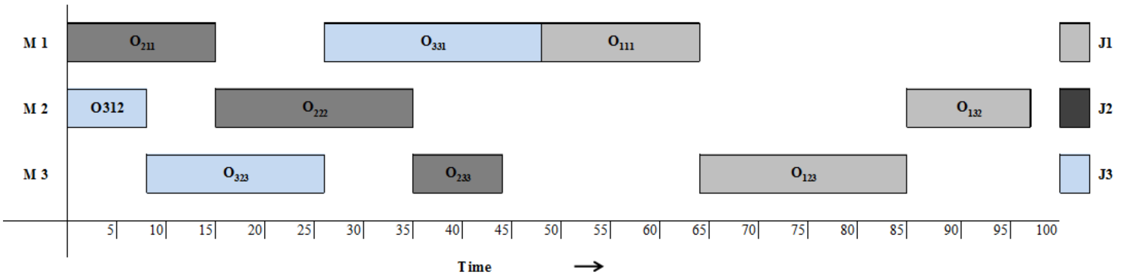


Figure 4. 6: Illegal move and infeasible schedule

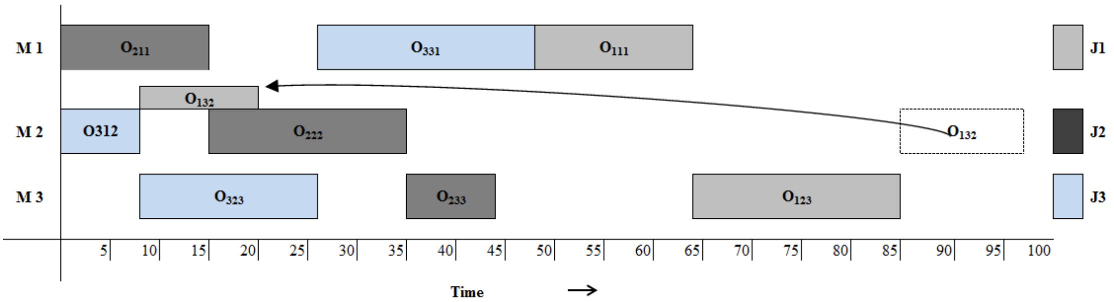


Figure 4. 7: Illegal move and infeasible schedule

In Figure 4.8, the O_{111} affects the rest of the operation on Job 1. The O_{123} is moved to the position 44 time units as its starting point. Hence, O_{123} ends at 65.

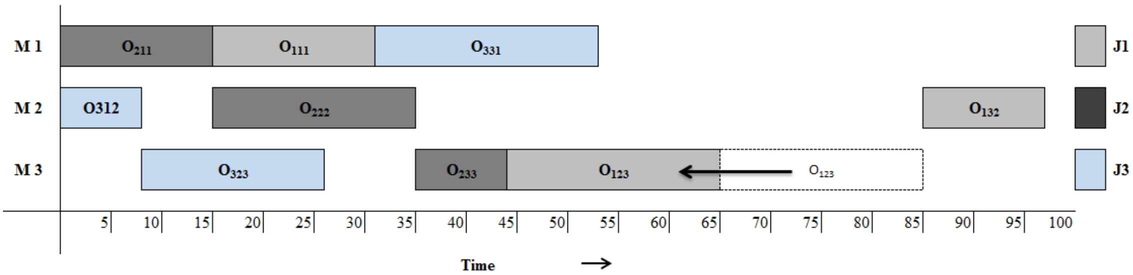


Figure 4. 8: Attempt 01 O_{123} movement

The O_{132} can also start early at 65 instead of 85 time units as shown in Figure 4.9. As a result, the overall completion time on Machine M_2 is reduced to 77 time units, which is an improvement as compared to the actual 97 by SPT.

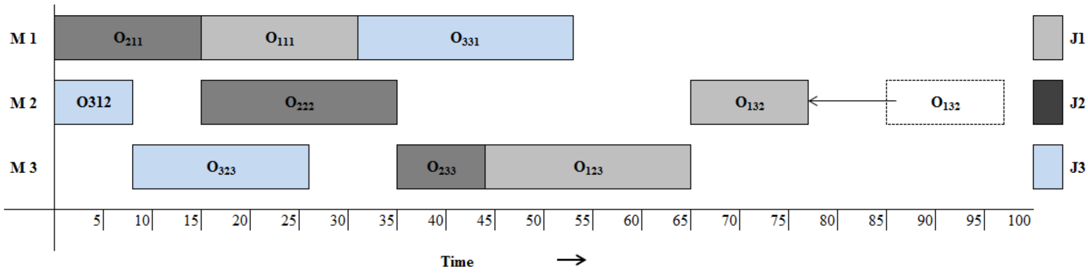


Figure 4. 9: Attempt 01 O_{132} movement

4.3.1.1.1 Attempt 01 – Conclusion

From the illustrated example, with manual procedure or human intelligence, a reduction in the Makespan of the example problem is achieved. The insertion technique creates complexities such as overlapping of jobs on a single machine. As pre-emption is not allowed in JSSPs, so therefore an alternated procedure of delay technique is used in order to resolve the overlapping issue. The delay technique is effective and helps in improvement of Makespan solution and to produce a feasible solution.

It is also observed during the procedure that to identify a job for movement or insertion in a gap there is a need for few parameters, which are to be calculated, such as start time and finish time of each job on Gantt chart, idle times on each machine, waiting times of each job, processing times of each job, precedence constraints of each job, and current and next machine for each job during the process. Hence, there is a need of a procedure that can calculate this information and list in a table (array) form.

4.3.1.2 Attempt 02

Consider the same example initial problem solution shown in Figure 4.3 for a second attempt to reduce the Makespan. In this attempt, the initial first three steps are the same i.e. selecting J_1, and inserting its first Operation O_{111} in the gap between O_{211}

and O_{331} resulting changes in O_{111} starting 48 to 15 time units as shown in Figure 4.5. The third step is also the same as in Attempt 01 i.e. the delaying of O_{331} for O_{111} in order to avoid the violation of the given precedence constraints.

In this attempt, rather than moving O_{123} to a position of 44 unit of time it is inserted in the gap between O_{323} and O_{233}. In this case, the overlapping of operations can be seen on Machine M_3 as shown in Figure 4.10.

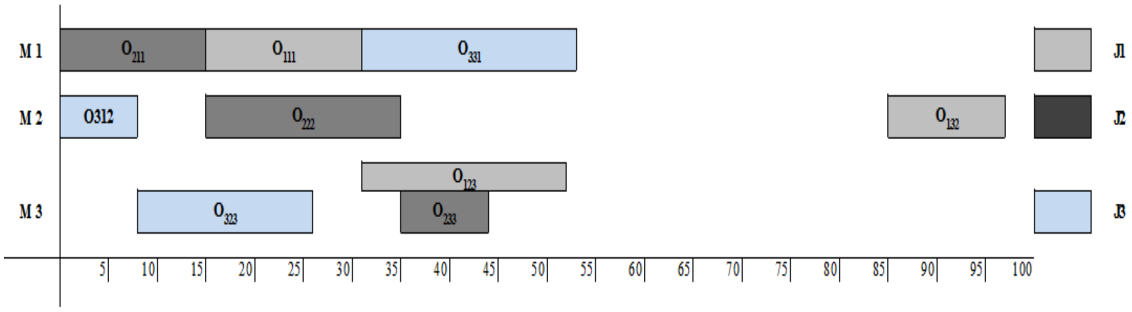


Figure 4. 10: Attempt 1 (O_{123}) on the example problem solution (Gantt chart)

In JSSPs, the pre-emption is not allowed, therefore the delay technique is applied in order to resolve the tie between Job J_2 and J_1 for O_{123} and O_{233}, respectively. Hence, the O_{233} is delayed for 17 unit of time i.e. the end of O_{123} as shown in Figure 4.11.

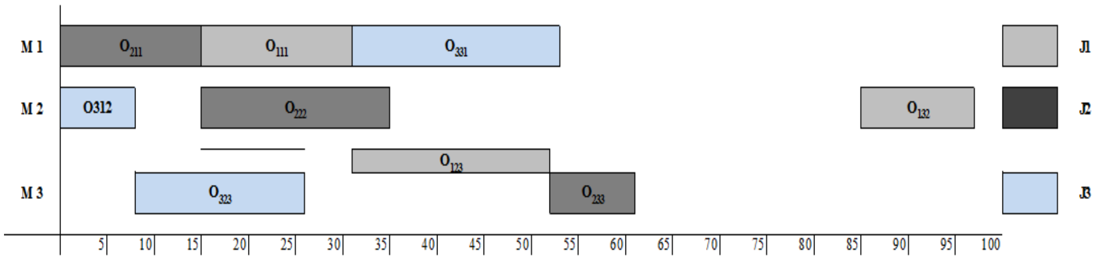


Figure 4. 11: Delay technique applied to O_{233}

In the final step, the O_{132} is moved to a starting point of 52 time units, which is the only option left. Hence, this results in new lower Makespan value for the problem.

The new lower Makespan value is 64 on the final feasible schedule and is shown in Figure 4.12.

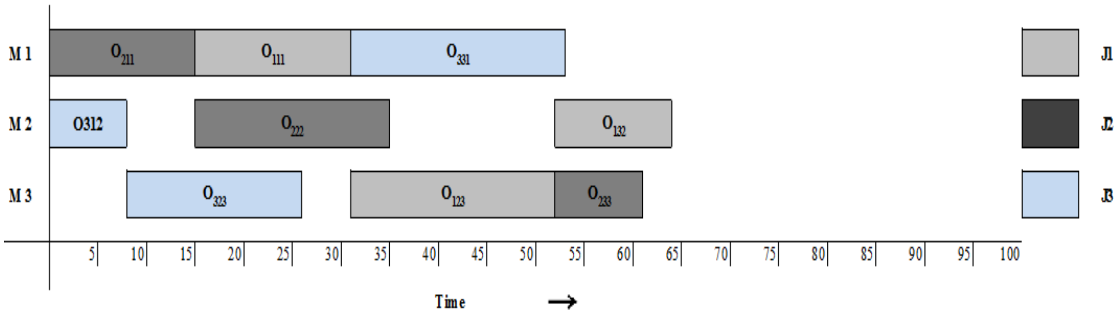


Figure 4. 12: Final schedule from Attempt 02

4.3.1.2.1 Attempt 02 – Conclusion

In this attempt, a similar procedure is used initially on machine i.e. M_1. For the second machine, rather than just moving rest of J_1 operation to new starting points, O_{123} was inserted in between O_{323} and O_{233}. This resulted in overlapping and of jobs on a M_3. Since, pre-emption is not allowed in JSSPs, therefore a delay technique is used to resolve overlapping issue on M_3. On Machine M_2 the only option of movement is applied and hence, a final feasible schedule is developed and with a better Makespan solution shown in Figures 4.10 to 4.12.

This attempt is also legal and can produce feasible schedules. However, it important that all the parameter should be available at the start of attempt as mentioned in Attempt 01 conclusions.

4.3.1.3 Attempt 03

Consider the same example problem solution shown in Figure 4.3 for another attempt to reduce the Makespan. Job J_1, is selected for this attempt the same way as mentioned in Attempts 01 and 02, for the first operation. In this attempt, instead of inserting a job in a gap, and its priority was changed. Instead of assigning it at the end, O_{111}, it is assigned first and O_{211} is assigned last as shown in Figure 4.13. In other words, the positions of these two job Operations O_{111} and O_{211} was exchanged or swapped.

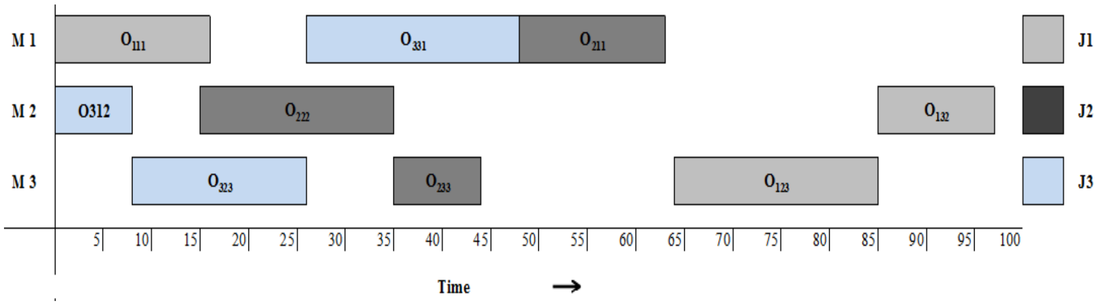


Figure 4. 13: Attempt 03 swapping or exchange priority on M_1

This swapping resulted in violation of precedence constraints. Therefore, the rest of the operations on each machines were also swapped i.e. the O_{123} and O_{233} were swapped on Machine M_3, as shown in Figure 4.14.

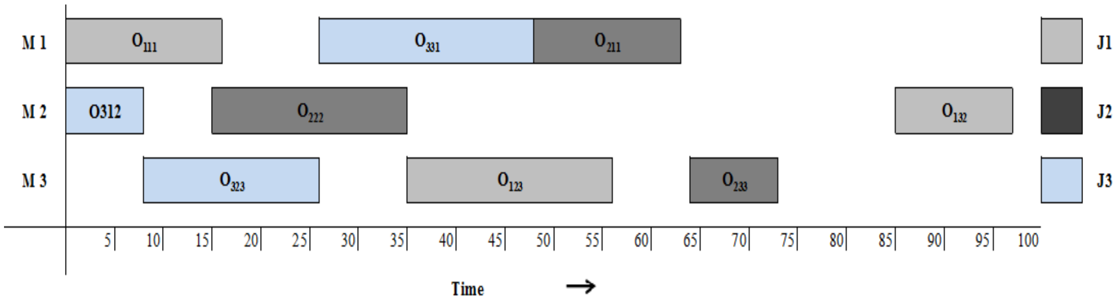


Figure 4. 14: Swapping of O_{123} and O_{233}

The schedule is still feasible and logical. However, if the same step is applied to the Operations O_{222} and O_{132}, it causes complications and the schedule will be illegal because the O_{132} (third operation of J_1) is executed before O_{123} (second operation of J_1) as shown in Figure 4.15.

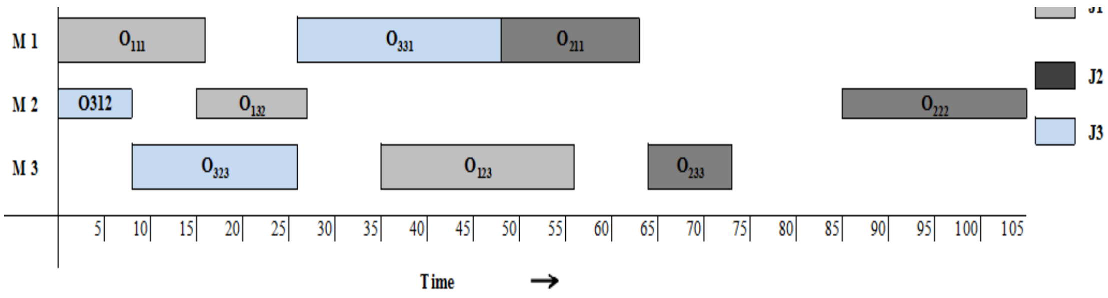


Figure 4. 15: Swapping of O_{222} and O_{132}

4.3.1.3.1 Attempt 03 – Conclusion

In this attempt a swapping or exchange priority procedure was applied to the problem. The exchange works fine for two steps, however, the final step, results in overlapping and violation of precedence constraints. Hence, a final schedule is not feasible and the swapping priorities turned out to be illegal.

Table 4. 3: Example process plan

Process Plan						
Jobs	O_1		O_2		O_3	
	M	PT	M	PT	M	PT
J_1	1	16	3	21	2	12
J_2	1	15	2	20	3	9
J_3	2	8	3	18	1	22

4.3.1.4 Attempt 04

Consider the same example problem solution shown in Figure 4.3 for another attempt to reduce the Makespan. Table 4.3 shows J_3 was assigned first because the processing time was the shortest among all and there was no tie of this job with others on the same M_2 as well. However, for the first Operation O_1 on Machine M_1 there is a tie, the between Jobs J_1 and J_2. To resolve the tie, the SPT rule selected J_2 (Processing time 15) because its processing time was shorter than J_1 (processing time

16). In this attempt, a procedure is developed and applied if there is tie of jobs on the same machine, the tie should be resolved in a way such that the longer processing time job should be assigned first. However, if there is no tie the jobs should be assigned using the SPT rule. Figure 4.16 shows that Job J_3 with shortest processing time of 8 is assigned first. The tie on Machine M_1 is resolved using larger instead of shorter processing times. Hence, J_1 having processing time of 16 is assigned before J_2 having processing time of 15 time units.

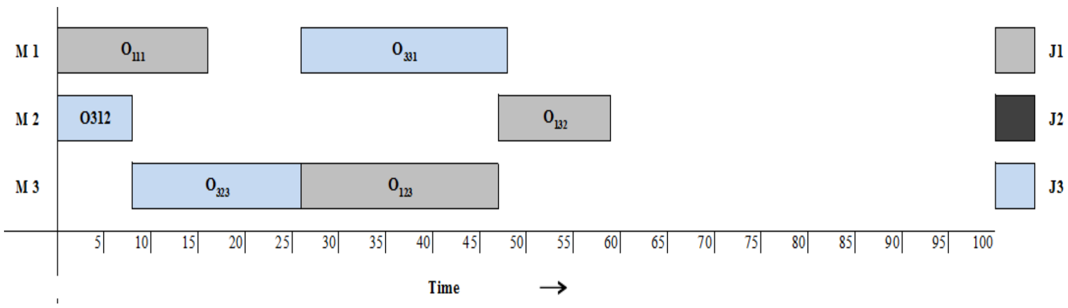


Figure 4. 16: Attempt 04 changing priority in case of tie for first operation

In Figure 4.17, the final schedule is shown with J_2 assigned in last. The schedule is feasible, the procedure is legal, and it has shown some improvement in the Makespan.

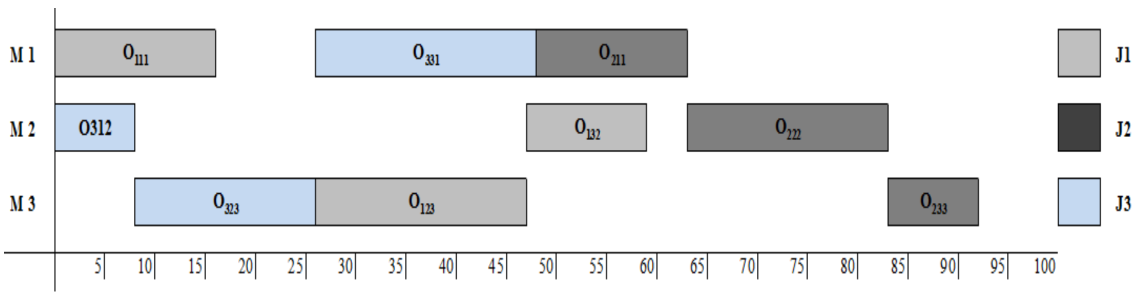


Figure 4. 17: Final Schedule attempt 04

4.3.1.4.1 Attempt 04 – Conclusion

In this attempt rather than looking at the final schedule Gantt chart, the problem data was considered for evaluation of the schedule. The procedure resolved ties between two or more than two job for same operation on a machine using larger processing time rule instead of SPT.. An improvement in the Makespan value is recorded. This attempt is legal and can produce feasible schedules. However, it important that all the parameter should be available at the start of an attempt, as mentioned in Attempt 01 conclusion.

4.3.1.5 Attempt 05

Consider the same example problem solution and the solution achieved by Attempt 04 as shown in Figure 4.17 and combine it with another procedure (Attempt 01) in order to reduce the Makespan. In this attempt, insert the O_{211} the gap between O_{111} and O_{331} as shown in Figure 4.18. The new starting time of O_{211} changed from 48 to 16 time units.

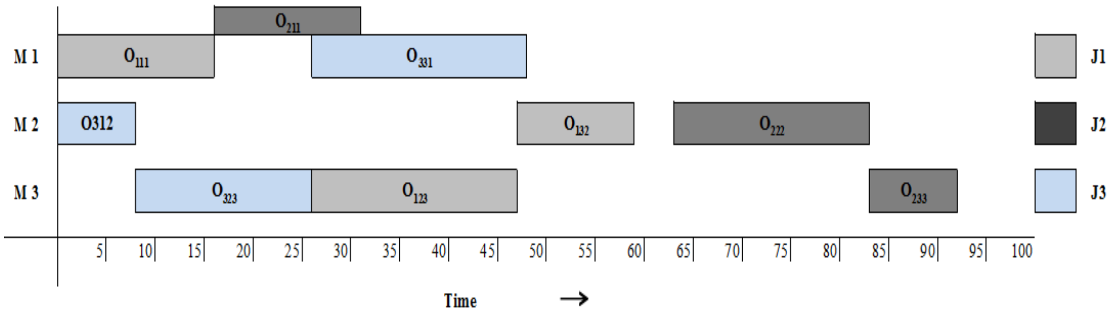


Figure 4. 18: Attempt 06 first step

This insertion move causes O_{211} to overlap with O_{331}. The delay technique is used to delay O_{331} a number of time units equal to the overlapped time units. For example in this problem, O_{331} is delayed by 5 time units as shown in Figure 4.19.

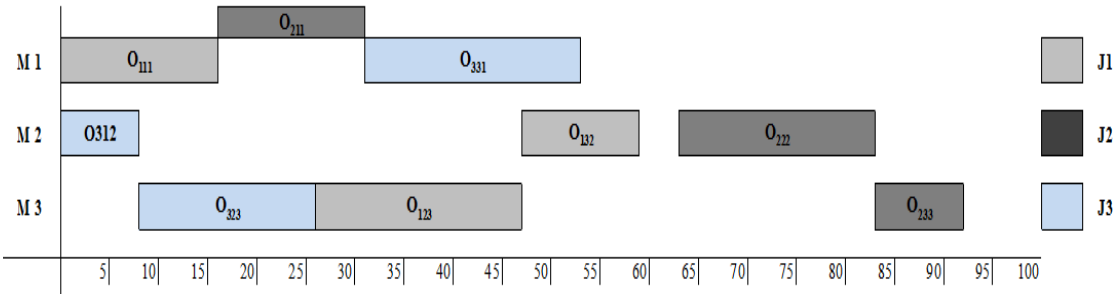


Figure 4. 19: Attempt 05 delay procedure to overlapped operations

The insertion and delay procedure is repeating on M_2, as shown in Figure 4.20. The start time for O_{222} is now 31 and O_{132} is delayed by 4 time units.

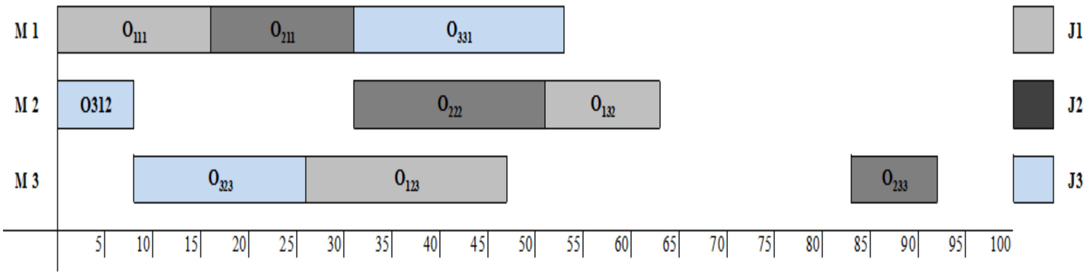


Figure 4. 20: Attempt 05 insertion on Machine M_2

The only option left on M_3 is the movement of O_{233} on M_3 to a new starting position, which is exactly the ending time of O_{222}. The combined procedure reduced the Makespan and the process is legal. The final schedule is shown in Figure 4.21.

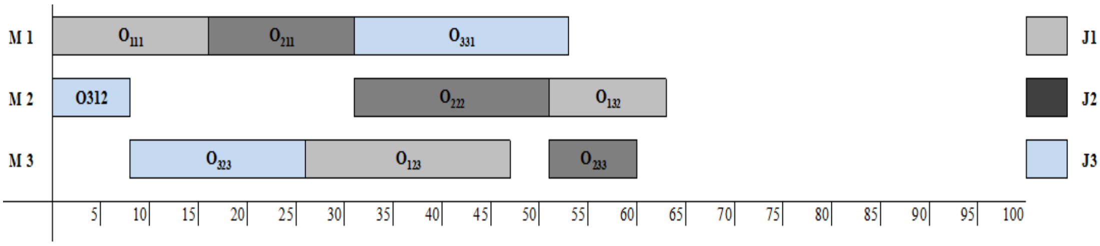


Figure 4. 21: Attempt 05 final schedule

4.3.1.5.1 Attempt 05 – Conclusion

In this attempt, a combined procedure of SPT, swap (resolve ties on longer processing time on same machine) and delay. The procedure yields a feasible schedule and is legal. Here, again the need for availability of all parameters rose, which should be kept in consideration in the algorithm development.

4.3.1.6 Other Attempts

Similar kind of attempts have also been made on other problems with different size and hardness level. In Appendix F shows the Makespan or the result (Gantt charts) for FT06 and LA02 benchmark job shop scheduling problem with some of the new and the discussed legal procedures. The appendix shows results achieved (shown in Gantt charts) and briefly discussed. The appendix also shows applied to some selected benchmark JSSPs. During the attempts, it was observed that the complexities are larger in larger problem. Despite the fact that the attempts lead to better results in the case of smaller problems. The same procedures may lead to poor solutions or almost impossible processes, which make the procedure hard to be programmed. For example, when a simple procedure (Attempt 04) is applied Fisher and Thompsons (1968) – FT06, a six job and six machine problem, it yields a poor solution. Table 4.4 shows process plan of the FT06 problem.

Table 4. 4: FT06 process plan

Process Plan for FT06												
	O_1		O_2		O_3		O_4		O_5		O_6	
Jobs	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
J_1	3	1	1	3	2	6	4	7	6	3	5	6
J_2	2	8	3	5	5	10	6	10	1	10	4	4
J_3	3	5	4	4	6	8	1	9	2	1	5	7
J_4	2	5	1	5	3	5	4	3	5	8	6	9
J_5	3	9	2	3	5	5	6	4	1	3	4	1
J_6	2	3	4	3	6	9	1	10	5	4	3	1

Figure 4.22 shows possible solution obtained using SPT rule. The Makespan achieved by SPT is 73. Attempt 04 is applied to this problem. From Table 4.3 shows,

that there are six numbers of jobs are to be processed on six machines. For Operation 1 on Machine M_3 three of the jobs J_1, J_3 and J_5 have a tie and J_2, J_4 and J_6 have tie on Machine M_2. According to the procedure used in attempt 04, J_1 is to be assigned first to Machine M_3 followed by J_5 and J_3 because J_5 has larger processing time than J_3. Hence, the sequencing order will be J_1 (having processing time of 1 unit), J_5 (having processing time of 9 units), and J_3 (having processing time of 5 units).

In next step, the J_2, J_4, and J_6 and ties for the same operation are resolved using the same technique. J_6 (having processing time 3 units) will be assigned first followed by J_2 (having processing time 8 units) and J_4 (having processing time 5 units). In Figure 4.23 shows, the final schedule evaluated with Attempt 04 procedure. The resulted makespan value is poor as compare to the result achieved by SPT. The J2 is still the poor job and finished last.

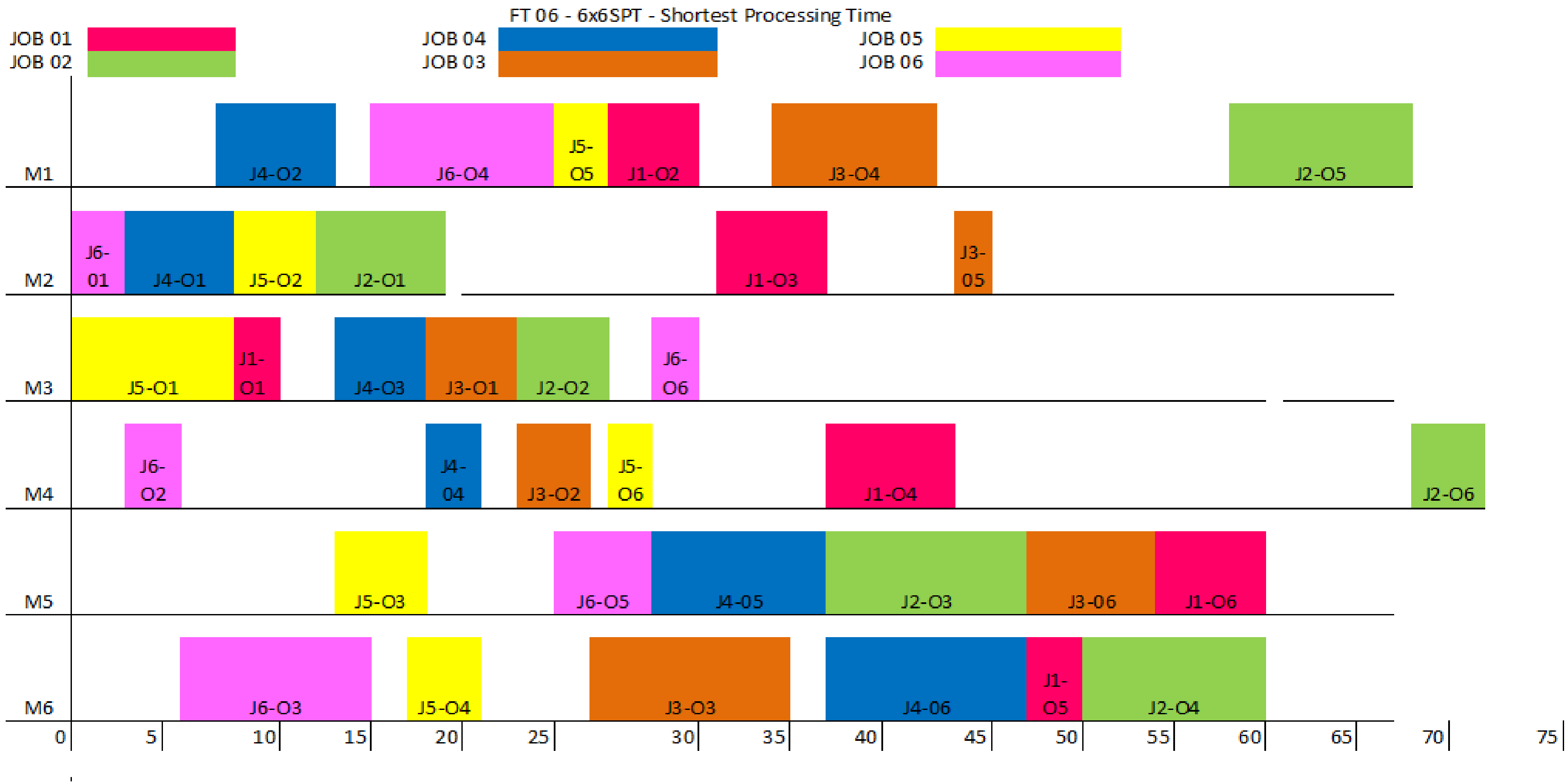


Figure 4. 22: FT06 solution machine Gantt chart using SPT rule

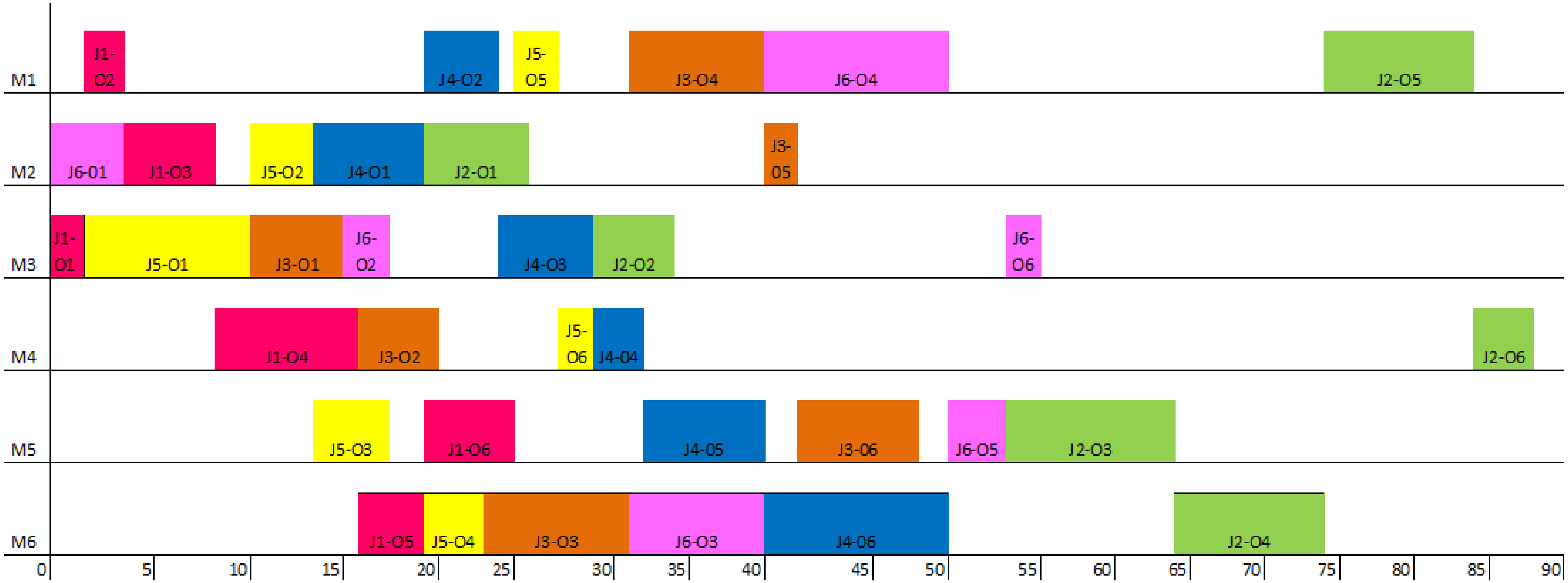


Figure 4. 23: FT06 solution machine Gantt chart using Attempt 04

4.3.2 Development of Logic used for heuristic rule development

In Figure 4.24, a block diagram is given which shows the logical flow for development of heuristic rule algorithms. It shows that general solution procedure for JSSPs.

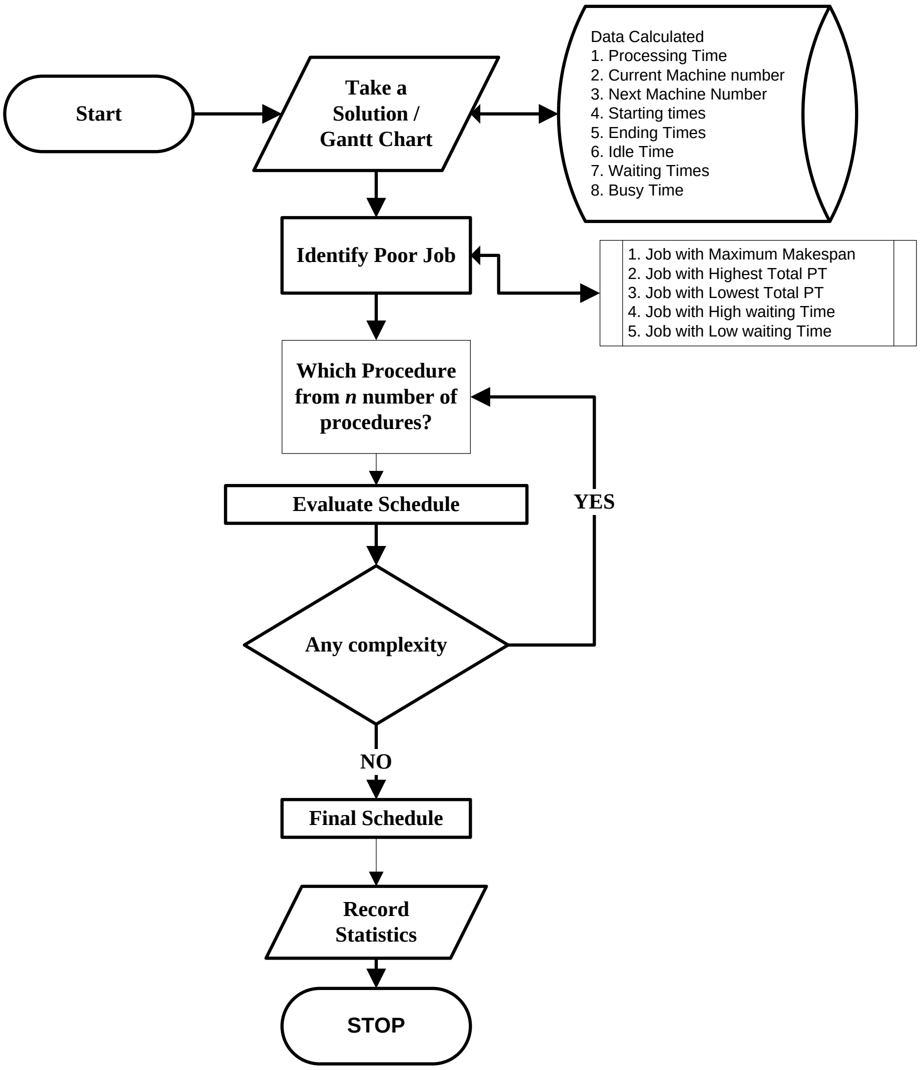


Figure 4. 24: Block diagram of the logic for development of heuristic rules

The algorithm, initially generates a feasible schedule using existing rules and record are the statistics i.e. processing time, in process current machine number, machine idles time, process starting time, process ending time, waiting time of a job for next machine, next machine number for operation. These statistics are used to identify the ‘poor’ job and in different step of procedure. The option for identifying poor job are listed as follow:

(i) job with maximum total completion time
(ii) Job with highest total processing time
(iii) Job with lowest processing time
(iv) Job with highest waiting time
(v) Job with lowest waiting time

Once the job is identified, a new single procedure or combined procedures are applied to the problem and checked for any improvement. If the schedule is feasible, the Makespan is recorded and the procedure is terminated.

4.4 Outcome of the Analysis and Synthesis

The existing need identification, followed by brainstorming of ideas, and subsequent experimentations with existing rules resulted in ideas and techniques for the development of new heuristic rules. These new techniques were then incorporated in the existing techniques and their solutions were analysed. With the implementation of the new techniques, improvements were recorded in the solution of the problems compared to the solution of the existing heuristics.

It was also observed that the newly developed procedure yielded mostly valid results. However, in the case of large scheduling problems the complexities increase. To resolve these complexities different procedures are combined to evaluate a feasible schedule. For example, the overlapping issue is resolved by introducing the delay technique in the procedure, the logic of which is shown in Figure 4.24. However, programming this logic for a new procedure was very difficult and in few cases (such as the delay technique) is not practically possible. Thus the program should be able to identify when and where to use these techniques.

In the above section, the attempts were explained with an example 3x3 JSSP for performance analysis and potential improvement. The objective function selected was Makespan minimization. The known Makespan value of the problem is 63 units of time. The SPT rule initially achieved a Makespan value of 97 with a %GAP of 53.96 with the best known optimum result. In the process, different techniques such as swapping, inserting in gap, changing priority, and delay were applied individually and in combination. Most of the procedures are legal and yielded feasible schedules. However, the combination of two procedures is key to improvement in the Makespan of a problem. For smaller problems the procedure are simple and are easily implemented. While in the case of larger problems the complexity and implementation of the procedure become very hard.

Among the techniques, ‘filling’ gap or inserting and changing priorities share a common ground and the programming of these techniques are comparatively easy than programming delay. Therefore, a combined technique of filling gaps, inserting and changing priority was termed as *swap technique*. The swap technique exchanges the position on the same machine, akin to a changing priority procedure and ‘fills’ gaps like inserting procedure. Another procedure was also incorporated which sorts job on the basis of processing time in ascending order, opposite to the SPT rule. In the following section, the swap technique is applied to the same problem cited earlier in order to check its performance and gauge its performance.

4.4.1 Consideration of Swap technique to SPT

In order to understand how the swap technique works, consider the same example problem (3x3 JSSP) cited earlier in Table 4.2. For Operation O_1, the candidate jobs are J_1 (with processing time as 16), J_2 (with processing time as 15), and J_3 (with processing time as 8). In simple SPT rule, J_3 with processing time of 8 should be assigned first. Instead, J_2 with processing time 15, which is greater than J_3 and lesser than J_1 will be selected and assigned first to the respective machine, followed by J_1 and J_3, will be assigned in the last. Hence, the procedure ignores the job with shortest processing time and assigns rest of the jobs based on SPT rule. The ignored job was then assign last in operation one. A similar pattern is followed for the rest of the operations. The final schedule obtained from this swapping technique with SPT is shown in Figure 4.25.

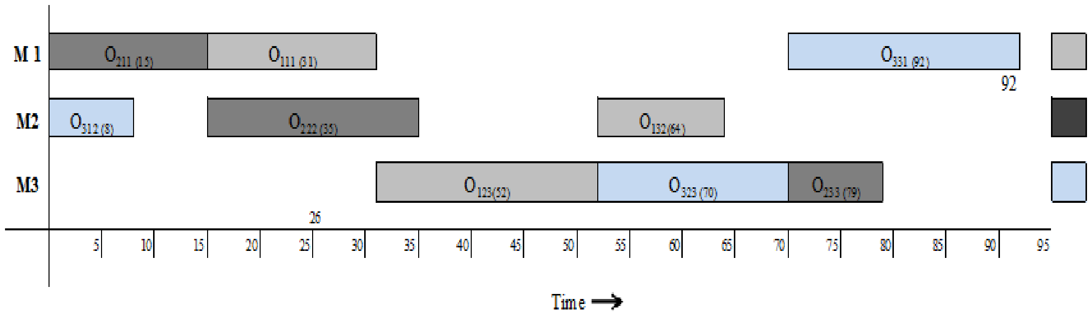


Figure 4. 25: SPT with Swap rule

In Table 4.4 comparisons of the performance of SPT and SPT with the swap is shown. The results for SPT with the swap is obtained through the procedure described and are listed in the third column. The results show that there is an improvement in the Makespan value by 3 units of time. The new Makespan value is 92 with a %GAP of 46.31 from the optimum result.

Table 4. 5: Performance comparison of SPT VS SPT with Swap

Performance parameters	**SPT**	**SPT with Swap**
LB	53	53
Best Known Optimum	63	63
Makespan found by SPT ($\boldsymbol{C_{max}}$)	97	92
Maximum Tardiness ($\boldsymbol{T_{max}}$)	97	92
Total Flow Time ($\sum C_j$)	189	235
Number of Late Jobs	3	3
% GAP with Best Known Optimum	53.96	46.31

The algorithm is allowed to swap again and this time the algorithm ignors the first two job with shortest processing time and prioritizes with the third job with shorted processing time among all candidate jobs. For example consider the same example, for Operation O_1, the candidate jobs are J_1 (with processing time as 16), J_2 (with processing time as 15), and J_3 (with processing time as 8). In simple SPT rule, the sequencing order will be J_3 with processing time of 8 will be assigned first followed by J_2 and then J_3. However, in second iteration of the swap technique first J_3 is assigned followed by J_2 and J_1. The final schedule obtained from this additional iteration is shown in Figure 4.26.

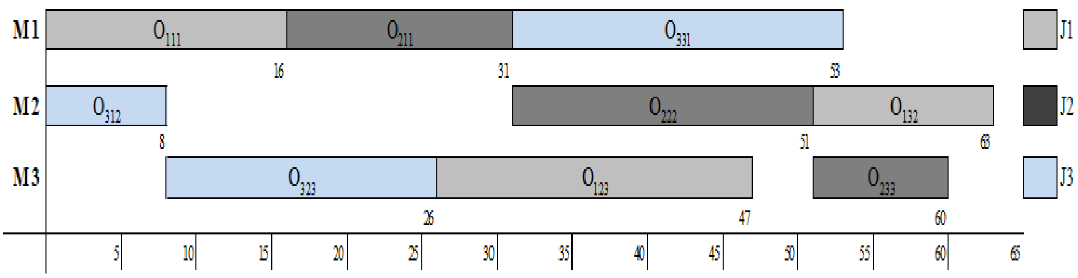


Figure 4. 26: Schedule obtained from swap 2nd iteration

In Table 4.5 comparisons of the performance of SPT and SPT with the swap is shown in 1st and 2nd column. The results for SPT with the swap for 2nd iteration is obtained through a similar procedure and are listed in the 3rd column. The results show that the Makespan in first iteration was 92 with a %GAP of 46.31 from the optimum result and 63 with 0 %GAP from optimum. To conclude, the SPT rule with swapping technique is an effective rule. This technique can be applied in order to achieve better Makespan value because it allows the practitioner or researcher to search in the solution space for a number of iterations equal to the number of jobs. Hence, the chance of achieving optimal or near-optimal solution increases. The computational cost is also very low and this algorithm can easily be adopted to any scheduling problem.

Table 4. 6: Performance comparison of SPT VS SPT with Swap

Performance parameters	SPT	SPT with Swap	SPT with Swap (2nd Iteration)
LB	53	53	53
Best Known Optimum	63	63	63
Makespan found by SPT (C_{max})	97	92	63
Maximum Tardiness (T_{max})	97	92	63
Total Flow Time ($\sum C_j$)	189	235	176
Number of Late Jobs	3	3	3
% GAP with Best Known Optimum	53.96	46.31	0.00

4.4.2 Consideration of Swap technique with Normalized Processing Time

During the analysis it was observed that all the heuristic rules directly take the processing time and do not incorporate the effect of other processing times for remaining of operations. Therefore, it was decided to normalize the processing time and convert them into a range of values from 0 to 1. These values are called IVals in this research. The IVal are then sorted in ascending order and the schedules are developed with the help of these values, which has eventually led to a novel heuristics i.e. Index Based Heuristic (IBH). IBH is explained in detail in the coming sections with and without swap technique.

4.5 Conclusion of the study

In this section, one existing heuristic SPT rule was selected and applied to the same 3 x 3 JSSP for performance analysis and potential improvement. The objective function selected was Makespan minimization. The known Makespan value of the problem is 63 with a LB (calculation shown) of 53 units of time. The SPT initially, achieved a Makespan value of 97 with a %GAP of 53.96 with the best known optimum result. In the process, a swapping technique was applied to the SPT and the performance parameter Makespan was recorded. An improvement was recorded in Makespan when the swap technique was applied with SPT and in second iteration, it found the optimum result. This achievement of the optimum result is due to the characteristics of the swap technique, which actually do a local search with number of iterations equal to the number of jobs. Hence, the swap technique is an effective tool.

During the experimentation and analysis procedure it was observed that the processing based heuristic rules normally do not consider the effect of remaining operation's processing time of the same job. During the brainstorming phase a need for a heuristic arose which should consider the effect of all operation's of a job. This idea led to the development of Index Based Heursitc (IBH) rule. This rule normalizes the processing times and assigns a normalized values or index values to processing times. The rule uses the normalized values and prioritises the jobs on the basis of these values. The effect was studied and improvement was recorded. The swap technique was also tried with IBH in order to do a local search, which yielded improved results.

This IBH rule was developed and implemented alone and with a swap techniques (see Section 4.6.1 for details). The shortcoming of the IBH rule, discussed in the next section led to the development of a combined or Hybrid Heuristic (HybH) Rule. The HybH rule combines the IBH with Finished Job Based (FJB) rule. The success of the benchmark JSSPs of different sizes using these new heuristic rules determinied their final algorithm steps.. A detail performance analysis of both these two heuristics is presented in Chapter 6 (see Sections 6.2 and 6.3).

4.6 Novel Heuristics

The processing time based heuristic rules normally do not consider the effect of remaining operation's processing time of the same job. In the above sections of the development phases, during the analysis and synthesis phase a need for a heuristic arose which could consider the effect of all operation's of a job into account. This idea lead to the development of Index Based Heursitc (IBH) rule. This rule was developed and implemented alone and with a swap techniques (see Section 4.6.1 for detail). The performance of this rule was check against a test-bed of JSSPs of different sizes and hardness (see Section 6.2). The shortcoming of IBH rule discussed in next section lead to develop a Hybrid Heuristic (HybH) Rule which combines IBH with Finished Job Based (FJB) rule (see Section 4.6.2). The performance of HybH is also check against the same test-bed of JSSPs used for IBH.

In the following sections, the procedure for both IBH and HybH is explained in detail with the same example, used to illustrate the earlier algorithms.

4.6.1 Index Based Heuristic Rule

In this section, the Index Based Heuristic (IBH), a novel approach for solving scheduling problems with an objective of minimizing the overall Makespan (C_{max}) is presented. The proposed heuristic calculates the indices, called Index Values (IVal) of the candidate jobs and then assigns the jobs to the available machine in the ascending order of the index values, i.e., jobs with lower index values are assigned first. This assigning process is similar SPT, in that the SPT rule selects the operation on the basis of the shortest processing time whereas, the IBH selects the operation on the basis of lower IVal. To minimize the idle time between jobs, a swap technique is introduced at a later stage if the algorithm initially fails to achieve the optimum value, when all candidate jobs have been assigned. The swap technique takes the candidate jobs for a machine and swaps them without violating the precedence constraint (explained in the next section). Several benchmark JSSPs from the literature are solved in order to check the validity and effectiveness of the proposed heuristic. Results show that the proposed IBH based algorithm has outperformed the traditional heuristics and is a valid methodology for scheduling optimization.

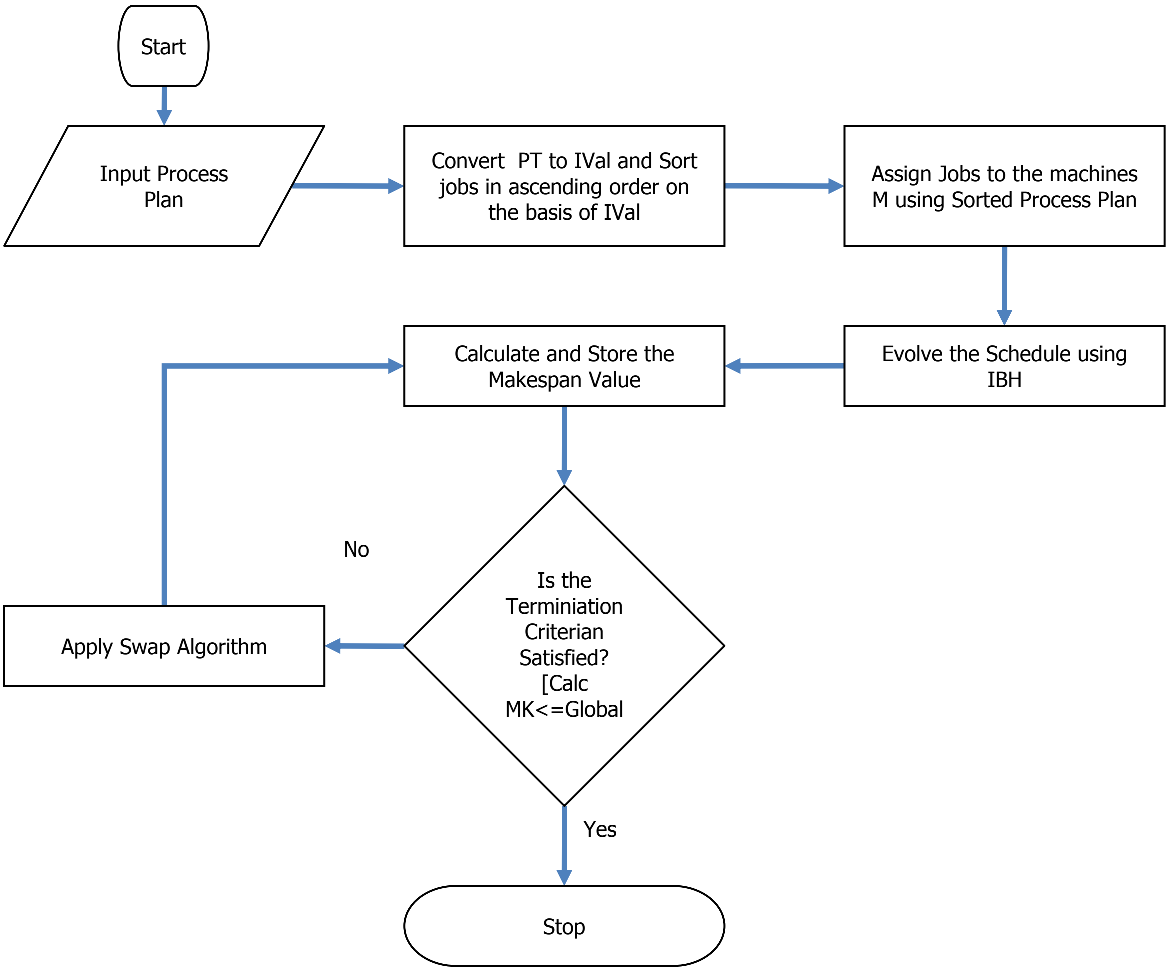


Figure 4. 27: Proposed IBH algorithm for the job shop scheduling problem

Figure 4.27 shows a flowchart which represents the proposed novel IBH algorithm. The IBH algorithm consists of the following steps:

Step 1: The IBH based algorithm for a given processing plan and Processing Times (PT), initially converts the PTs to Index Based Values (IVal) for each candidate job. Once the conversion is completed, the jobs for each operation are sorted in an ascending order with respect to their IVal.

Step 2: The sorted jobs are then assigned to the respective machines in an ascending order for all the operations. For example, the candidate jobs for the Operation O_1 will be assigned on the basis of the ascending order of the IVal, followed by the remaining jobs for operations.

Step 3: When all the candidate jobs on the respective machines are allocated, the algorithm records the maximum time taken by a machine amongst all the machines as the minimum Makespan value.

Step 4: The calculated Makespan Value is then compared with the known Global Makespan Value. If the calculated Makespan is greater than the global Makespan value, the swap algorithm is used to attempt calculate a better schedule. This swap technique swaps two jobs on the same machine without violating the precedence constraint of jobs by selecting the next second lowest IVal (see example in next sub-section).

Step 5: The output data from the schedule for each job is in terms of its PT, start time, end time, waiting time, next machine, idle time and flow time. These outputs are then used to produce the final Gantt chart with the minimum Makespan.

4.6.1.1 Example Problem

A same simple example cited earlier is taken from literature (Gen et al., 2008) of *3-jobs and 3-machines* with Makespan 63, given in Table 4.7, is used to illustrate the IBH. Gen et. al. (2008), have used this example and compared few well known conventional techniques. The result from the IBH is compared with the conventional heuristics rules results.

Table 4. 7: Process plan for three jobs and three machines (3x3)

Process Plan						
Jobs	**O_1**		**O_2**		**O_3**	
	M	**PT**	**M**	**PT**	**M**	**PT**
J_1	1	16	3	21	2	12
J_2	1	15	2	20	3	9
J_3	2	8	3	18	1	22

Table 4.7 shows the process plan for JSSP. For example, J_1 that has three Operations O_1, O_2, and O_3, on M_1 (with a Processing Time (PT) unit of 16), M_3 (with a PT unit of 21), and M_2 (with a PT unit of 12) respectively.

The IBH initially takes and converts the PT to Index Based Values (IVal). Table 4.8 shows the index-based representation of the problem.

Table 4. 8: Index Based representation of process plan

Process Plan: Index Value Based									
	O_1			**O_2**			**O_3**		
Jobs	**M**	**PT**	**IVal**	**M**	**PT**	**IVal**	**M**	**PT**	**IVal**
J_1	1	16	***0.32***	3	21	0.63	2	12	***1***
J_2	1	15	***0.34***	2	20	*0.68*	3	9	***1***
J_3	2	8	***0.16***	3	18	0.45	1	22	***1***

The IVal (which is a normalized value) for any job is calculated by adding all the processing times for the job and then dividing the sum by the processing time of the remaining operations. For example, in J_2, the index value is **0.34** [15/(15+20+9)] for Operation O_1, **0.68** [20/(20+9)] for O_2, and **1** [9/(9)] for O_3.

Table 4.9 shows trace table. In 1st , 2nd and 3rd columns it shows the task, operations for jobs, and respective IVals for each job. The 4th column shows the job, operation number and the machine on which the candidate job is to be processed prioritized by

IBH. In O_{ikj} index i represents job number, k represents an operation number and j represents machines number. 5th column represents the corresponding processing time for each candidate job, and in the 6th column the scheduled operations are shown.

Using trace table (See Table 4.9) for the operation selection process, the schedule can be constructed by using equation 4.2 as follows, and Figure 4.28 illustrates a Gantt Chart showing the schedule for IBH dispatching rule.

Schedule $S = \{(O_{ik\,j}\ (t_{ijk} - t^{F}_{\mathrm{ijk}})\}$ Equation 4.2

Where

t_{ijk} Shows starting time of an Job's (J_i) Operation j on Machine M_{k}.

t^{F}_{ijk} Shows ending time of an Job's (J_i) Operation j on Machine M_{k}.

$$S = \{\ O_{312}(t_{312} - t^{F}_{312}), O_{111(t_{111} - t^{F}_{111})}, O_{211}(t_{211} - t^{F}_{211}), O_{323}(t_{323} - t^{F}_{323}), O_{123}(t_{123} - t^{F}_{123}), O_{222}(t_{222} - t^{F}_{222}), O_{132}(t_{132} - t^{F}_{132}), O_{233}(t_{233} - t^{F}_{233}), O_{331}(t_{331} - t^{F}_{331})\}$$

$$S = \{\ O_{312}(0 - 8), O_{111}(0 - 16), O_{211}(16 - 31), O_{323}(8 - 24), O_{123}(24 - 45), O_{222}(31 - 51), O_{132}(51 - 63), O_{233}(51 - 58), O_{331}(31 - 53)\}$$

Table 4. 9: Trace table for example problem

Task	**Operation Number**	**Index Values**	**Selected Operation to be processes**	**PT for selected operation**	**Scheduled Jobs**
$\boldsymbol{l}$	$\boldsymbol{O_i}$	$\boldsymbol{IVal = V_{ijk} = V_{(job)(operation)(mach)}}$	$\boldsymbol{O_{ikj}}$	$\boldsymbol{p_{ikj}}$	$\boldsymbol{S(l)}$
1	$\{O_{111}, O_{211}, O_{312}\}$	V_{111} = .32 $V_{211} = 0.34$ $V_{312} = 0.16$	O_{312}	8	$\{O_{312}\}$
2	$\{O_{111}, O_{211}, O_{323}\}$	$V_{111} = 0.32$ $V_{211} = 0.34$ $V_{323} = 0.45$	O_{111}	16	$\{O_{312}, O_{111}\}$
3	$\{O_{211}, O_{323}, O_{123}\}$	$V_{211} = 0.34$ $V_{323} = 0.45$ $V_{123} = 0.63$	O_{211}	15	$\{O_{312}, O_{111}, O_{211}\}$
4	$\{O_{323}, O_{123}, O_{222}\}$	$V_{323} = 0.45$ $V_{123} = 0.63$ $V_{222} = 0.68$	O_{323}	18	$\{O_{312}, O_{111}, O_{211}, O_{323}\}$
5	$\{O_{123}, O_{222}, O_{331}\}$	$V_{123} = 0.63$ $V_{222} = 0.68$ $V_{331} = 1$	O_{123}	21	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}\}$
6	$\{O_{222}, O_{331}, O_{132}\}$	$V_{222} = 0.68$ $V_{331} = 1$ $V_{132} = 1$	O_{222}	20	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}\}$
7	$\{O_{331}, O_{132}, O_{233}\}$	$V_{331} = 1$ $V_{132} = 1$ $V_{233} = 1$	O_{132}	12	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}, O_{132}\}$
8	$\{O_{331}, O_{233}\}$	$V_{331} = 1$ $V_{233} = 1$	O_{233}	9	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}, O_{132}, O_{233}\}$
9	$\{O_{331}\}$	$V_{331} = 1$	O_{331}	22	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}, O_{132}, O_{233}, O_{331}\}$

The Makespan achieved by IBH for the problem is 63 as shown in Figure 4.28. Gen et al. (2008), have applied SPT, Longest Processing Time (LPT), Longest Remaining Time (LRT), Shortest Remaining Time (SRT), and LRM rules to the same problem and the reported Makespan values for each of the heuristic rules are 77, 100, 63, 97, and 63 respectively. The SPT, LPT and SRT rules failed to achieve the optimal result of a simple problem, where as IBH performed, well and archived the optimum value. Thus shows that the IBH rule is an efficient and reliable new technique and has a tendency to achieve optimal results.

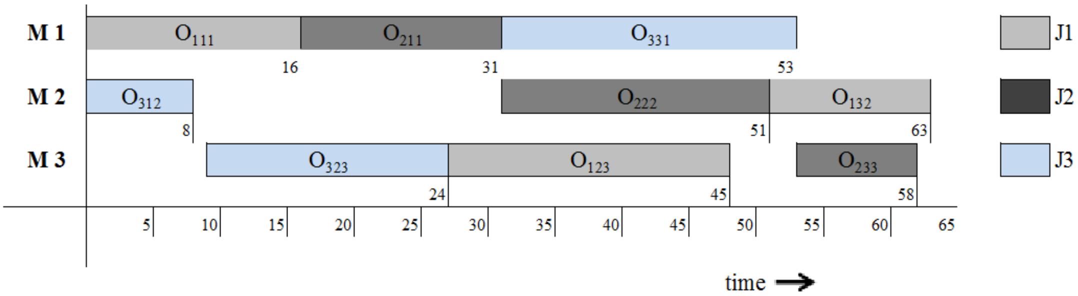


Figure 4. 28: Gantt chart for IBH rule

In case the IBH failed to achieve the Makespan optimum value, the decision module will allow it to generating another schedule by re-constructing a schedule on the basis of the 2nd lower IVal in the assignment of task. The procedure is the same as shown in Table 4.9. For example, instead of an operation O_{312} the schedule will start constructing from Operation O_{111}. This result from the IBH is illustrated in the Gantt chart shown in Figure 4.29. In Figure 4.28, the IBH has achieved the optimal value therefore, the second iteration result is poor than the first iteration shown in Figure 4.29. The IBH algorithm terminates when it reaches global optimum.

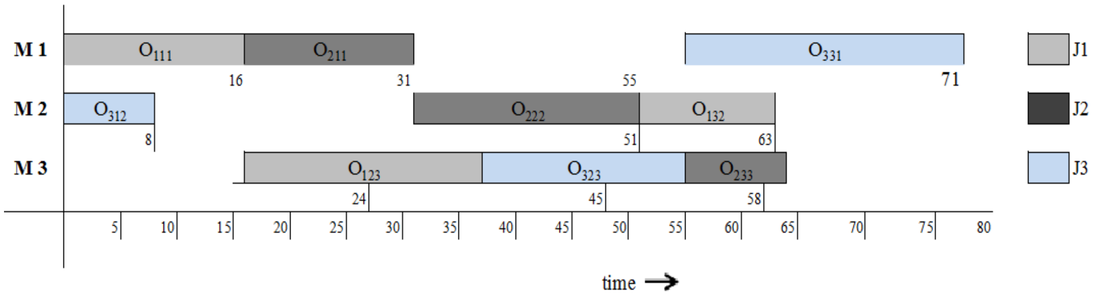


Figure 4. 29: Final schedule after swap technique

The IBH is applied to a set of selected benchmark JSSPs in order to test its performance (see Chapter 6, Section 6.2).

4.6.2 Hybrid Heuristics

The IBH technique alone without swap techniques often yields larger Makespan and with swap technique IBH take longer time in the convergence. Therefore, a Hybrid Heuristic (HybH) solution approach for scheduling problems is developed to overcome the deficiencies of IBH. The proposed HybH is a combination of the IBH and the Finished Job Based (FJB) Heuristic. Various techniques are tried in a hit and trial process, in order to develop a stable hybrid heuristic which would not only yield (near) optimum results but also converge quickly. The HybH or IBH-FJB performed better on different size of the problem (See Chapter 6, Section 6.2.1). The HybH assigns the first operation to a job using the IBH and the remaining operations on the basis of FJB. The FJB gives priority to the job with the earliest finished operations i.e. the first idle job among candidate jobs is prioritized, without violating the constraints of process order. The proposed HybH is explained with the help of a detailed example (see Section 4.6.2.1). Several benchmark problems from the literature are solved to check the validity and effectiveness of the proposed heuristic in Chapter 6.

The algorithm steps that are followed in the proposed heuristic is summarized in the flowchart shown in Figure 4.30. The proposed HybH consists of five steps as follows:

Step 1: For the first operation, assign jobs to machines using IBH.

Step 2: Using IBH for the first operation, attempt different combinations (equal to the number of operations) for the best possible schedule and therefore evolves a schedule for each combination. During a schedule evaluation, record output data such as the job processed, next process due, start time for each process and finish time for each process. When an operation is completed, delete that operation from the list of all possible operations for a job.

Step 3: For the remaining operations, use the HybH to assign jobs to machines using the proposed FBJ schedule. The FBJ takes candidate job and assigns it to the available machine for the next operation, keeping the precedence constraint.

Step 4: Repeat the procedure until all the jobs are processed for all the operations on the basis of the earliest finished time.

Step 5: Find the maximum from amongst the highest of finish times for all the processes, i.e., Makespan.

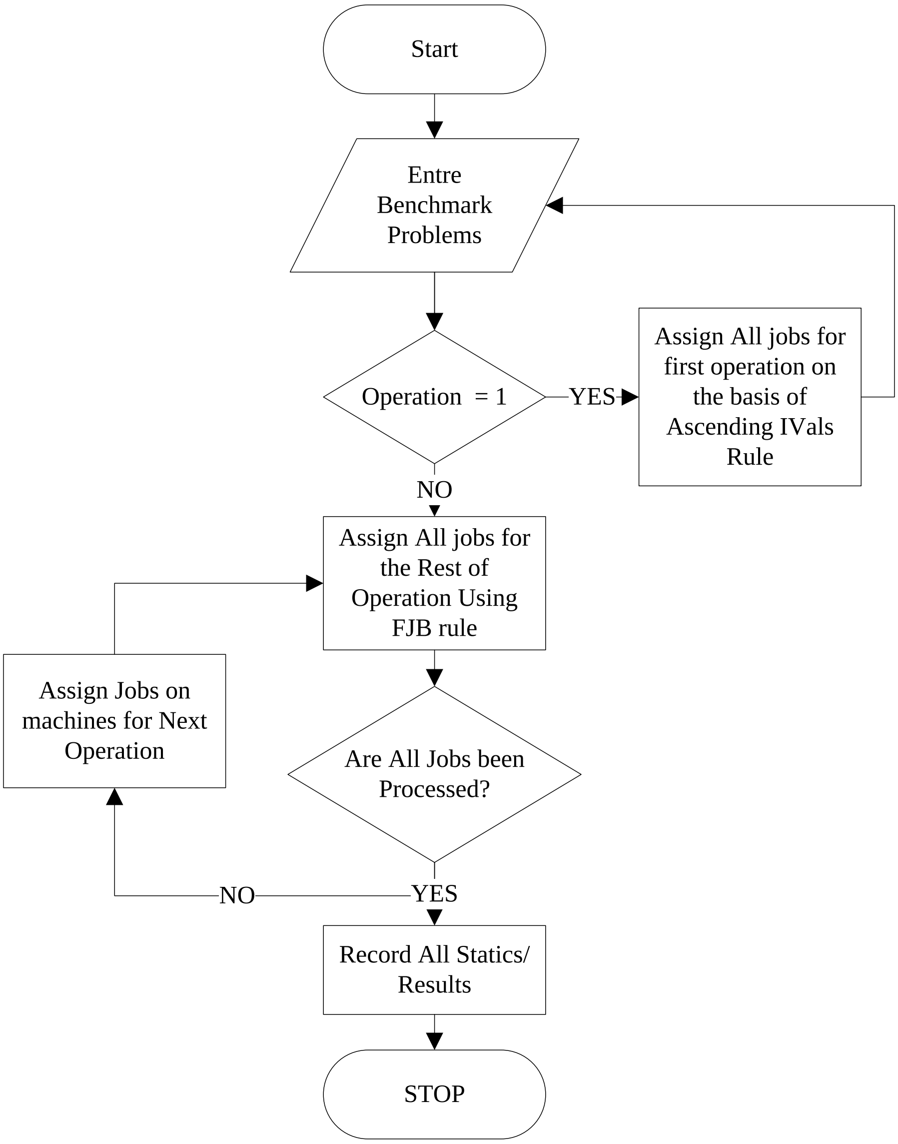


Figure 4. 30: Proposed hybrid heuristic (HYBH) for job scheduling problem

4.6.2.1 Example Problem

Consider the same example used for IBH i.e. 3 jobs and 3 machine JSSP with Makespan of 63, given in Table 4.10 is used to illustrate the HybH. Table 4.9 shows trace table for the example using HybH techniques.

Table 4. 10: Trace table for example problem

Task	Operation Number	Index Values	Selected Operation to be processes	PT for selected operation	Scheduled Jobs
l	O_i	$IVal = V_{ijk} = V_{(job)(operation)(mach)}$	O_{ikj}	p_{ikj}	$S(l)$
1	$\{O_{111}, O_{211}, O_{312}\}$	$V_{111} = .32$ $V_{211} = 0.34$ $V_{312} = 0.16$	O_{312}	8	$\{O_{312}\}$
2	$\{O_{111}, O_{211}, O_{323}\}$	$V_{111} = 0.32$ $V_{211} = 0.34$ $V_{323} = 0.45$	O_{111}	16	$\{O_{312}, O_{111}\}$
3	$\{O_{211}, O_{323}, O_{123}\}$	$V_{211} = 0.34$ $V_{323} = 0.45$ $V_{123} = 0.63$	O_{211}	15	$\{O_{312}, O_{111}, O_{211}\}$
4	$\{O_{323}, O_{123}, O_{222}\}$	O_{312} finished assign Next	O_{323}	18	$\{O_{312}, O_{111}, O_{211}, O_{323}\}$
5	$\{O_{123}, O_{222}, O_{331}\}$	O_{111} finished assign Next	O_{123}	21	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}\}$
6	$\{O_{222}, O_{331}, O_{132}\}$	O_{211} finished assign Next	O_{222}	20	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}\}$
7	$\{O_{331}, O_{132}, O_{233}\}$	O_{123} finihed assign Next	O_{132}	12	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}, O_{132}\}$
8	$\{O_{331}, O_{233}\}$	O_{211}finished assign Next	O_{233}	9	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}, O_{132}, O_{233}\}$
9	$\{O_{331}\}$	O_{323}finished Assign Next	O_{331}	22	$\{O_{312}, O_{111}, O_{211}, O_{323}, O_{123}, O_{222}, O_{132}, O_{233}, O_{331}\}$

The Makespan achieved by HybH for the problem is 63 as shown in Figure 4.31. The result indicates that HybH also has a tendency to achieve optimal results. It can be observed that the result from HybH resemble IBH in this particular example. However, its results from benchmark JSSPs are quite encouraging and discussed in detail in Chapter 6.

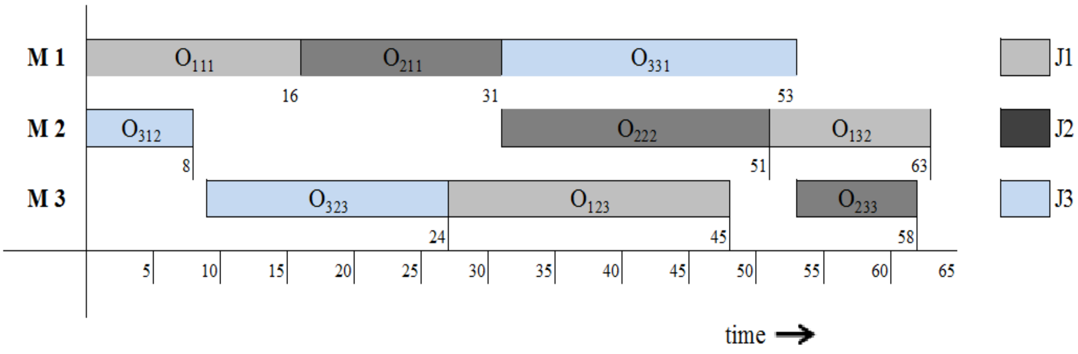


Figure 4. 31: Gantt chart for HybH

4.7 Conclusion

In this chapter, development stages for two novel heuristic rules are presented in detail. In the analysis and synthesis phase, it was also observed that the newlydeveloped procedures yield mostly valid results. However, in the case of larger problems the complexities increase. To resolve these complexities different procedures are combined to evaluate a feasible schedule. For example, the overlapping issue is sorted by introducing the delay technique in the procedure. The logic or algorithm seems very simple. However, to program this logic for new procedures was very difficult and few of the problem are not practically possible to solve.

During the analysis and synthesis phase, it was observed that the swap technique is easy to implement and can be programmed easily. Experimentation on the swapping technique showed improvement in the Makespan value and was also the most effective of all the procedures tried in the analysis phase. During the experimentation phase , it was also observed that the existing rules do not take the effect of other operations or their processing time into consideration. Thus, led to the concept of normalization and finally IBH. This IBH showed encouraging results and with the introduction of the swapping and FJB techniques, it outperformed many existing heuristic rules.

The chapter has covered the development of these heuristic rules is explained with the help of examples. However, a detailed performance analysis is carried out on a test bed of JSSPs with different sizes and hardness in Chapter 6 (see Sections 6.2 and 6.3). The proposed heuristic rules overcome the deficiencies in the traditional existing heuristics for manufacturing process scheduling.

In this research, only the HybH has been used in the main GA loop because of the reason that HybH performed better than IBH on benchmark problem JSSPs. The HybHis used in the evaluation process and the calculating initial solution of the benchmark problems. The evaluation process in a GA for the JSSP is a key step that determines the fitness of the objective function. The IBH can also be applied in combination with GA for the evaluation process. Therefore, future work shall focus on hybridization of IBH with other optimization techniques. The IBH shall also be applied to some larger size benchmark problems and real scheduling scenarios.

CHAPTER 5

HYBRID GENETIC ALGORITHM FOR JOB SHOP SCHEDULING PROBLEMS

5.1 Introduction

The Job Shop Scheduling Problem (JSSP) is a difficult combinatorial optimization problem. In the past, exact methods have been used to provide optimal solutions for scheduling problems. However, these methods are very expensive and difficult to solve in real time, especially in larger scheduling problems where the computational complexity grows exponentially. Therefore, in the past four decades, researchers have been developing novel intelligent optimization techniques in order to solve these types of problems. Despite the recent progress in optimization techniques, there are still many instances in which intelligent optimization techniques become trapped in local minima. Therefore, there is a need to develop algorithms that effectively explore and navigate the solution space and provide optimal solutions. This chapter highlights a novel heuristic based Genetic Algorithm (GA) or a Hybrid Genetic algorithm (HGA) with the aim of achieving optimal or near-optimal solutions for benchmark JSSPs. The chapter also presents the detailed GA approach used to encode the JSSP, different genetic parameters and parametric analysis (sensitivity analysis) for a wide range of benchmark JSSPs. The results from the HGA evaluation and their analyses are provided in the final section.

5.2 Genetic Algorithms (GAs) for scheduling

A GA has been one of the most popular optimization tools and is capable of being applied to an extremely wide range of problems (Goldberg, 1989; Gen et al., 2008; Low and Yeh, 2009; Pan and Huang, 2009), although some researchers have used GAs with an experimental perspective. According to Low and Yeh (2009), a GA is especially suitable for combinatorial optimization problems since it can simulate more phenomena of living systems than any other evolutionary algorithm. Moreover, unlike other popular conventional search techniques that start a global search from only one initial point and search sequentially, the GA starts its global search simultaneously from many initial points or a set of initial solutions called a population, satisfying boundary and/or system constraints to the problem (Gen et al., 2008; Xu et al., 2011). Hence, compared with the other optimization techniques, a GA probably has the highest possibility of reaching the global optima in a defined time interval and makes the best compromise between solution effectiveness and efficiency. However, in the case of NP-hard scheduling problems such as the JSSP, it is widely accepted that any single tool may not be able to find optimal or near-optimal solutions.

5.3 Hybrid Genetic Approach to the Job Shop Scheduling Problem

From the review of each AI techniques in Chapter 3, it was concluded that each technique has strengths and weaknesses in scheduling NP-hard problems. Therefore, many recent works are based on hybrid frameworks consisting of GAs which have performed better than simple GAs. The GA can be hybridized either with another AI technique or with a conventional heuristic algorithm (Noor, 2007). In Chapter 3, hybrid genetic algorithms based on heuristics, local search and AI for JSSPs were also reviewed in detail.

From the literature review, it was concluded that solutions to deterministic JSSPs depend on the significance of selecting the best heuristic rules and meta-heuristic techniques with few or no assumptions about the problem and which can search very large spaces of candidate solutions. The novel heuristic rule HybH and meta-heuristic tools such as the GA have been proposed as search tools for the hybrid model. Although a GA is simple to describe and program, its behavior can be complicated due to its exploitation features. GA solutions are based on various elements that are not separated, but enclosed within the original elements. The proposed HybH rule is used to generate the initial solutions and store the chromosomes with the fittest value in the initial solution pool. The computational experiments showed that the HybH yield fitter the initial solutions compared to other conventional heuristic rules (See Chapter 6 for results and performance analysis) which enables GA to provide better overall solutions.

5.4 Development of Hybrid Genetic Algorithm for the JSSP

The GA is hybridized by introducing a novel heuristic HybH into the loop of the GA as shown in Figure 5.1. The development of the proposed HybH has been covered in Chapter 4 in detail. It has outperformed the traditional heuristics and produced comparatively stable results across the benchmark JSSPs [Chapters 6 and (Maqsood et al., 2011)]. The HybH is incorporated in the GA loop in such a way that it evolves and records each generation's best chromosome, schedule and Makespan.

The process terminates when the solution reaches the best known Makespan value or reaches the set number of generations. This procedure for *HGA* is explained with the help of a flow chart in the following steps that are further explored in the sections to follow:

i. Initialize the population randomly.
ii. Decode each solution and calculate its fitness value using the HybH.
iii. **IF** the initial search (HybH) achieves the best known Makespan value **THEN** Stop, **ELSE** repeat for number of iterations equal to the number of jobs.

iv. **IF** the initial search terminates **THEN** place the fittest in the initial chromosome in the population pool.

v. Select the best chromosomes for crossover. The best chromosome is the one which is with a best Makespan value among chromosomes generated in the initial solutions by HybH and stored in the population pool.

vi. **IF** children are illegal **THEN** repair, **ELSE** go to next step.

vii. Evaluate the children and place them into the population.

viii. Randomly select a chromosome and carry out mutation.

ix. Evaluate the mutated chromosome and place it into the population.

x. Select the next generation by Stochastic Universal Sampling (SUS).

xi. Select the best chromosome of the generation.

xii. Evaluate each recorded chromosome using the HybH and record the events and statistical data for the Number of Parts, Number of Operations, Arrival Time, Waiting Time, Start Time, Processing Time, Machine Idle Time, Finish Time and the Next Machine on which the job is to be processed.

xiii. Using the data obtained in Step xii, record the Makespan (C_{max}) of each machine.

xiv. Add Increment to the value of 'Gen = K' (Gen ← Gen + 1).

xv. **IF** Gen < Max Gen (Maximum number of generations, i.e., 200 discussed in Chapter 6), **THEN** repeat Step v to Step xiv, **ELSE** go to the next step.

xvi. Stop.

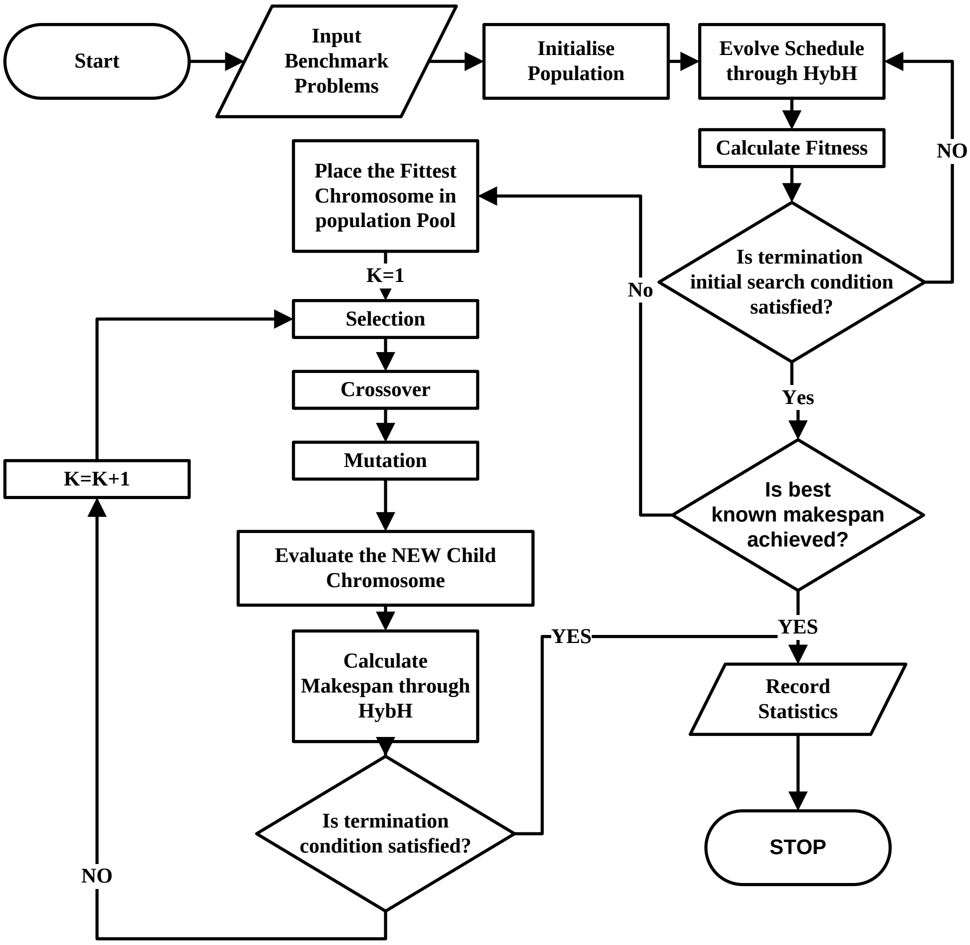


Figure 5. 1: Proposed hybrid Genetic Algorithm

5.4.1 Genetic Algorithm (GA)

This section covers the concepts of GA coding, representation of a JSSP in GA and its evaluation strategy with the help of examples.

5.4.1.1 Chromosome Representation and Initialization

Because of the existence of the precedence constraints of operations, JSSP is not as easy as the Traveling Salesmen Problem (TSP) to find a representation (Gen et al., 2008). A key step in building a GA for JSSP is to devise an appropriate representation of solutions together with problem-specific genetic operations in order that all chromosomes generated in either initial phase or evolutionary process will produce feasible schedules. This is a crucial phase that conditions all the subsequent steps of GAs. In the past twenty years, the following nine representations for JSSPs have been proposed:

- Operation-based representation
- Job-based representation
- Job pair relation-based representation
- Preference list-based representation
- Priority rule-based representation
- Completion time-based representation
- Random key-based representation
- Disjunctive graph-based representation
- Machine-based representation

The sequence and precedence constraint among operations for a job must be maintained in the schedule. According to Gen et al. (2008), in a job-based representation a list of n jobs and a schedule is constructed according to the sequence of jobs. For a given sequence of jobs, all operations of the first job in the list are scheduled first, and then the operations of the second job in the list are considered. The first operation of the candidate job is allocated as the best available processing time for the corresponding machine that the operation requires, then the second operation, and so on until all operations of the job are scheduled. The process is repeated with each of the jobs in the list considered in the appropriate sequence. Any permutation of jobs corresponds to a feasible schedule. If there are n jobs, the permutations of n will give the job sequence. For example, if there are three jobs (1,2,3), one of the chromosomes is represented as [2 3 1] which is a permutation of 3 jobs. Many researchers (listed in Table 3.4) have used this representation for static scheduling.

2	**3**	**1**

Figure 5. 2: An example of Job-based Chromosome

Consider the same three-job three-machine problem given in Table 4.1. Suppose a chromosome is given as shown in Figure 5.2, where 1 stands for Job J_1, 2 for Job J_2 and 3 for Job J_3. The first job to be processed is Job J_2 with the operation precedence constraint for J_2 is [M1, M2, M3] and the corresponding processing time for each machine is [15, 20, 9]. First, Job J_2 is scheduled as shown in Figure 5.3a. Then Job J_3 is processed. Its operation precedence among machines is [M2, M3, M1] and the corresponding processing time for each machine is [8, 18, 22]. Each of its operations is scheduled in the best available processing time for the corresponding machine the operation required as shown in Figure 5.3b. Finally, Job J_1 is scheduled as shown in Figure 5.3c.

5.4.1.2 Legality and Feasibility of Chromosomes

During the chromosome generation process, it is very important to look into the legality and feasibility of each chromosome. Legality refers to whether or not the chromosome represents a solution to the problem; feasibility refers to whether the chromosome gives a feasible solution to the problem when decoded. According to Cheng et al., (1996), any permutation of jobs corresponds to a feasible schedule and in this representation every chromosome is a permutation of n as discussed in an earlier section, therefore there are no legality or feasibility issues. The chromosome can be easily decoded to an active schedule through any heuristic/dispatching rule. In this research, chromosomes are decoded by the developed HybH within GA loop, explained in the next Section.

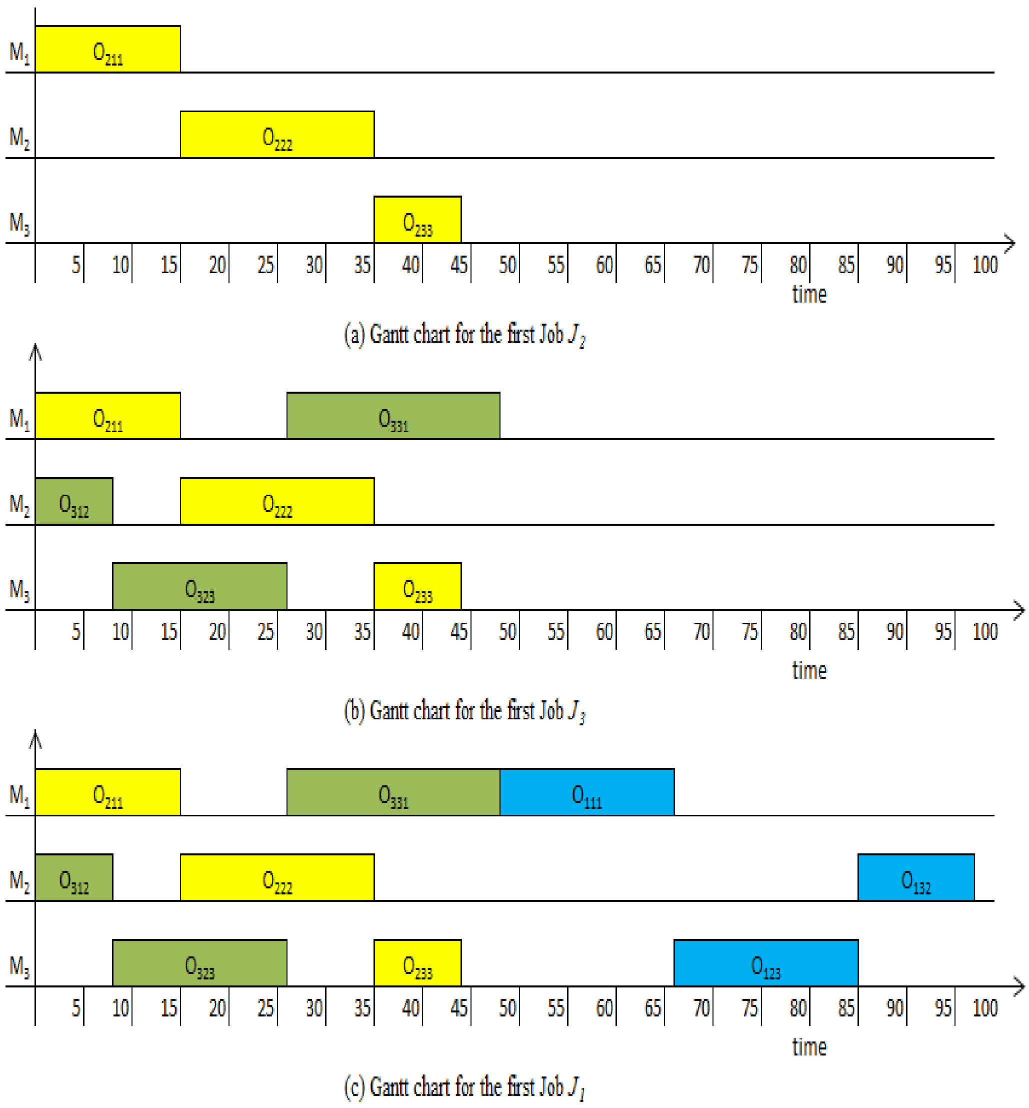


Figure 5. 3: Gantt chart for job-based representation (Gen at al., 2008)

5.4.1.3 Evaluation and fitness

For the purpose of evaluation, the Makespan (C_{max}) is chosen as the measure of performance or fitness of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be used in the reproduction process. Each chromosome is decoded using the novel heuristic

HybH and its Makespan (C_{max}) is calculated. The evaluation procedure employed during this research is shown in Figure 5.4 and explained in the form of a stepwise procedure as follows:

i. Start by reading the benchmark problem for processing time and sequencing order.
ii. Convert the obtained data to IVals and sort the job order for the HybH.
iii. Identify the candidate job, its operation number and the machine on which the operation is to be performed.
iv. **IF** the candidate job is for the operation O_1 on the machine **AND** also has the lowest IVal, **THEN** assign that job to the respective machine with an assumption of arrival time value as zero, followed by the next lower IVal job to be assigned until all jobs for O_1 are assigned, **ELSE** go to step 6.
v. Record the statistical data for the first operation as discussed in Section 5.4, in 9 columns, and using this data, assign the rest of the jobs on the basis of FJB heuristic.
vi. Calculate the overall completion time for all the operations and determine the maximum completion time (C_{max}) from amongst the completion times from individual machines.
vii. '**IF** *Calculated* $C_{max} > C_{max}$, **THEN** $i \leftarrow i + 1$, **ELSE** record *Calculated* C_{max} and Go to Step x.
viii. **IF** $i <$ Population Size, **THEN** repeat Steps iii through vii, **ELSE** go to the next step.
ix. Makespan $\leftarrow C_{max}$.
x. Stop.

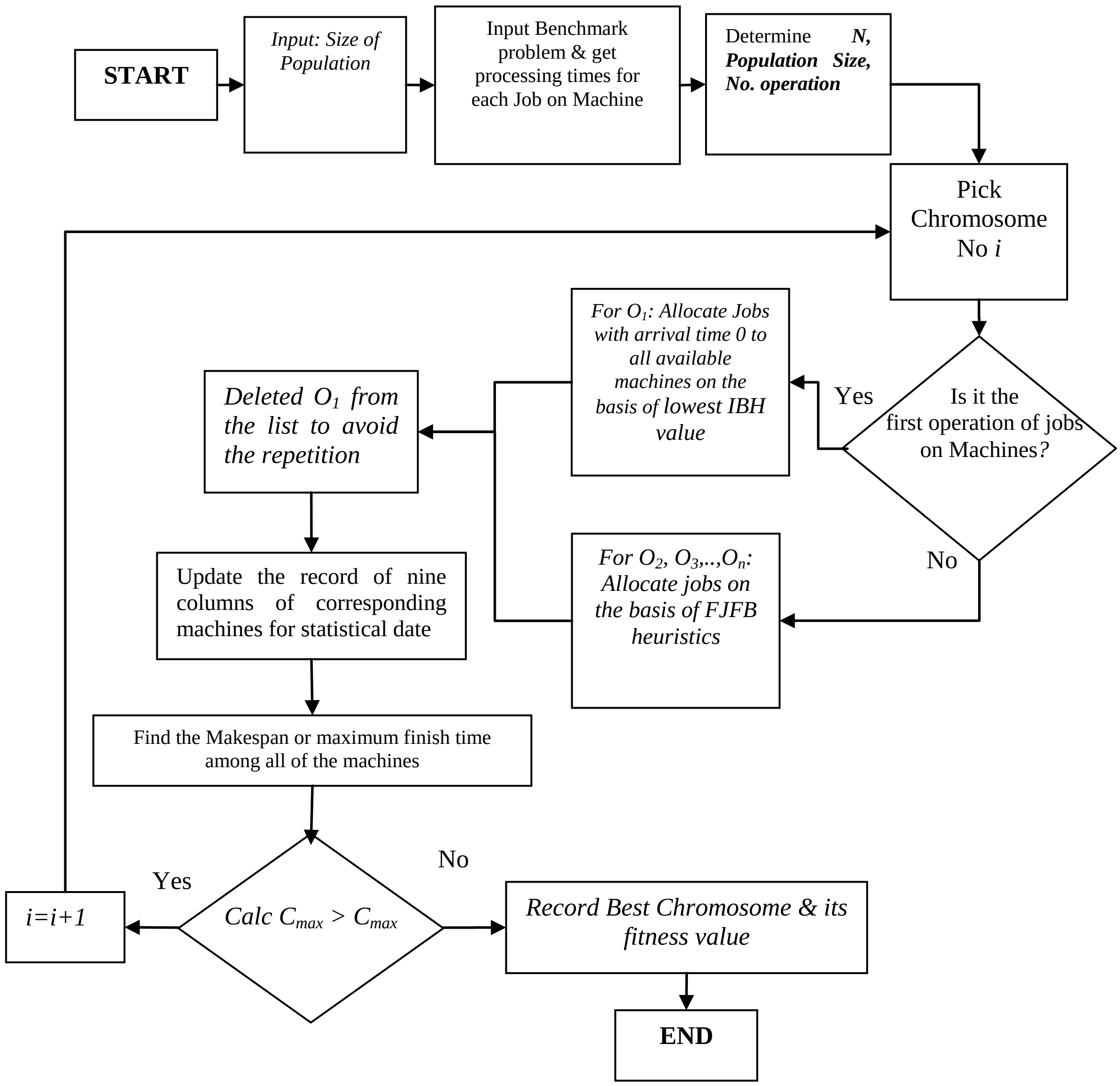


Figure 5. 4: Evaluation Procedure

5.4.1.4 Initial Population

A GA is a parallel search tool and requires an initial set of chromosomes in order to start the search. In the published literature, various heuristic or dispatching rules such as the FIFO, SPT are used to generate the initial set of chromosomes. In this research, HybH rule is used to generate the initial set of solutions as discussed in Chapter 4 and (Maqsood et al., 2011). The total number of chromosomes created by using the HybH is equal to the number of jobs. From the results of the HybH rule (see Chapter 6, Section 6.2.1), it can be seen that it has outperformed the traditional heuristics and performed well across a range of problems, producing high quality solutions. Therefore, seeding these initial populations will yield high-quality solutions and experiments have shown it helps the search tools in finding optimal or near-optimal solutions.

5.4.1.5 Selection

In literature, there are many selection techniques based on the Darwin's Theory of Evolution for the selection of parent from a finite number of chromosomes. The parent chromosome is selected with a probability related to their fitness. Highly fit chromosomes have a higher probability of being selected for mating than the less fit. In this research, Stochastic Universal Sampling (SUS) technique developed by Baker (Baker, 1987) is used for selection of choromsome because in the reproduction of offspring SUS exhibits no bias and minimal spread. It uses a single random value to sample all of the solutions by choosing them at evenly spaced intervals, which simply means that this gives a chance to poor members of the population (on fitness basis) to have a chance to be chosen. Hence, SUS helps in reducing the unfair nature of fitness-proportional selection methods (Baker, 1987). Recently, Chipperfield et al. (2011) and Pohlhein (2009) have used SUS in their work and applied it to JSSPs. Other methods like a roulette wheel performs poorly if the population has a really large fitness in comparison with other members.

Using a comb-like ruler, SUS starts from a small random number, and choose next candidates from the rest of population remaining, not giving the fittest members to saturate the candidate space. In the SUS technique, initially the chromosomes with their fitness values obtained from the HybH are mapped over a line. Then, equally spaced pointers are placed for the selection of N chromosomes. The distance between the pointers is given by 1/N and the position of the first pointer is given by a randomly generated number in the range [0, 1/N] (Noor and Khan, 2007).

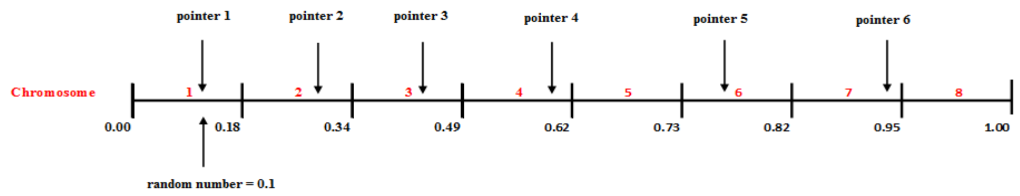


Figure 5. 5: Stochastic Universal Sampling

Consider the random number in the range [0, 0.167] to be 0.1, and the first pointer falls on 0.1, which is between 0.0 - 0.18 as shown in Figure 5.5. Thus, select Chromosome No. 1. Then the next pointer will fall on 0.267 (0.1+0.167), between 0.18 - 0.34, and hence select Chromosome No. 2, and so on. The chromosomes selected through the SUS method will be {1, 2, 3, 4, 6, and 8} as shown by the pointers.

5.4.1.6 Crossover

Once a pair of chromosomes is selected for mating from the current population, the crossover operator is used for the reproduction of the child chromosomes for the next generation. There are various types of crossover operators as discussed in Chapter 3 (Section 3.6). In this research, due to the job-based representation a *job-based crossover technique* proposed by Braune et al. (2005) is applied to the selected parent chromosomes for the production of offspring or child chromosomes with different identities from those of the parents as shown in Figure 5.6. Noor (2007) applied the same crossover technique in his research.

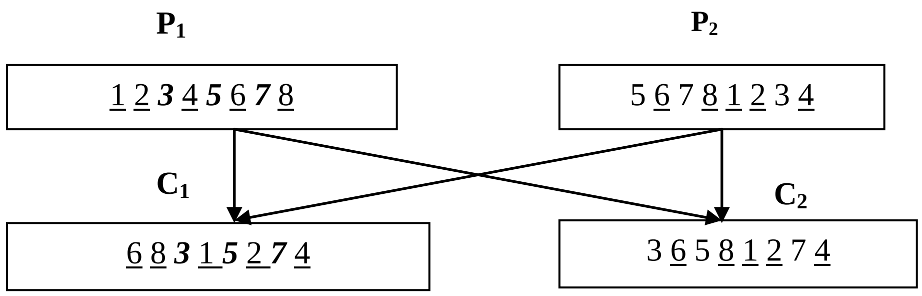


Figure 5. 6: Job Based Crossover Scheme

In the crossover process, initially two parents (P_1 and P_2) are randomly selected. The genes (≤total number of gene/2) are randomly selected, which will be preserved in the child from one of the parents and the remaining genes will be replaced by those from the second parent. For example Genes 3, 5 and 7 have to be preserved in the child and are copied to the offspring chromosome Child C1 in the same absolute positions as in its Parent P_1. The remaining vacant positions will be copied from the Parent P_2 and are shown as underlined. A similar procedure is adopted for the Child C_2.

5.4.1.7 Mutation

The crossover and mutation operators of the GA complement each other for the effective exploration of the solution space. For the mutation operation, a randomly selected gene (but not the last one) is exchanged with the next adjacent gene. This kind of the mutation process suits the job-based representation and yields a feasible mutated offspring. For example consider Child C_1, where the genes or Job 5 and Job 1 are randomly selected and are then mutated as shown in Figure 5.7.

6 8 3 **<u>5</u>** **<u>1</u>** 2 7 4 ⟹ 6 8 3 **<u>1</u>** **<u>5</u>** 2 7 4

Figure 5. 7: Mutation Process

The process continues until 10% randomly selected genes of the total number of genes in the population get mutated.

5.4.1.8 Evaluation and selection of the final solution

For each offspring chromosome, the HybH is applied to record the final statistical data for generating the Gantt chart and records the Makespan value. The process will terminate once the condition is satisfied, i.e., either the calculated Makespan value equals global Makespan or the HybH runs for the maximum number of generations.

5.4.2 Sensitivity analysis:

The performance of the GA technique is mostly dependant on a few critical parameters such as the number of generations, population size and crossover and mutation rates. No technique is yet known to find the best combination of the parameter set for optimum output of the a GA (Maqsood et al., 2011). However, exploring more solution space or, in other words, large numbers of population and generation, tend to provide the optimal or near-optimal solutions at high computational costs, with a large number of these two characteristics, it will be very difficult to find the best combination of crossover and mutation probability. For these reasons, the generation number and population size for all the benchmark problems have been taken as 100 and 50 respectively, based on literature recommendations (Morshed, 2006; Tariq, 2008). These suggested values have not only reduced the computational costs but have also helped in finding the best possible combinations of Crossover Rates (XR) and Mutation Rates (MR). The best combinations of XR and MR are very important parameters which enable schedulers to save time in producing schedules.

To carry out the sensitivity analysis of the developed HGA, a set of benchmark problems is selected, from literature as shown in Table 5.1.

Table 5. 1: Selected Benchmark problems from literature

Problem Code	**Source**	***Instances Machs×Jobs***	**Known Optimum Value**
FT 06	Fisher and Thompson, (1963)	6 x 6	55
FT 10	Fisher and Thompson, (1963)	10 x 10	930
LA 01	Lawrence, (1984)	10 x 5	666
LA 06	Lawrence, (1984)	15 x 5	926
LA 11	Lawrence, (1984)	20 x 5	1222
LA 12	Lawrence, (1984)	20 x 5	1039
LA 26	Lawrence, (1984)	20 x 10	1218
LA36	Lawrence, (1984)	15 x 15	1268

For each experiment, a wide range of the Crossover Rates (XR) of 0.1 to 1.0 and Mutation Rates (MR) of 0.01 to 0.10 are taken. The experimental results are shown in bar charts in Figures 5.8 to 5.15. Each bar chart represents the MR on x-axis and the Makespan on y-axis. As it can be seen in Figures 5.8 to 5.15, the HGA has achieved the global Makespan value in most of the cases. For cases FT06, LA06, LA11 and LA12, the GA has achieved the optimal values. However, the XR-MR combinations for the cases LA06, LA11, and LA 12 show no effect. This is due to the fact that LA06 and LA12 are computationally easy problems. Thus, the HybH has found the optimal values of Makespan for LA06 for each of the cases and the algorithm terminates, and therefore, there is no role of the GA and its parameters.

The other reason is that the type of operator used for the crossover, mutation and selection suits the LA11 and LA12 problems and consequently the solution rapidly converges (see results in Figures 5. 8 and 5. 9). For difficult problems like FT06, FT10 and LA01, the XR-MR combinations play their role. The XR-MR combinations (0.6, 0.7 – 0.03, 0.04) achieve minimum Makespan values as shown for all the cases.

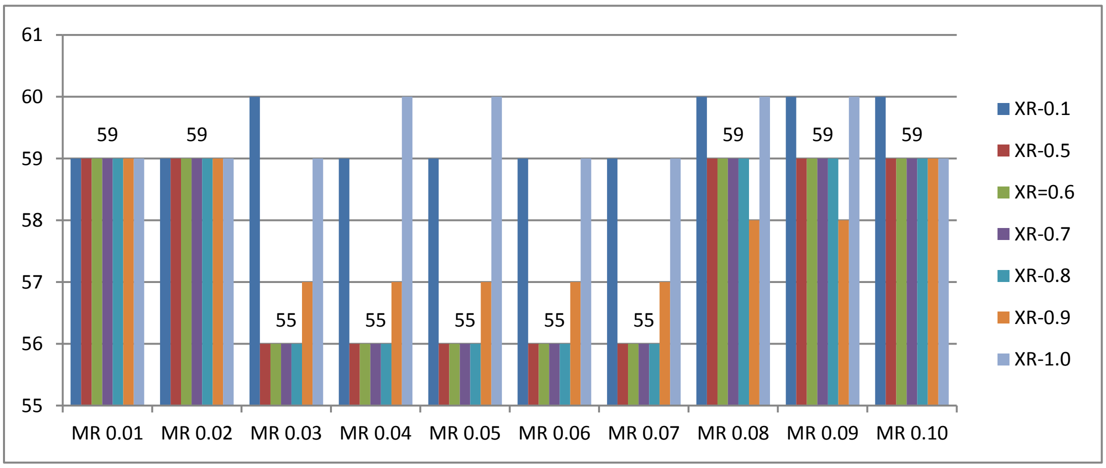


Figure 5. 8: FT 06 benchmark problem [Optimum = 55]

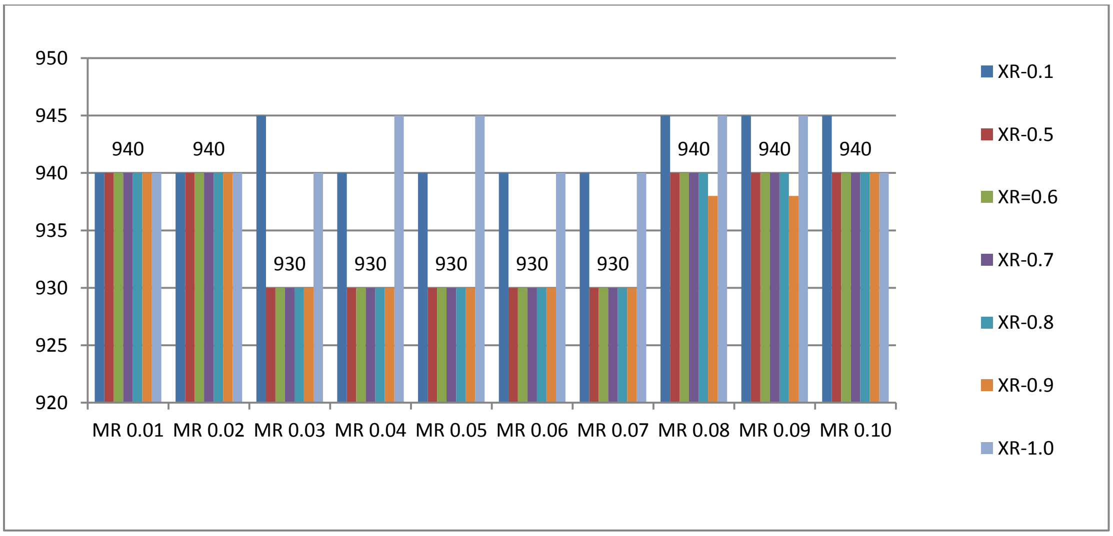


Figure 5. 9: FT 10 Benchmark Problem [Optimum = 930]

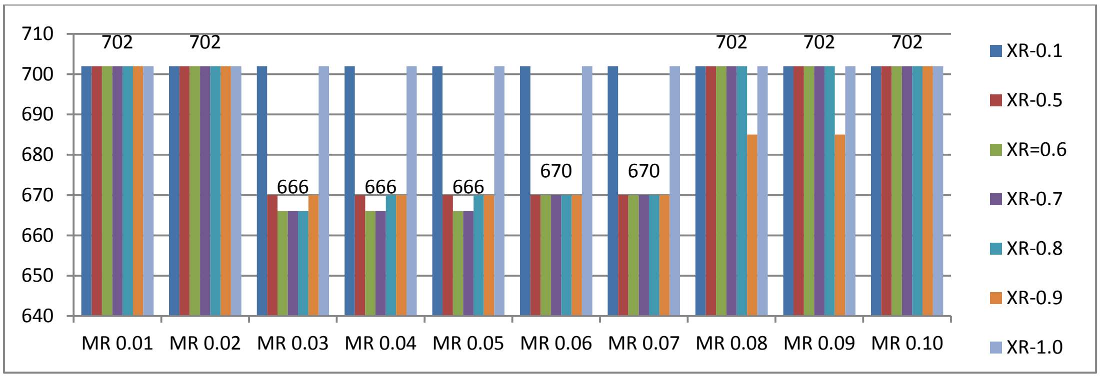


Figure 5. 10: LA 01 Benchmark Problem [Optimum = 666]

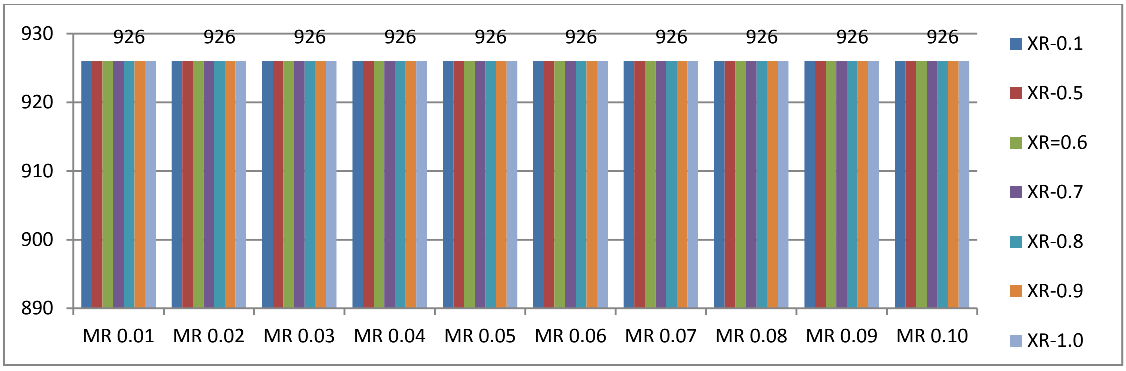


Figure 5. 11: LA 06 Benchmark Problem [Optimum = 926]

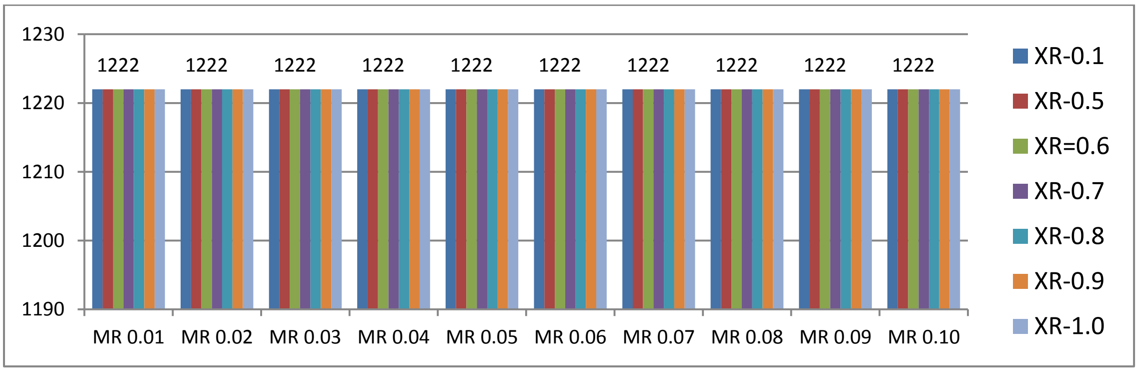


Figure 5. 12: LA 11 Benchmark Problem [Optimum = 1222]

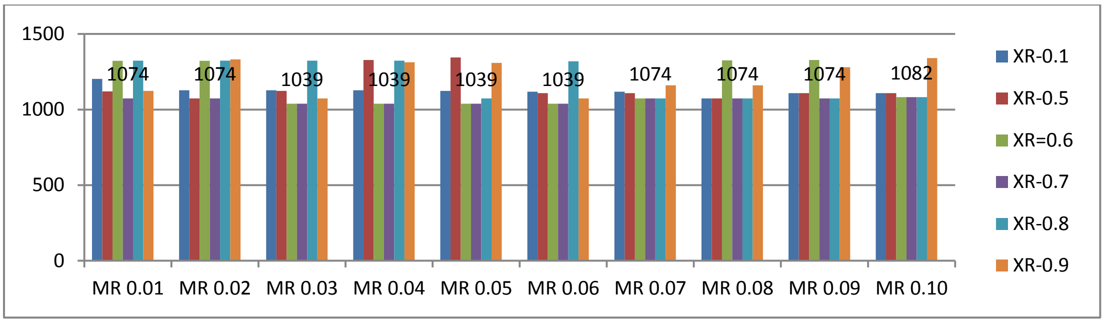


Figure 5. 13: LA12 Benchmark Problem [Optimum = 1039]

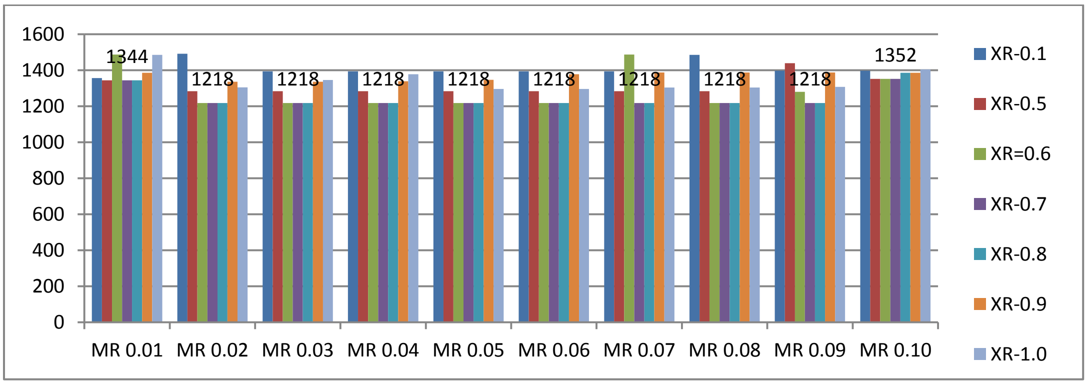


Figure 5. 14: LA26 Benchmark Problem Optimum [1218]

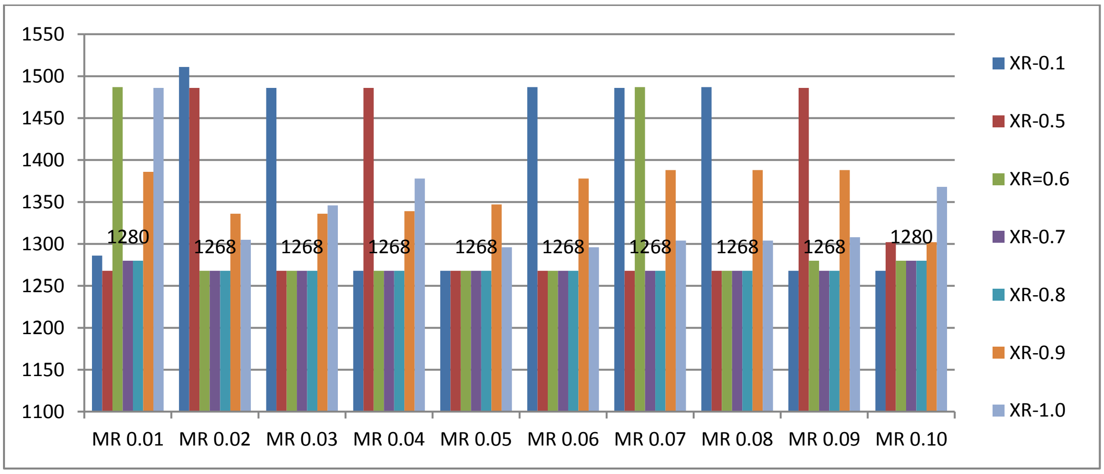


Figure 5. 15: LA 36 Benchmark Problem [1268]

To decide which parameter combination should be used in GAs is a very difficult job. Although, the sensitivity analysis has shown that the GA can provide some good results on certain range combination such as at XR-MR combinations (0.6, 0.7 – 0.03, 0.04) the algorithm is achieving minimum Makespan values. Therefore, it is recommended that further experimentation is required with more benchmark problems and various types of genetic operators.

5.5 Important Features of HGA

The developed HGA can be applied to Flow Shop Scheduling Problems (FSSPs) along with JSSPs. The performance of HGA on both these scheduling problems has been covered in detail in Chapter 6 by testing benchmark JSSPs and FSSPs. The only difference in both the environments is with regards to the flow pattern (discussed in detail in Chapter 2). The HGA reads the processing time and the process plan from spreadsheets. The only difference in spreadsheets is the data set with different flow patterns. For example, consider a simple 3 jobs and 3 machines problem as shown in Table 5.2. The Table 5.2a shows a typical JSSP flow pattern whereas Table 5.2b shows a typical FSSP flow pattern. The HGA can read both the patterns with the corresponding processing from spreadsheets and can produce active schedules and optimal values.

Table 5. 2a: Typical JSSP Flow Pattern

Jobs	O_1	O_2	O_3
J_1	M_1	M_3	M_2
J_2	M_2	M_3	M_1
J_3	M_3	M_1	M_2

5.2b: Typical FSSP Flow Pattern

Jobs	O_1	O_2	O_3
J_1	M_1	M_2	M_3
J_2	M_1	M_2	M_3
J_3	M_1	M_2	M_3

In Chapter 6, the results of HGA for both types of problems (JSSP and FSSP) have been discussed in detail.

5.5.1 Statistics for Each Machine

The HGA is designed in such a way that it can produce and record statistics in the evaluation process for the initial solution and the final active schedule. It provides the status of each machine for each job. These statistics are very useful and various charts such as the Gantt Chart can easily be constructed.

5.5.2 Modular Approach

A modular approach has been adopted in the code development. Various functions have been first generated and tested independently and then incorporated in the main GA loop. This means that any function can be replaced by another function. For example, the function *chromosome.m (MATLAB function)* is a job-based representation of chromosomes. This function is called during the main loop of the GA. This function can easily be replaced by operation based representation function (MATLAB function) with the same variables and function name as *chromosome.m.* So instead of job-based representation function, the main GA loop can call operation-based representation function.

5.6 Summary

This chapter proposes a heuristic based hybrid GA or HGA for JSSPs. The operational performance of the HGA was tested in detail with various sizes of benchmark problems in order to provide a reference for future research in this area and to fill the gap for a parametric analysis or sensitivity analysis for GAs. The GA in combination with a hybrid heuristic for the evaluation of the schedule performed well across the benchmark test-bed problems. The HybH was used for the initial evaluation of job shop schedule from which the fitness value (Makespan) was determined and used for final evaluation of the schedule after the GA application. The novel combination of a HybH and a GA is a contribution in this area of research. The HGA results (%GAP between the calculated Makespan values and Global Makespan Values) were as low as zero across the test bed for most of the benchmark problems and 36 industrial case studies (see Chapter 6, Section 6.5).

This chapter also discussed the study of XR-MR combinations for JSSPs in order to achieve optimum combinations of these two parameters, while keeping the generation number and the population size constant. The findings from the parametric study of the best possible XR-MR combinations were then used in the HGA. The results (optimum XR-MR values) achieved from the study can be used as a platform for the selection of the XR-MR combinations in the optimization of JSSPs. In the future, it is recommended that the similar sensitivity analysis procedure should be carried out, while applying GA to real scheduling problems in order to find more cost-effective XR-MR combinations for GAs.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Introduction

This chapter presents the results from the two novel heuristic rules, the HybH and the IBH, which were discussed in Chapter 4 for job shop scheduling problems. It also presents the results of the Hybrid Genetic Algorithm (HGA) for job shop scheduling problems developed during this research as described in Chapter 5. To determine the strengths of the proposed heuristics and the HGA, they are applied to a computational test-bed consisting of benchmark job shop and flow shop scheduling problems of various sizes and hardness. The developed HGA is also compared with other models developed in the literature. The details and sources of the selected benchmark job shop and flow shop scheduling problems, case studies, their solutions, analysis of results and comparisons with other techniques are also described in this chapter.

6.2 Performance Analysis of Novel Heuristics

The developed novel heuristics are tested against published benchmark scheduling problems to gauge the strengths and comparative merits of these methods. These benchmark problems are developed by various researchers and are listed and reviewed in Chapter 3. In this section, the selected benchmark instances, FT (06, 10) and LA (01, 06, 11, 12, 26 and 36), are used as *test-beds* to check the performance and gauge the effectiveness of the developed heuristic rules compared with those of the traditional heuristics. Table 6.1 shows the test-bed problems, their sizes and the best-known optimum values for the performance analysis. The selected problems are of different sizes and hardness, ranging from 6 x 6 (6 jobs and 6 machines) to 15x15

(15 jobs and 15 machines) so that the performance of the developed approaches could be tested on various datasets.

The heuristic rules were implemented in MATLAB on an Intel (R) Core 2 Duo processor (2.00GHz). Each problem was solved ten times using the developed heuristics. For each run, the objective value (Makespan) was observed and recorded. The data required for the algorithm was in the form of processing times and process plans recorded in spreadsheets. These spreadsheets were used for inputting this data into MATLAB. The traditional heuristics used for the comparison, are taken from literature (Maqsood et al., 2011; Maqsood et al., 2011) and are also simulated in the LAKIN scheduling software.

Table 6. 1: Selected Benchmark problem from literature

Problem Code	**Source**	***Instances*** **Jobs x Machines**	**Best Known Optimum Makespan**
FT 06	Fisher and Thompson (1963)	6 x 6	55
FT 10	Fisher and Thompson (1963)	10 x 10	930
LA 01	Lawrence (1984)	10 x 5	666
LA 06	Lawrence (1984)	15 x 5	926
LA 11	Lawrence (1984)	20 x 5	1222
LA 12	Lawrence (1984)	20 x 5	1039
LA 26	Lawrence (1984)	20 x 10	1218
LA36	Lawrence (1984)	15 x 15	1268

6.2.1 Performance Analysis of the HybH

Tables 6.2 and 6.3 present the computational results of the proposed HybH. These also provide comparative analysis of the HybH with the following well-known traditional heuristics from literature: Shortest Processing Time (SPT), Longest Processing Time (LPT), First In First Out (FIFO), Earliest Due Date (EDD), Critical Ratio (CR), Minimum Slack (MS) and Weighted Shortest Processing Time (WSPT). These comparisons are made using the Relative Deviation (RD) measure or the Mean Relative Error (MRE), also known as the percent GAP (% GAP). The measure % GAP is the deviation of the Makespan value obtained by a particular heuristic from the optimum or the global Makespan. It represents a measure of the quality of the best global Makespan.

The % GAP for a particular heuristic is calculated from the best-known global Lower Bound (LB) or optimum Makespan and the Makespan obtained from particular algorithm using the following relative deviation formula:

$$\% \, GAP = \left\lfloor \frac{Makespan\ found - Makespan\ Optimum}{Makespan\ Optimum} \right\rfloor \times 100 \qquad \text{Equation 6.1}$$

Morshed (2006), reports that in the analyses based on the % GAP, the traditional heuristics achieve results extremely quickly but are of very poor quality (the %GAP from the optimum schedule can be as great as 74%), and in general, the solution quality degrades as the problems' dimensionality increase.

6.2.1.1 Computational experiments and results for HybH

Using the proposed HybH and the traditional heuristics, the Makespan values were obtained for the defined benchmark problem sets of Fisher and Thompson (1963) for

FT and Lawrence (1984) for LA as shown in Table 6.2. For example, the Makespan value obtained by the FIFO heuristic for FT06 (6x6 – six jobs and six operations) case is 65 with a 18.2 % GAP or relative deviation from the optimum. Although, the traditional heuristics are computationally fast, yet none of them achieved the optimum or near-optimum Makespan. Thus the % GAP for the FIFO rule is 18.2% of the optimal value and clearly indicates that it is inefficient for FT06. Looking at the FT06 results, it can be seen that the average Makespan for the seven heuristic rules is 70 with an average GAP of 27%. The best result recorded is the Makespan of 63 with a GAP of 14.5% (with the EDD rule), whilst the worst result is the Makespan of 81 with a GAP of 47.3% (with the CR rule). For the FT10 (10x10) benchmark problem, the heuristic rule's performance was different. The best Makespan achieved for FT10 was 1168 with 25.6% GAP (with the LPT rule) and the worst result achieved was 1338 with 43.87% GAP (with the SPT and WSPT rules). The Proposed HybH, in comparison with the traditional heuristics, has performed much better against all test-bed problems except the FT10.

Table 6. 2: HybH vs. Traditional Heuristics for FT Benchmark Problems

Test Bed	**Fisher and Thompson (1963) – FT**				**Overall Mean GAP%**
Problem	**FT06 (*Opt=55)**		**FT10 (Opt=930)**		
Instances	**6x6**	**Mean GAP%**	**10x10**	**MeanGAP%**	
FIFO	65	18.2%	1184	27.3%	22.7%
LPT	67	21.8%	**1168**	25.6%	23.7%
SPT	73	32.7%	1338	43.9%	38.3%
CR	81	47.3%	1181	27.0%	37.1%
EDD	63	14.5%	1246	34.0%	24.3%
MS	67	21.8%	1168	25.6%	23.7%
WSPT	73	32.7%	1338	43.9%	38.3%
Average	***70***	*27.0%*	***1232***	*32.5%*	*29.7%*
Minimum	***63***	*14.5%*	***1168***	*25.6%*	*20.1%*
Maximum	***81***	*47.3%*	***1338***	*43.9%*	*45.6%*
HybH	**61**	10.9%	1175	26.3%	18.6%

* Optimum Makespan value

For the FT06 problem, the HybH achieved a Makespan of 61 with 10.9% GAP against the best performing traditional heuristic, the EDD with Makespan of 63 and 14.5% GAP). For the FT10 (10x10) results, HybH did not exhibit the best performance achieving a Makespan of 1175 with 26.3% GAP. The LPT and the MS showed the best results (Makespan of 1168 with 25.6% GAP). This is due to the fact that the LPT and the MS heuristics suit the FT10 because of two main reasons. Firstly, Fisher and Thompson (1963) assigned lower number of machines to the earlier operations and higher number of machines for the later ones. Secondly, the first operation has comparatively larger processing times and the HybH sorts the first operation on the basis of ascending index values. Hence, the FT10 is reported in literature as “notoriously” hard because it is different from other benchmark cases. It might be fruitful if the proposed heuristic solves this FT10 problem with the first operation job order sorted on the basis of larger index values.

However, the HybH was close to the minimum value and certainly performed better than the traditional heuristics, as shown in Table 6.3. To further explore the strengths and weaknesses of the proposed heuristic, the HybH was tested against benchmark cases developed by Lawrence (1984). These are of various instances (10x5, 15x5, 20x5, 15x10 and 15x15) as shown in Table 6.1. Referring to the results in Table 6.3, the traditional heuristic FIFO, achieved the optimum in two cases, the LA06 and the LA12, whereas the LPT and the MS achieved the optimum in LA06 and the EDD achieved it in LA12. In comparison, the proposed HybH not only achieved the optimum values for LA06 and LA12, but also comparatively the best Makespan values for all the test-bed cases.

Table 6. 3: HybH vs. Traditional Heuristics for LA Benchmark Problems (Lawrence, 1984)

Test Bed	**Lawrence (1984) – LA**												**Overall Mean GAP%**
Problem	**La01 (Opt=666)**		**La06 (Opt=926)**		**La11 (Opt=1222)**		**La12 (Opt=1039)**		**La26 (Opt=1218)**		**La36 ((Opt=1268)**		
Instances	**10x5**	**GAP%**	**15x5**	**GAP%**	**20x5**	**GAP%**	**20x5**	**GAP%**	**20x10**	**GAP%**	**15x15**	**GAP%**	
FIFO	772	15.9%	**926**	0.0%	1272	4.1%	**1039**	0.0%	1505	23.6%	1516	19.6%	10.5%
LPT	752	12.9%	**926**	0.0%	1300	6.4%	1167	12.3%	1394	14.4%	1480	16.7%	10.5%
SPT	1122	68.5%	1475	59.3%	1802	47.5%	1439	38.5%	1993	63.6%	2250	77.4%	59.1%
CR	979	47.0%	1140	23.1%	1792	46.6%	1401	34.8%	2069	69.9%	2229	75.8%	49.5%
EDD	865	29.9%	1024	10.6%	1272	4.1%	**1039**	0.0%	1430	17.4%	1550	22.2%	14.0%
MS	752	12.9%	**926**	0.0%	1300	6.4%	1167	12.3%	1394	14.4%	1480	16.7%	10.5%
WSPT	1122	68.5%	1475	59.3%	1802	47.5%	1439	38.5%	1993	63.6%	2250	77.4%	59.1%
Average	***909***	*36.5%*	***1127***	*21.7%*	***1506***	*23.2%*	***1242***	*19.5%*	***1683***	*38.1%*	***1822***	*43.7%*	*30.5%*
Minimum	***752***	*12.9%*	***926***	*0.0%*	***1272***	*4.1%*	***1039***	*0.0%*	***1394***	*14.4%*	***1480***	*16.7%*	*8.0%*
Maximum	***1122***	*68.5%*	***1475***	*59.3%*	***1802***	*47.5%*	***1439***	*38.5%*	***2069***	*69.9%*	***2229***	*75.8%*	*59.9%*
HybH	**700**	**5.1%**	**926**	**0.0%**	**1272**	**4.1%**	**1039**	**0.0%**	**1358**	**11.5%**	**1453**	**14.6%**	**5.9%**

Table 6.3 also shows the overall mean % GAP taken across the LA-problems. The proposed HybH has a lesser % GAP value of 6% in comparison with that of the best of the traditional heuristics, which have an overall mean GAP value of 10.5% (the LPT and the MS rules). Hence, the HybH reduced the overall % GAP by 77.9%, which reflects a considerable gain in the process efficiency.

In summary, the proposed HybH performed well consistently across the test-bed FT and LA benchmark problems and can be applied to any size of a problem. However, in the case of traditional heuristics, the performance of each heuristic depended on the type and size of the benchmark problem.

6.2.1.2 Summary of HybH Results

The majority of the processing time based heuristics reported in the literature, which have Makespan optimization as the objective function, is computationally very fast but their relative difference (% GAP) from the optimum is as large as 75%. Furthermore, for the traditional heuristics, no single rule performed well across all the test-bed problems. The proposed HybH overcame the deficiencies in the traditional heuristics for manufacturing scheduling. The novel HybH performed well across all the test-bed benchmark problems and successfully achieved new optimal or near-optimal solutions for the job scheduling problems. It reduced the % GAP in each test-bed problem and the overall mean % GAP by considerable amount.

For the evaluation process, the HybH is applied in combination with Genetic Algorithms (GA). The evaluation process in the GA for JSSPs is a key step that determines the fitness of the objective function and the final solution of the problem.

6.2.2 Performance Analysis of IBH

Tables 6.4 and 6.5 present the computational results of the proposed IBH. These tables also provide comparative analyses of the IBH with the same known traditional heuristics used in Tables 6.2 and 6.3. These comparisons are also made using the Mean Relative Error (MRE) or the % GAP.

6.2.2.1 Computational experiments and results

Using the proposed IBH algorithm and the traditional heuristics, the Makespan values are obtained for the defined benchmark problem sets of Fisher and Thompson (1963) – FT and Lawrence (1984) – LA as shown in Tables 6.4 and 6.5. For example, in Table 6.4, the Makespan value obtained by the FIFO heuristic for the FT06 (6x6 – 6 jobs and 6 operations) case is 65 with an 18.2% GAP or relative deviation from the optimum. Although, the traditional heuristics are computationally fast, yet none of them have achieved the optimum or near-optimum Makespan. Thus, the % GAP for the FIFO rule is 18.2% from the best known optimum Makespan value and clearly indicates that the FIFO rule for FT06 is inefficient. Looking at the FT06 results, it can be seen that the average Makespan for the seven heuristic rules is 70 with a GAP of 27%. The best result recorded a Makespan of 63 and a GAP of 14.5% for the EDD rule, whilst the worst result is a Makespan of 81 and a GAP of 47.3% for CR rule. For the FT10 (10x10) benchmark problem though, the heuristic rule's performance was different. The best Makespan was achieved for FT10 and was 1168 with 25.6% GAP (by the LPT rule) and the worst result achieved was 1338 with 43.87% GAP (by the SPT and WSPT rules).

The proposed IBH in comparison with the traditional heuristics, performed much better. For the FT06 problem, it achieved a Makespan of 59 with 7.3% GAP, whereas the EDD rule, being the best amongst the traditional heuristics, achieved a Makespan of 63 with 14.5% GAP. For the FT10 (10x10), the IBH achieved a better Makespan value of 1136 with 22.1% GAP compared to that achieved by the LPT, where the Makespan is 1168 with 25.6% GAP. Although the values are not equal to the global optimal Makespan values, the IBH algorithm achieved a better Makespan in both the cases with the results being close to the minimum value. The IBH therefore, has performed better than the traditional heuristics as shown in Table 6.4.

Table 6. 4: IBH vs. Traditional Heuristics for FT Benchmark Problems (FT06 (Fisher and Thompson, 1963))

Test Bed	**Fisher and Thompson (1963) - FT**				**Overall Mean GAP%**
Problem	**FT06 (*Opt=55)**		**FT10 (Opt=930)**		
Instances	**6x6**	**Mean GAP%**	**10x10**	**Mean GAP%**	
FIFO	**65**	**18.2%**	1184	27.3%	22.7%
LPT	67	21.8%	**1168**	**25.6%**	23.7%
SPT	73	32.7%	1338	43.9%	38.3%
EDD	63	14.5%	1246	34.0%	24.3%
CR	81	47.3%	1181	27.0%	37.1%
MS	67	21.8%	1168	25.6%	23.7%
WSPT	73	32.7%	1338	43.9%	38.3%
Average	***70***	*27.0%*	***1232***	*32.5%*	*29.7%*
Minimum	***65***	*14.5%*	***1168***	*25.6%*	*20.1%*
Maximum	***81***	*47.3%*	***1338***	*43.9%*	*45.6%*
IBH	**59**	7.3%	**1136**	22.1%	14.7%

* Optimum Makespan value

To further explore the strengths and weaknesses of the proposed heuristic, the IBH was tested against the benchmark cases developed by Lawrence (1984), the same as were used for the HybH, the results of which are shown in Table 6.5. This table shows that from amongst the traditional heuristics, the FIFO achieved the optimum for two cases: the LA06 and the LA12, the LPT and the MS achieved it for the LA06: whereas the EDD achieved the optimum for the LA12 case. The proposed

IBH on the other hand, not only achieved the optimum values for LA06 and LA12, it also achieved better Makespan values for all the test-bed cases.

Table 6.5 also shows the overall mean % GAP taken across the six LA-problems. The proposed IBH has a lesser % GAP value of 5.4% in comparison with that of the best traditional heuristics (LPT and MS rules) that have an overall mean GAP value of 10.5%. Hence, the IBH reduced the overall % GAP by 94.6%, which is again a significant increase in process efficiency.

In summary, the proposed IBH performed well consistently across the test-bed for a range of problem data sets and can be applied to any size of a problem. However, in the cases of traditional heuristics, the performance of each heuristic depended on the type and size of the benchmark problem.

Table 6. 5: IBH vs. Traditional Heuristics for LA Benchmark Problems (Lawrence, 1984)

Test Bed	**Lawrence (1984) – LA**												
Problem	**La01 (Opt=666)**		**La06 (Opt=926)**		**La11 (Opt=1222)**		**La12 (Opt=1039)**		**La26 (Opt=1218)**		**La36 (Opt=1268)**		**Overall Mean GAP%**
Instances	**10x5**	**GAP%**	**15x5**	**GAP%**	**20x5**	**GAP%**	**20x5**	**GAP%**	**20x10**	**GAP%**	**15x15**	**GAP%**	
FIFO	772	15.9%	**926**	0.0%	1272	4.1%	**1039**	0.0%	1505	23.6%	1516	19.6%	10.5%
LPT	752	12.9%	**926**	0.0%	1300	6.4%	1167	12.3%	1394	14.4%	1480	16.7%	10.5%
SPT	1122	68.5%	1475	59.3%	1802	47.5%	1439	38.5%	1993	63.6%	2250	77.4%	59.1%
CR	979	47.0%	1140	23.1%	1792	46.6%	1401	34.8%	2069	69.9%	2229	75.8%	49.5%
EDD	865	29.9%	1024	10.6%	1272	4.1%	**1039**	0.0%	1430	17.4%	1550	22.2%	14.0%
MS	752	12.9%	**926**	0.0%	1300	6.4%	1167	12.3%	1394	14.4%	1480	16.7%	10.5%
WSPT	1122	68.5%	1475	59.3%	1802	47.5%	1439	38.5%	1993	63.6%	2250	77.4%	59.1%
Average	***909***	*36.5%*	***1127***	*21.7%*	***1506***	*23.2%*	***1242***	*19.5%*	***1683***	*38.1%*	***1822***	*43.7%*	*30.5%*
Minimum	***752***	*12.9%*	***926***	*0.0%*	***1272***	*4.1%*	***1039***	*0.0%*	***1394***	*14.4%*	***1480***	*16.7%*	*8.0%*
Maximum	***1122***	*68.5%*	***1475***	*59.3%*	***1802***	*47.5%*	***1439***	*38.5%*	***2069***	*69.9%*	***2229***	*75.8%*	*59.9%*
IBH	**700**	**5.1%**	**926**	**0.0%**	**1272**	**4.1%**	**1039**	**0.0%**	**1324**	**8.7%**	**1453**	**14.6%**	**5.4%**

6.2.2.2 Summary of IBH Results

The IBH heuristics has also overcome the deficiencies in the traditional heuristics for the Job Shop Scheduling Problems. This heuristic has performed well across all the test-bed benchmark problems and successfully achieved new optimal or near-optimal solutions for the JSSPs. The IBH reduced the % GAP in each test-bed problem and the overall mean % GAP by a considerable amount.

For the future, this proposed IBH could be applied in combination with Genetic Algorithms (GA) for the evaluation process. The evaluation process in the GA for a JSSP is a key step that determines the fitness of the objective function. Therefore, future work could focus on the hybridization of IBH with other optimization techniques. The IBH may also be applied to some large sized benchmark and real scheduling problems.

6.3 Analysis of the developed Hybrid Genetic Algorithm (HGA)

The performance of the HGA is analyzed based on the example problem as discussed in detail in Chapter 5. In the following section, the HGA is analyzed for the benchmark problems listed in Table 6.7 in order to gauge the different performance measures of the developed scheduling technique. The results and conclusions are discussed in Section 6.3. The inherent randomness of the selection, crossover, and mutation operators allows a GA to explore the solution space to find new solutions. Introducing random changes to the current best solutions is a trade-off between exploration (find new valleys in search or solution space) of new territory in the solution space and exploitation (digging into a given valley in solution space) of the currently-known best solutions (local optima) (Messenger and Dove, 2012). The mutation operator is normally applied much less often than a crossover, but importantly helps the GA avoid getting entrenched in local optima. The repeated runs with the same data set usually results in the same best solution being discovered. The reason for this may be that many data sets have a similar solution value, and it only takes a minor change in chromosome to produce similar best solution value offspring. However, it is possible that the average Makespan may be different from the best among all trials. This inherent randomness also affects the number of generation and CPU time for in each trial simulation. For example, consider a simple benchmark problem LA01 – a 10 jobs 5 machine problems. This problem was solved ten times without changing GA parameter. The parameters were population size = 100, generation = 50,c Rate = 0.80 %, mutation Rate = 0.070%. The solutions are listed in Table 6.6. The HGA results in different Makespan values, number of generations and CPU time.

Table 6. 6: Data collected for ten different runs of HGA with same parameters

HGA Run No.	Global Makespan	Makespan by HGA	%GAP	CPU Time (Sec)	Number of generations
1	666	666	0.00	166.8	50
2	666	666	0.00	166.1	50
3	666	666	0.00	168.4	50
4	666	666	0.00	166.4	50
5	666	666	0.00	167.1	50
6	666	666	0.00	166.2	50
7	666	671	0.75	167.6	50
8	666	666	0.00	168.0	50
9	666	666	0.00	168.4	50
10	666	666	0.00	168.7	50
Mean =	**666**	**675.2**	**0.75**	**167.38**	**50**

Table 6.6 shows that the HGA has gone to its maximum generation number ‘50’ for searching of the optimal Makespan value of 666 units of time, although it has

achieved this most of the time, it could not achieve optimum results for run number 7. This run, which was adaptively determined, was the highest of all of the runs. It is likely that the system state became caught in a deep local minimum, allowing few if any new states to be explored. Nine optimum results achieved by HGA are due to the reason that the data sets have a similar solution value (LA01 is an easy problem), and it only takes a minor change in chromosome to produce similar best solution value offspring. However, in NP-hard problems, the GA with same GA parameters might result in poor Makespan values for more number of runs than compared to this example or easy problems. The CPU time varied in each run from 166.1 to 168.7 seconds with an average CPU time of 167.38 seconds.

6.3.1 JSSP benchmark problems for HGA

To gauge the strengths and comparative merits of the HGA, it is tested against published benchmark problems. Theses benchmark problems are developed by various researchers (Fisher and Thompson (1963) - FT; Carlier (1978) - CAR; Lawrence (1984) - LA; Adams et al., (1988) - ABZ; Applegate and Cook (1991)-ORB; Storer et al., (1992) - SWV; Yamada and Nakano (1992) – YN and Taillard (1993)). Jain and Meeran (1999) have presented a list of the 242 important benchmark problems of various sizes and hardness level. According to Jain and Meeran (1999) a problem is said to be hard if the total number of operations is greater or equal to 200, number of jobs (N) greater than or equal to 15, machines (M) greater than or equal to 15, and N/M is less than 2.5. The problem types Taillard ‘TA01’, for example, obey this structure and that is why they are hard and are still not yet solved optimally.

Table 6. 7: List of benchmark problems, problem size and best-known optimum Makespan

Prob No.	Source	Prob Code	N / M	Prob. Size N x M	C_{max}		Prob No.	Source	Prob Code	N / M	Prob. Size N x M	C_{max}
1	*Fisher and Thompson (1963)*	FT 06	1	6 × 6	55		**32**	Lawrence (1984)	LA 29	2	20x10	1152
2		FT 10	1	10 × 10	930		**33**		LA 30	2	20x10	1355
3		FT 20	4	20 x 5	1165		**34**		LA 31	3	30x10	1784
4	*Lawrence (1984)*	LA 01	2	10 × 5	666		**35**		LA 32	3	30x10	1850
5		LA 02	2	10 × 5	655		**36**		LA 33	3	30x10	1719
6		LA 03	2	10 × 5	597		**37**		LA 34	3	30x10	1721
7		LA 04	2	10 × 5	590		**38**		LA 35	3	30x10	1888
8		LA 05	2	10 × 5	593		**39**		LA 36	1	15x15	1268
9		LA 06	3	15 × 5	926		**40**		LA 37	1	15x15	1397
10		LA 07	3	15 × 5	890		**41**		LA 38	1	15x15	1196
11		LA 08	3	15 x 5	863		**42**		LA 39	1	15x15	1233
12		LA 09	3	15 x 5	951		**43**		LA 40	1	15x15	1222
13		LA 10	3	15 x 5	958		**44**	*Adams et al., (1988)*	ABZ5	1	10 x 10	1234
14		LA 11	4	20 × 5	1222		**45**		ABZ6	1	10 x 10	943
15		LA 12	4	20 x 5	1039		**46**		ABZ7	1.33	20 x 15	656
16		LA 13	4	20 x 5	1150		**47**		ABZ8	1.33	20 x 15	665
17		LA 14	4	20 x 5	1292		**48**		ABZ9	1.33	20 x 15	679
18		LA 15	4	20 x 5	1207		**49**	*Applegate and Cook (1991)*	ORBI	1	10 x 10	1059
19		LA 16	1	10 x 10	945		**50**		ORB2	1	10 x 10	888
20		LA 17	1	10 x 10	748		**51**		ORB3	1	10 x 10	1050
21		LA 18	1	10 x 10	848		**52**		ORB4	1	10 x 10	1005
22		LA 19	1	10 x 10	842		**53**		ORB5	1	10 x 10	887
23		LA 20	1	10 x 10	902		**54**	*Taillard (1993)*	TA01	1	15x15	1005
24		LA 21	1.5	15 x 10	1046		**55**		TA02	1	15x15	953
25		LA 22	1.5	15 x 10	927		**56**		TA06	1	15x15	1134
26		LA 23	1.5	15 x 10	1032		**57**		TA11	1.33	20x15	1254
27		LA 24	1.5	15 x 10	935		**58**		TA12	1.33	20x15	1267
28		LA 25	1.5	15 x 10	977		**59**		TA13	1.33	20x15	1243
29		LA 26	2	20x10	1218		**60**		TA16	1.33	20x15	1211
30		LA 27	2	20x10	1235		**61**		TA31	2	30x15	1764
31		LA 28	2	20x10	1216		**62**		TA37	2	30x15	1771

Table 6.7 shows 62 selected benchmark JSSPs, author (s), problem code, its size (number of jobs and the number of machines) and the three important hardness parameters (N/M, number of machines and the number of operations for the

problems). The processing times and machines' order for each problem is shown in Appendix A.

6.3.2 Computational experiments and results for JSSP

Table 6.8 presents the computational results of the proposed HGA for the benchmark JSSPs. The HGA was implemented in MATLAB on an Intel (R) Core 2 Duo processor (2.00 GHz). Each problem was solved twenty times and for each run time, the Makespan value and the number of generations were computed and recorded for the best evolved schedule under the following GA parameters:

- Population Size = 30 to 200
- Generation = 10 to 200
- Crossover Rate = 70 to 80 %
- Mutation Rate = 1 to 10%
- Iterations = 20 to 100
- Job number = 6 to 30
- Machine Number = 5 to 20

The algorithm stops when either the best known optimal solution is proven or the number of generations reaches the predefined maximum number. For each generation, the objective value (Makespan) was observed and recorded. The input data required for the algorithm was recorded in spreadsheets, in the form of processing times and process plans. These spreadsheets were then used for inputting the data into MATLAB. Table 6.8 provides the comparisons between the Makespan found by HGA and the global known Makespan values using the percent GAP (% GAP).

Table 6.8 shows the experimental results of the HGA for the benchmark JSSPs. The results obtained are tabulated for the calculated Makespan results from the HGA and their corresponding deviations from the known optimum values using the tabulated %GAP.

Columns 1 through to 5 show the benchmark JSSP number, problem source, problem code, problem size and the best-known optimum Makespan respectively. The 6^{th}, 7^{th} and 8^{th} column show the initial solution obtained from the HybH and the relative % GAP from the optimum respectively. The 7^{th} column also shows the optimum that has been initially achieved using the HybH. Hence, the best solution with zero percent GAP for some problems can be seen in the 7^{th} column. Columns 9 and 10 show the optimum or near-optimum Makespan achieved by the HGA, their relative % GAP, number of generations at which it achieved the best Makespan. Columns 11, 12 and 13 show average Makespan, each average Makespan's %GAP from optimum result and the average CPU time (seconds) of 20 runs respectively.

The HGA has been able to achieve the optimal solutions for a considerable number of benchmark problems in reasonable computational time. In general, most of the squared problems are found to be harder than non-squared hard problems. In the following sub-sections the chapter provides an analysis of the performance of the HGA in graphs showing Makespan and best known Makespan or LB, Makespan %GAP from optimal, effect of number of generations on Makespan, and average CPU time for each problem. The comparison of HGA with some of the published models from literature is given in Section 6.4.3.

Table 6. 8: Experimental Results HGA for JSSP

					HybH			HGA Best			HGA Average		
Prob. No	Source	Prob Code	Prob. Size N x M	C_{max}	Best Makespan	%age GAP	GenNo.	Best Makespan	%age GAP	Gen No.	Average Makespan	% age GAP	Average CPU Time (Sec)
1	*Fisher and Thompson (1963)*	FT 06	6 × 6	55	61	10.91	0	55	0	1	55	0	3.86
2		FT 10	10 × 10	930	1175	26.34	0	930	0	80	930	0	28.3
3		FT 20	20 x 5	1165	1570	34.76	0	1165	0	28	1165	0	13.8
4	*Lawrence (1984)*	LA 01	10 × 5	666	700	5.11	0	666	0	9	666	0	15.84
5		LA 02	10 × 5	655	808	23.36	0	655	0	14	655	0	17.4
6		LA 03	10 × 5	597	726	21.61	0	597	0	18	597	0	19.2
7		LA 04	10 × 5	590	660	11.86	0	590	0	7	590	0	22.9
8		LA 05	10 × 5	593	593	0	0	593	0	0	593	0	0.16
9		LA 06	15 × 5	926	926	0	0	926	0	0	926	0	0.13
10		LA 07	15 × 5	890	976	9.66	0	890	0	17	890	0	12.2
11		LA 08	15 x 5	863	925	7.18	0	863	0	9	863	0	11.1
12		LA 09	15 x 5	951	951	0	0	951	0	0	951	0	0.26
13		LA 10	15 x 5	958	958	0	0	958	0	0	958	0	0.19
14		LA 11	20 × 5	1222	1272	4.09	0	1222	0	1	1222	0	6.39
15		LA 12	20 x 5	1039	1039	0	0	1039	0	0	1039	0	0.5
16		LA 13	20 x 5	1150	1153	0.26	0	1150	0	1	1150	0	0.22
17		LA 14	20 x 5	1292	1292	0	0	1292	0	0	1292	0	8.11
18		LA 15	20 x 5	1207	1466	21.46	0	1207	0	29	1207	0	168.6
19		LA 16	10 x 10	945	1093	15.66	0	945	0	63	945	0	188.4
20		LA 17	10 x 10	748	907	21.26	0	748	0	67	748	0	129.1

21		LA 18	10 x 10	848	988	16.51	0	848	0	86	848	0	126.7
22		LA 19	10 x 10	842	968	14.96	0	842	0	92	842	0	133.8
23		LA 20	10 x 10	902	902	0	0	902	0	0	902	0	0.14
24		LA 21	15 x 10	1046	1265	20.94	0	1046	0	107	1046	0	221.2
25		LA 22	15 x 10	927	1171	26.32	0	927	0	92	927	0	148
26		LA 23	15 x 10	1032	1130	9.5	0	1032	0	32	1032	0	178.8
27		LA 24	15 x 10	935	1138	21.71	0	935	0	79	935	0	219.34
28		LA 25	15 x 10	977	1215	24.36	0	977	0	86	977	0	188.22
29		LA 26	20x10	1218	1358	11.49	0	1218	0	94	1218	0	211.78
30		LA 27	20x10	1235	1538	24.53	0	1235	0	109	1235	0	306.01
31		LA 28	20x10	1216	1471	20.97	0	1216	0	74	1216	0	604.95
32		LA 29	20x10	1152	1448	25.69	0	1152	0	95	1152	0	648.41
33		LA 30	20x10	1355	1550	14.39	0	1355	0	82	1355	0	719.08
34		LA 31	30x10	1784	1897	6.33	0	1784	0	75	1784	0	1433.21
35		LA 32	30x10	1850	1950	5.41	0	1850	0	81	1850	0	1467.25
36		LA 33	30x10	1719	1830	6.46	0	1719	0	96	1719	0	1481.39
37		LA 34	30x10	1721	1876	9.01	0	1721	0	72	1721	0	1495.52
38		LA 35	30x10	1888	2008	6.36	0	1888	0	83	1888	0	1505.23
39		LA 36	15x15	1268	1453	14.59	0	1268	0	79	1268	0	1514.37
40		LA 37	15x15	1397	1588	13.67	0	1397	0	84	1397	0	1533.21
41		LA 38	15x15	1196	1466	22.58	0	1196	0	104	1196	0	1482.39
42		LA 39	15x15	1233	1491	20.92	0	1233	0	92	1233	0	1608.38
43		LA 40	15x15	1222	1441	17.92	0	1222	0	79	1222	0	1642.9
44	*Adams et al., (1988)*	ABZ5	10 x 10	1234	1351	9.48	0	1234	0	94	1234	0	1782.45
45		ABZ6	10 x 10	943	1014	7.53	0	943	0	86	943	0	1813.7
46		ABZ7	20 x 15	656	778	18.6	0	656	0	127	657.2	0.11	1054.18

47		ABZ8	20 x 15	665	790	18.8	0	665	0	116	665	0	1755.24
48		ABZ9	20 x 15	679	875	28.87	0	679	0	102	679	0	1410.72
49	*Applegate and Cook (1991)*	ORBI	10 x 10	1059	1251	18.13	0	1059	0	75	1059	0	1496.58
50		ORB2	10 x 10	888	983	10.7	0	888	0	81	888	0	1529.56
51		ORB3	10 x 10	1050	1365	30	0	1050	0	106	1050	0	1664.85
52		ORB4	10 x 10	1005	1225	21.89	0	1005	0	76	1005	0	1493.7
53		ORB5	10 x 10	887	1013	14.21	0	887	0	87	887	0	1548.41
54	*Taillard (1993)*	TA01	15x15	1231	1451	17.87	0	1231	0	169	1231	0	6552.13
55		TA02	15x15	1244	1442	15.92	0	1244	0	182	1248	0.322	6814.92
56		TA06	15x15	1240	1462	17.9	0	1281	3.31	200 (171)	1410	12.06	8317.28
57		TA11	20x15	1364	1688	23.75	0	1364	0	149	1591	14.27	6183.51
58		TA12	20x15	1367	1657	21.21	0	1367	0	168	1403	2.57	5706.26
59		TA13	20x15	1350	1798	33.19	0	1350	0	181	1512	10.71	6958.08
60		TA16	20x15	1368	1678	22.66	0	1368	0	192	1537	11.00	6452.49
61		TA31	30x15	1766	2213	25.31	0	1837	4.02	200 (139)	2127	16.97	9846.98
62		TA37	30x15	1784	2233	25.17	0	1871	4.88	200 (162)	2018	11.60	8041.47
					Overall Mean GAP	15.31		Overall Mean GAP	0.2				

6.3.2.1 Average CPU time vs. Problem Size

Table 6.8 shows that the average CPU time for easy problems such as FT06, LA01 to LA15 is as high as 29 seconds. For hard problems. For FT10 a well known hard problem it took 28.3 seconds to converge. The problems 'LA36-LA40' , 'ABZ5-ABZ10' and 'ORB1-ORB5' are comparatively difficult problems, the CPU time shows an increase in comparison to easy problems. While the well known hardest TA problems show a considerable increase in the mean CPU solution time. Again, it is likely that the algorithms trapped in the local search and the system is allowed for extra iterations and the system spent excessive amount of time. Hence, the CPU time and the number of function evaluations performed during the execution of the runs was very high as compared to the rest of problems which successfully converegєd.

The trend in CPU solution time shows that it does not depend on N/M ratio (hardness ratio) but it rather depends on the size and nature of the problem, population size and number of generations. The termination criteria, which is algorithm, will stop if the best known optimum value or maximum number of generation arrives. In case, the problem is solved optimally before the maximum number of generations, the CPU solution time is saved. As shown in Table 6.8, the optimal solutions for the benchmark Problems LA01 to LA20 have been found in a maximum of two generations because of the less hardness of the problems.

Similarly, explanations to the question that why different groups of problems have a larger CPU solution time with same hardness level and same number of generations and population is that there is an increase in the number of total operations from problem to problem and the way the problems are created by the author.

6.3.2.2 Makespan achieved by HGA vs. known optimal values

Figure 6.1 shows the experimental results of HGA for 62 benchmark job shop scheduling problems. The problems are along x-axis and Makespan values are along y-axis. The first line (dark blue) shows the global known optimum value, while 2nd (red) and 3rd (light blue) lines show HGA results at zero generation and final result respectively. Table 6.8 and Figure 6.1 shows that the HGA achieved optimum results (%GAP = 0) for 59 out of 62 problems with an overall Mean GAP of 0.20% of all 62 problems. The HGA achieved the optimal results for seven (LA5, LA6, LA9, LA10, LA12, La14, LA20) benchmark problems in zero generation. The overall mean GAP for solutions at zero generation is 15.31% which indicates the quality of initial solution produced by HybH in HGA loop. For eight problems (LA21, LA27, LA38, LA40, ABZ7, ABZ8, ABZ9 and ORB3), the HGA found optimum results, however, it exceeded the generation number 100.

The HGA behavior on Taillard’s hard problems is also encouraging. Nine problems generated from Taillard’s algorithm (Taillard, 1993) are tested on HGA. HGA successfully achieved the best known optimal results (reported by Jain and Meeran, 1999). For three problems (TA06, TA31 and TA37), the HGA was unable to achieve the optimal values of the problems. However, the results of these three problems are near optimal solutions with a %GAP of 3.31 (TA06), 4.02 (TA31), and 4.88 (TA37).

Hence, the HGA results are encouraging and it has the ability to produce optimal results even for a larger set of NP hard scheduling problems. In Section 6.5, results are presented of the HGA applied to real world scheduling problems (36 case studies) from literature.

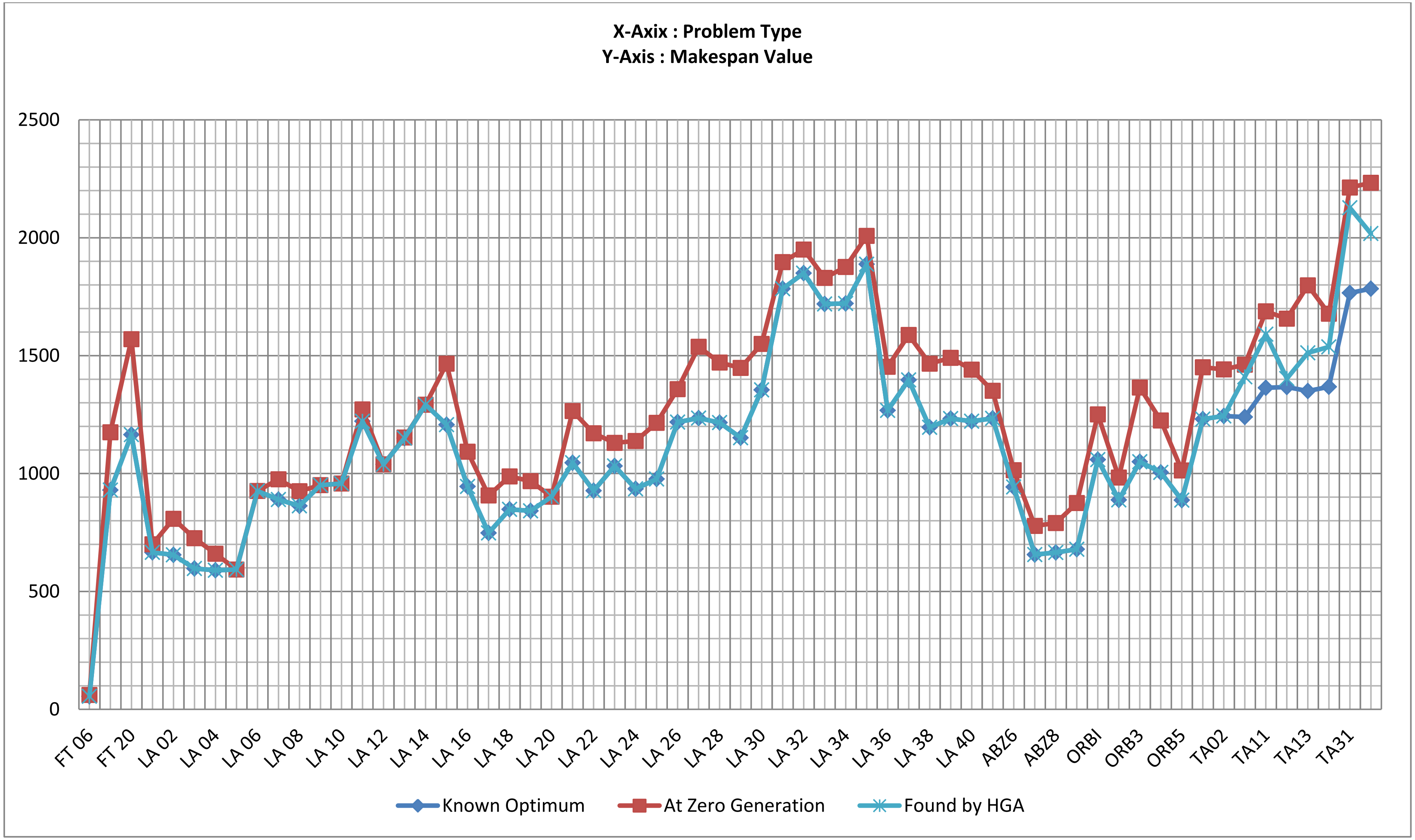


Figure 6. 1: Makespan values achieved by HGA and Optimum values vs. problem

6.3.2.3 Percent GAP between the calculated Makespan by HGA and optimum

Figure 6.2 shows the % GAP between the optimum Makespan results: in zero generation and optimum or near optimum achieved by HGA. The green line represents the % GAP of solution for initial solution obtained by HybH, while the orange and light blue line represents the % GAP of solution achieved by HGA and overall mean %GAP repectively. Figure 6.2 shows that the initial solution depends on the size and hardness of problems. For easy problems such as LA5, LA6, LA9, LA10, LA12, La14, LA20 HybH achieved the optimum makespan. For comparitivly hard problem than easy problems such as FT06, LA24, LA26, LA31, LA32, LA33, LA34, LA35, LA36, ABZ5, ABZ6 and ORB3 HybH achieved near optimal results with less than % GAP of 10. For hard problem the HybH %GAP reaches a value of 33.19. Which is encouraging and helped HGA in achieving optimum or near optimum Makespan results. The HGA has achieved optimal for 59 problems and near optimal results for remaining 3 problems. The HGA near optimal results are for the problems TA06, TA31 and TA37 with a % GAP of 3.31, 4.02 and 4.88.

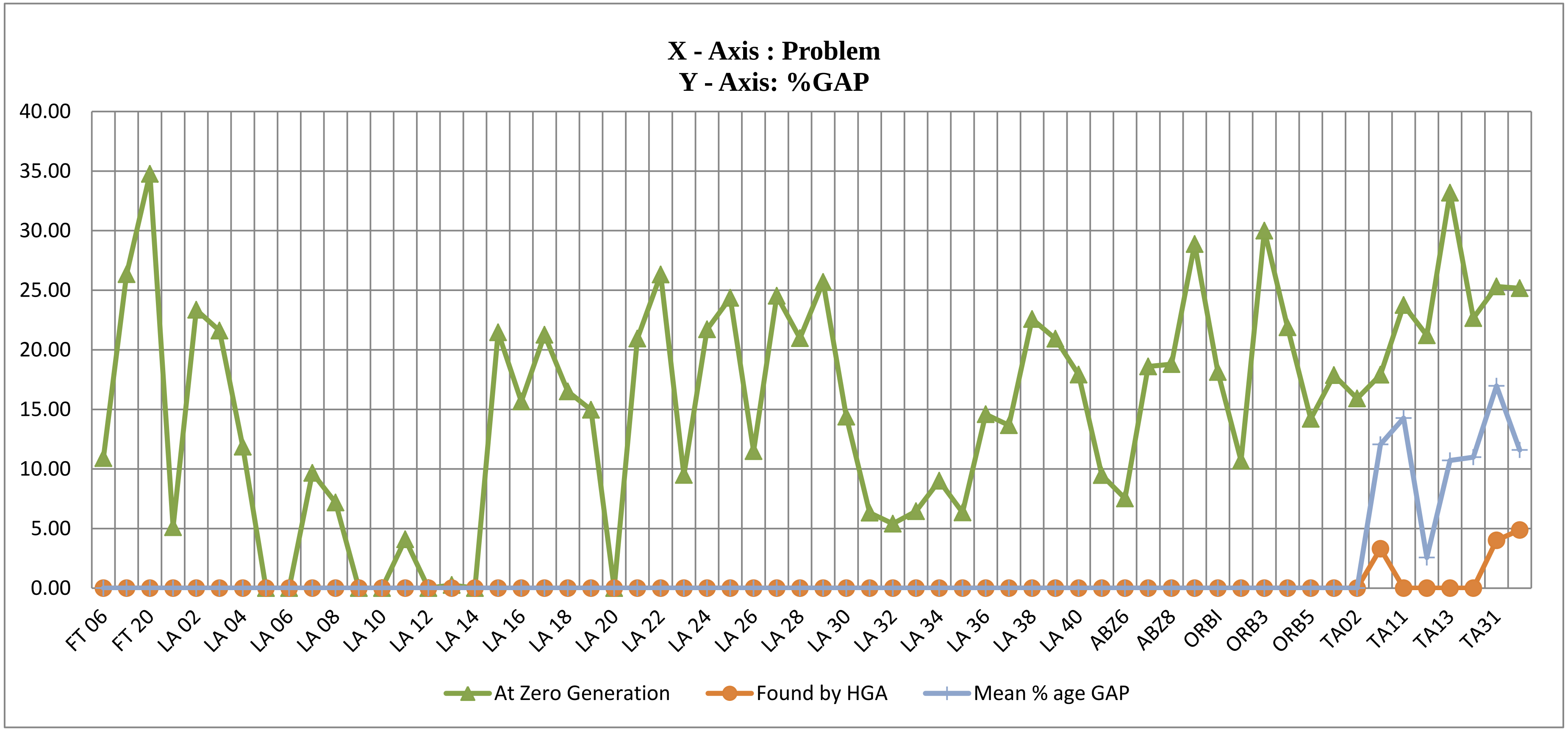


Figure 6. 2: % GAP vs. Problem

6.3.2.4 Effect of generation on Makespan

Figure 6.3 shows that the behavior of HGA is following a trend such as larger and harder problem size resulting in the high number of generations although few exceptions are there such as LA23 and LA28. The dark blue line represents the Makespan achieved by HGA and light blue line represents the generation number. The figure also shows that most of the squared problems result comparatively in a higher number of generations than non squared problems due to the fact that in square problems the resources are limited and the jobs take longer time in waiting or in queue. Seven problems (LA05, LA06, LA09, LA10, LA12, LA14 and LA20) converged in the initial solution search at the zero generation which are 7.8% of the problems. The FT06, LA11, and LA13 problems converged at first generation. A total 65.21% (46 problems) problems converged in 100 generations in which 4.16% (12 easy problems) problems converged in just 32 generations. For 4.08% (13 hard problems) problems including eight Taillzard's problems covered in more than 100 generations. For three problems TA06, TA31 and TA37, that found near optimal solutions at the end of 200 generations. Hence, the problems convergence tendency increases if the algorithm is allowed to iterate for more generations.

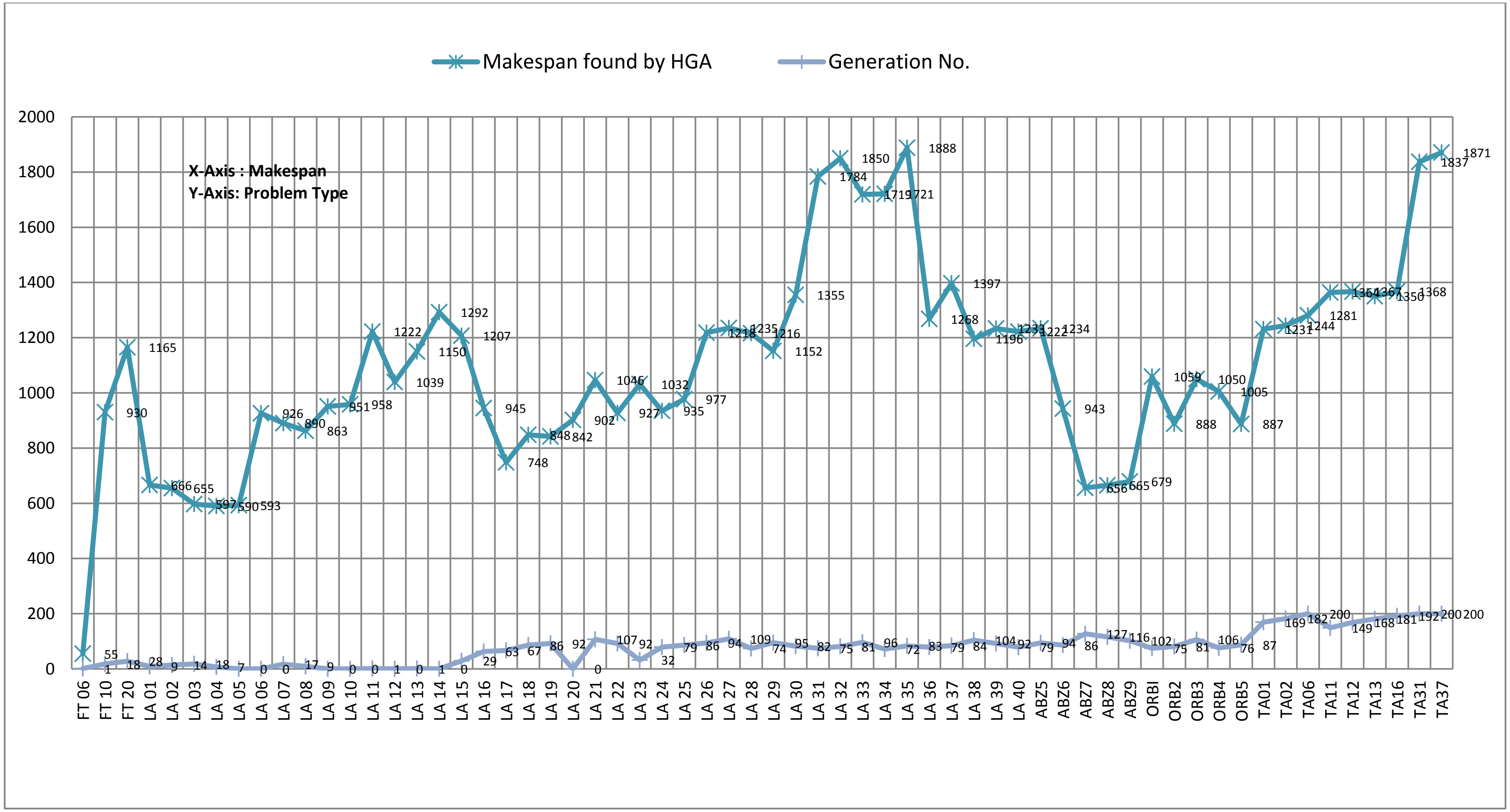


Figure 6. 3: No. of generation vs problem

6.3.2.5 Summary of HGA Results for JSSP

In this section, the results of the developed HGA for JSSP were discussed. The operational performance of the HGA was tested in detail with various sizes of benchmark problems. The GA in combination with a hybrid heuristic for evaluation of the schedule performed well across the data of benchmark problems of various sizes. The HGA results (%GAP between the calculated Makespan values and Global Makespan Values) were as low as zero across the test bed for 59 out of 62 problems. In Section 6.5, HGA was further tested against case studies from literature. The results from case studies were also discussed in the same section.

6.4.1 Benchmark FSSPs for HGA

The flow shop type of a flow pattern is a typical one of mass production, i.e., high rate of production and lower manufacturing cost as explained in Chapter 2, Section 2.4. The experimental results of the flow shop benchmark problems applied to the HGA are discussed in this section. Here, unlike the job shop, specialized machines are used. Each flow line is organized according to the processing requirements of a product. In the simplest case, each job consists of the same set of activities to be performed sequentially on the same set of machines within multiple sets of machines as shown in Figure 6.4.

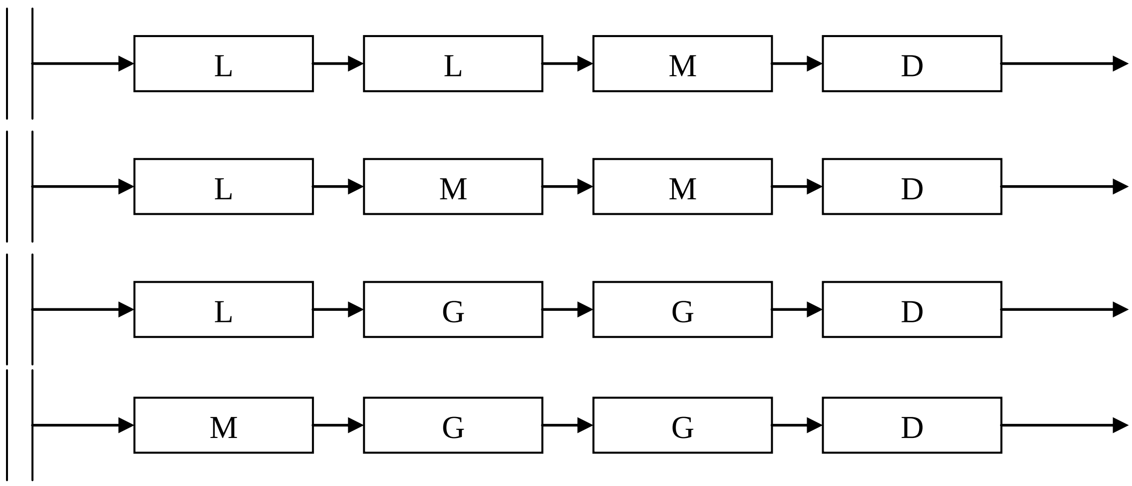


Figure 6. 4: Flow Shop Environment

The same GA parameters are used in the computational tests for the flow shop benchmark problems as proposed by Carlier (1978). The HGA was also test on Taillard's (Taillard, 1993) problems. These benchmark problems are tabulated in Table 6.9 and in Appendix B their respective dimensions are given. The range of processing times for each of these problems is higher than that of the benchmark JSSPs. The optimum results and deviation from the optimum for different flow shop problems are shown in Table 6.9.

Table 6. 9: Benchmark Flow Shop Scheduling Problems

Problem No	Source	Problem Code	Problem Size N x M	C_{max}
1	*Carlier (1978)*	CAR1	11 x 5	7038
2		CAR2	13 x 4	7166
3		CAR3	12 x 5	7312
4		CAR4	14 x 4	8003
5		CAR5	12 x 5	7702
6		CAR6	8 x 9	8313
7		CAR7	7 x 7	6558
8		CAR8	8 x 8	8264
9	*Taillard (1993)*	TA01	20x5	1278
10		TA02	20x5	1359
11		TA03	20x5	1081

6.4.2 Computational experiments and results for FSSPs

Table 6.10 shows the experimental results of the HGA for the benchmark FSSPs (See Appendix B for the processing times and process plan for each of these problems). The results are tabulated for the calculated Makespans obtained from the HGA and their corresponding deviations from the known optimum values and tabulated as %GAPs.

Columns 1 through 5 show the benchmark FSSP number, problem source, problem code, problem size and the best-known optimum Makespan respectively. The 6th, 7th and 8th columns show the results from the initial solution and columns 9,10 and 11 show the results for the FSSPs using HGA. Columns 12, 13 and 14 show the mean Makespan value, mean generations and mean CPU time for the problems. Each problem was solved twenty times and for each run time, the Makespan value and the number of generations were computed and recorded for the best evolved schedule under the following GA parameters:

- Population Size = 30 to 200
- Generation = 10 to 200
- Crossover Rate = 70 to 80 %
- Mutation Rate = 1 to 10%
- Iterations = 20 to 100
- Job number = 6 to 30
- Machine Number = 5 to 20

Table 6.10 shows an overall %GAP of 10.40 for the initial solution using HybH. The smallest deviation in the initial solution is for CAR7 and is 6.92%. The HGA achieved optimum results for all of the problems. However, the average Makespan is with an overall %GAP of 0.1621. It is likely that the system state became trapped in a deep local minimum, allowing few more generation may help in exploring new states. For CAR1 to CAR8, the average generation numbers on which the HGA achieved optimum or near-optimum results were 17, 60, 21, 9, 52, 29, 17, 21. In CAR2 and CAR5, although the HGA did not achieve the mean optimum with 0.0 %GAP but the mean near-optimum results with small relative deviations of 0.32% and 0.91% were recorded respectively.

The average CPU time ranged from 3797.12-5288.2 seconds for problems whose mean optimum is equal to the best known optimum results. In the case of problem CAR02 and CAR05 it is likely that the system spent excessive amount of time in local random search. Hence, the CPU time and the number of function evaluations performed during the execution of the runs was quite comparable with those of the successful cases such as CAR01, CAR03.

The HGA has also achieved optimum Makespan results for all three TA problems. However, the computation time is higher due to the size and hardness of the problems. The average CPU time ranges from 4162.12 to 7209.22. The Overall % GAP for initial solution of TA problems achieved by HybH is almost similar to the CAR problems, which further supports the good HybH performance. Although, the average Makespan is comparatively higher for TA01, TA02 and TA03 FSSPs, but still the %GAPs are less than 5.

Table 6. 10: Experimental Results of Flow Shop Scheduling Problem (FSSP)

Benchmark FSSPs					Makespan Achieved by HGA									
					Makespan at Generation No. = 0 (zero)			Makespan at Respective Generation No.			Average Makespan with % GAP			
Prob. No	**Source**	**Prob Code**	**Prob. Size N x M**	**C_{max}**	**C_{max}**	**% GAP**	**Gen. No.**	**C_{max}**	**% GAP**	**Gen. No.**	**C_{max}**	**Average % GAP**	**Average CPU Time**	Results for CAR Problems
1	*Carlier (1978)*	CAR1	11 x 5	7038	7718	9.66	0	7038	0	17	7038	0	3797.12	
2		CAR2	13 x 4	7166	7741	8.02	0	7166	0	60	7176	0.32	13532.4	
3		CAR3	12 x 5	7312	8237	12.65	0	7312	0	21	7312	0	4628.61	
4		CAR4	14 x 4	8003	8679	8.45	0	8003	0	9	8003	0	4097.52	
5		CAR5	10 x 6	7702	8416	9.27	0	7702	0	52	7767	0.98	11915.8	
6		CAR6	8 x 9	8313	9811	18.02	0	8313	0	29	8313	0	8288.2	
7		CAR7	7 x 7	6558	7012	6.92	0	6558	0	17	6558	0	3935.33	
8		CAR8	8 x 8	8264	9109	10.23	0	8264	0	21	8264	0	4663.89	
Overall % Mean GAP						**10.4**	Overall % Mean GAP		**0**	Overall Mean% GAP		**0.1621**		
Prob. No	**Source**	**Prob Code**	**Prob. Size N x M**	**C_{max}**	**C_{max}**	**% GAP**	**Gen. No.**	**C_{max}**	**%GAP**	**Gen. No.**	**C_{max}**	**Average % GAP**	**Average CPU Time**	Result for TA Problems
9	*Taillard (1993)*	TA01	20x5	1278	1381	8.059	0	1278	0	86	1297	1.487	4162.12	
10		TA02	20x5	1359	1426	4.930	0	1359	0	117	1402	3.164	5323.49	
11		TA03	20x5	1081	1293	19.611	0	1081	0	172	1132	4.718	7209.22	
Overall % Mean GAP						**10.867**	Overall % Mean GAP		**0**	Overall Mean% GAP		**3.123**		

6.4.3 Result Comparison with Other Developed Models from Literature

Table 6.11 shows the results comparison of the developed HGA with some developed models from the literature (Morshed, 2006). The comparison is made on the basis of mean Makespan and overall mean % GAP for each of the benchmark FSSPs. Table 6.11 shows that HGA has performed overall better than Nowicki and Smutnicki (1996) – NS96 and Demirkol et al. (1997)- DMU97. The overall mean %GAP of HGA is 0.1621 while NS96 and DMU97 are having overall mean % GAP of 1.0148 and 5.6529 respectively. Jain and Meeran (2002) – JM02 has achieved optimum results for 7 out of 8 problems with an overall mean GAP in 0.0343. However, Morshed (2006), has achieved the result for all 8 problems with 0.00 % GAP. He has not provided CPU times for his models and therefore the HGA cannot be compared on the basis of CPU times with his work. However, the HGA convergence is a lesser number of generations than Morshed's model and it can be concluded that the HGA is faster than Morshid's model. In conclusion, the HGA has outperformed some well known models available in the literature.

Table 6. 11: HGA result comparison of general FSSPs with other models

Probs	CAR1 (Opt=7038)		CAR2 (Opt=7166)		CAR3 (Opt=7312)		CAR4 (Opt=8003)		CAR5 (Opt=7702)		CAR6 (Opt=8313)		CAR7 (Opt=6558)		CAR8 (Opt=8264)		Overall Mean GAP
	Result	% GAP	Result	% GAP	Result	% GAP	Result	% GAP	Result	% GAP	Result	% GAP	Result	% GAP	Result	% GAP	
Nowicki and Smutnicki (1996)	7038	***0***	7376	2.93	7531	3	8003	***0***	7720	0.23	8313	***0***	6573	0.23	8407	1.73	***1.0148***
Demirkol et al. (1997)	7220	2.59	7741	8.02	8237	12.7	8423	5.25	8380	8.8	8739	5.12	6617	0.9	8420	1.89	***5.6529***
Jain and Meeran (2000)	7038	***0***	7166	***0***	7312	***0***	8003	***0***	7702	***0***	8313	***0***	6576	0.27	8264	***0***	***0.0343***
Morshed (2006)	7038	***0***	7166	***0***	7312	***0***	8003	***0***	7702	***0***	8313	***0***	6576	***0***	8264	***0***	***0.00***
HGA	7038	***0***	7176	0.14	7312	***0***	8003	***0***	7767	0.84	8313	***0***	6558	***0***	8264	***0***	***0.1621***

6.4.4 Summary of HGA Results for FSSPs

In this section, the results of the developed HGA for FSSPs are discussed. The operational performance of the HGA was tested in detail with various sizes of benchmark FSSPs. The HGA performed well across the benchmark problems and has achieved optimal values for six out of the eight benchmark problems and near-optimal for the rest of the problems. The initial solution is very good as compared to the other models developed in the literature, which is very encouraging.

The HGA was also compared with the other developed models from literature and performed better in more than two-third of the models.

6.5 Application of HGA in Industrial Case Studies

Three industrial case studies were taken from the literature for the purpose of validating HGA. Two of these cases were taken from Morshed (2006) and one from Altaf et al., (2010). A brief introduction of the case studies is presented in the following sections.

6.5.1 Industrial Case Study 1: Shauful Alam Steel Mills (SASM)

The Shauful Alam Steel Mills (SASM) is a Bangladesh based local factory established in 1986. The SASM produces various products such as torsion steel bars, plain round bars, equal angles, channels, low carbon steel wire, pipes and spare parts. For the production of these products, numbers of related operations are performed on jobs on various types of machines. These include cutting, shaper, grinding, milling, turning, drilling/boring, polishing, painting, drying and other special turning machines.

Table 6. 12: SASM Scheduling Problems [Morshed (2006)]

Problem Name	Description	Size
SASM JSSP1	8 jobs and 6 machines Job shop problem	8 x 6
SASM JSSP2	6 jobs and 6 machines Job shop problem	6 x 6
SASM JSSP3	6 jobs and 6 machines Job shop problem	6 x 6
SASM FSSP1	7 jobs and 6 machines flow shop problem	7 x 6
SASM FSSP2	6 jobs and 6 machines flow shop problem	6 x 6

Jobs are processed through different routes on these machines and hence result in a variety of JSSPs and FSSPs on the shop floor. Some of these problems are shown in Table 6.12.

Tables 6.13 and 6.14, show the processing times and process plans for three JSSPs and two FSSPs (shown in Table 6.12) respectively. The 1st Column shows the number of jobs, whereas in columns 2 through 13, all the six operations are listed along with the corresponding processing times and the machines on which each job is to be executed. The SASM JSSP1 has eight jobs and six operations while the SASM JSSP2 and JSSP3 have six jobs and six operations. The flow shop scheduling problems FSSP1 and FSSP2 both have seven jobs and six operations.

Table 6. 13: Machine Sequences and Processing Times for SASM's JSSPs [Case study 1]

SASM JSSP1												
	O1		O2		O3		O4		O5		O6	
Jobs	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
J1	1	10	2	32	3	21	4	40	5	53	6	23
J2	2	35	1	0	3	29	4	51	5	48	6	20
J3	4	54	1	0	5	53	3	0	6	21	2	0
J4	3	38	2	23	4	65	5	61	6	21	1	0
J5	1	15	2	30	3	31	4	54	5	53	6	18
J6	1	16	2	25	3	17	4	68	5	64	6	0
J7	4	54	1	0	5	53	3	0	6	21	2	0
J8	2	28	1	16	3	22	4	55	5	59	6	20

SASM JSSP2												
	O1		O2		O3		O4		O5		O6	
Jobs	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
J1	3	25	1	12	2	18	4	56	6	23	5	65
J2	2	20	3	32	5	67	6	24	1	13	4	46
J3	3	21	4	43	6	19	1	10	2	23	5	55
J4	2	24	1	15	3	40	4	61	5	68	6	21
J5	3	35	2	27	5	71	6	19	1	12	4	55
J6	2	30	4	65	6	18	1	15	5	66	3	35

SASM JSSP3												
	O1		O2		O3		O4		O5		O6	
Jobs	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
J1	3	21	1	10	2	32	4	40	6	23	5	53
J2	2	15	3	8	5	61	6	35	1	14	4	45
J3	3	21	4	55	6	22	1	10	2	30	5	58
J4	2	34	1	9	3	19	4	50	5	52	6	20
J5	3	23	2	35	5	63	6	25	1	11	4	48
J6	2	38	4	41	6	18	1	10	5	65	3	43

Table 6. 14: Processing time for SASM's FSSPs [Case Study 1]

FSSP1												
Job No	O1		O2		O3		O4		O5		O6	
	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
J1	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23
J2	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23
J3	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23
J4	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23
J5	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23
J6	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23
J7	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23

FSSP2												
Job No	O1		O2		O3		O4		O5		O6	
	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
J1	M1	10	M2	32	M3	21	M4	40	M5	53	M6	23
J2	M1	14	M2	15	M3	8	M4	45	M5	61	M6	35
J3	M1	10	M2	30	M3	21	M4	55	M5	58	M6	22
J4	M1	9	M2	34	M3	19	M4	50	M5	52	M6	20
J5	M1	11	M2	35	M3	23	M4	48	M5	63	M6	25
J6	M1	10	M2	38	M3	43	M4	41	M5	65	M6	18

6.5.2 Industrial Case Study 2: Pilkington PLC

A part manufacturing and supplier company, Pilkington PLC supply glazing products for building and automotive industries. The company supplies two main types of products: (a) building products and (b) automotive glass replacement (AGR) products. Scheduling problems at the King Norton's site of Pilkington PLC are presented in Table 6.15.

Table 6. 15: Various Scheduling Problems at Pilkington PLC [Morshed (2006]

Problem Name	Descriptions	Line	Size *N x M*
L319 heated	Land Rover Dsicovery-L319 Heated windscreen	B	10 x 2
L319 Non heated	Land Rover Dsicovery-L319 Non-heated windscreen	B	10 x3
CB40	Land Rover free Lander-CB40 Windscreen	A	10 x 3
Honda CRV	Honda Civic-Honda CRV Windscreen	B	10 x 4
R3 (AGR) Rover 200	Rover-R3 (AGR) Rover 200 Bagged windscreen	A	10 x 2
HHR (AGR) Rover 400	Rover-HHR (AGR) Rover 400 Bagged windscreen	A	10 x 5

These are flow shop problems during the production of various types of windscreens. Therefore, parts are processed in a typical FSSP fashion, i.e., in a sequence of the machines (M_1, M_2, M_3,…), depending on the number of operations on a job. The operation O_1 is performed on the machine M_1, O_2 on M_2 and so on. The processing times of operations for each type of problem are given in Table 6.16.

Table 6.16 shows only 10 parts for each problem, though each type of problem has various numbers of parts, ranging from 10 to 100, and therefore, the processing times may extend accordingly.

Table 6. 16: Processing Times for Various Scheduling Problems at Pilkington PLC

Job No	L319 heated		L319 Non heated		CB40			Honda CRV				R3 (AGR) Rover 200		HHR(AGR)Rover 400				
	Processing Times		Processing Times		Processing Times			Processing Times				Processing Times		Processing Times				
	O1	O2	O1	O2	O1	O2	O3	O1	O2	O3	O4	O1	O2	O1	O2	O3	O4	O5
1	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
2	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
3	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
4	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
5	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
6	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
7	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
8	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
9	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42
10	68	68	54	54	47	44	42	43	34	34	47	37	28	34	36	37	43	42

6.5.3 Industrial Case Study 3: Match Factory Peshawar, Pakistan (MFPP)

Figure 6.5 shows the flow line of a local match factory in Peshawar, Pakistan. The raw material, i.e., wood logs, are cut along the cross-section in 18″ long segments and peeled with a peeler machine, after which, they are converted into thin sheets and then stacked. These stacks of sheets are then chopped into sticks with the chopper machine, followed by drying, polishing and cleaning. The sticks are then processed for the match-head-chemical. Matchboxes, produced in a parallel process, are then filled. A pack of a boxing lot of ten is termed “Dozen”, and is produced through the box filling machine. The final process is called the grossing operation in which a pallet of a hundred dozens is produced.

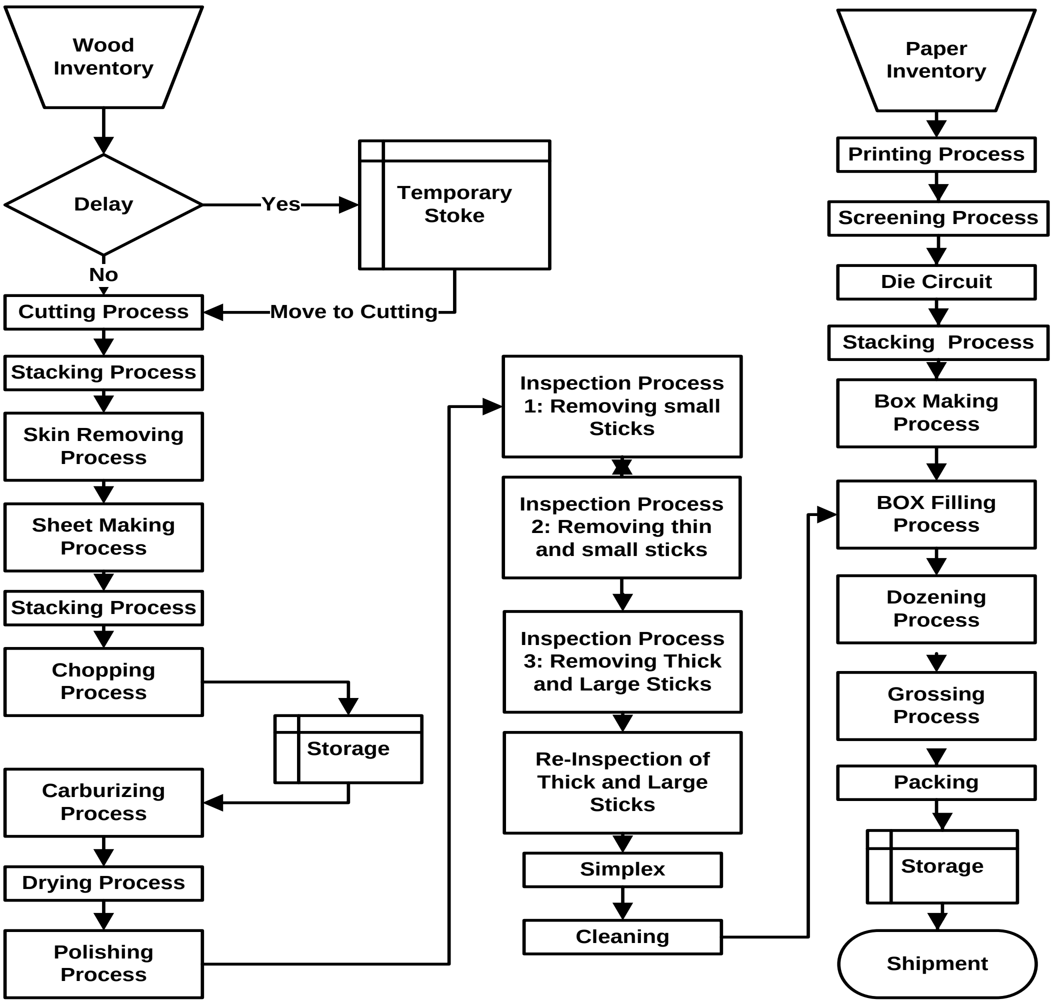


Figure 6. 5: Process Flowchart of Local Match Industry (Hussain et al., 2010)

The scheduling problem of match making is shown in Table 6.17.

Table 6. 17: Processing times for MFPP [Case Study 3]

Job No	O1 (Box Making 1)		O2 (Box Making 2)		O3 (Box Filling 1)		O4 (Box Filling 2)		O5 (Dozenining)		O6 (Grossing)	
	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
J1	M1	66.4	M2	0	M3	31	M4	0	M5	0	M6	3.6
J2	M1	66.4	M2	0	M3	31	M4	0	M5	0	M6	3.6
J3	M1	66.4	M2	0	M3	31	M4	0	M5	0	M6	3.6
J4	M1	66.4	M2	0	M3	31	M4	0	M5	0	M6	3.6
J5	M1	0	M2	20.2	M3	0	M4	27.8	M5	27.8	M6	3.6
J6	M1	0	M2	20.2	M3	0	M4	27.8	M5	27.8	M6	3.6
J7	M1	0	M2	20.2	M3	0	M4	27.8	M5	27.8	M6	3.6
J8	M2	0	M3	20.2	M4	0	M5	27.8	M6	27.8	M7	3.6

6.5.4 Results and Discussion of the Industrial Case Studies

Using the same combination of the GA parameters as discussed in Section 6.3, the HGA is applied to all the industrial case studies given in Tables 6.13 to 6.17.

Table 6.18 shows the computational result summary for the SASM's five scheduling problems to which the HGA was applied. The HGA obtained optimal solutions for all the problems for which the solution gap is 0% with Morshed's (Morshed, 2006) results at zero generation for all cases except two FSSPs in Case Study 1. The optimum result for these two problems is then achieved through HGA. All results (for 36 cases) are shown in Appendix C. Morshed (2006), claimed time savings of 3 to 4 folds with his scheduling system as compared to the traditional and current practices in industries. Achieving almost similar results with the HGA, the same amount of time reduction is claimed. However, the advantage of the HGA is that it achieved three solutions at zero generation number. Two problems, the JSSP2 and the JSSP3, for which achieving the optimal solutions with the HybH proved to be hard, were then achieved with HGA with generation numbers of 9 and 15 respectively.

Table 6. 18: Computational results from HGA for SASSM [Case Study 1]

Problem Name	**Size**	**Makespan Reported by Morshed (2006)**	**HybH**	**% GAP**	**Makespan by HGA**	**% GAP**
SASM JSSP1	8 x 6	505	505	0	-	-
SASM JSSP2	6 x 6	444	514		444	0
SASM JSSP3	6 x 6	379	425		379	0
SASM FSSP1	7 x 6	497	497	0	-	-
SASM FSSP2	6 x 6	452	452	0	-	-

Table 6.19 shows the summary of computational results of Pilkington PLC's 30 scheduling problems. The 1st and 2nd columns give information about the name of the problem and its size and the 3rd column shows the Makespan reported in literature by Morshed (2006). The 4th column shows the Makespan achieved by the HGA, where the % GAP is shown in the last column after comparing the HGA results with Morshed's (2006). The HGA obtained the optimum solution at zero generation number for all the problems with 0% solution gap. Hence, the computational cost is considerably low. The results are encouraging as the HGA achieved all the Makespans reported in literature, with 0 %GAP. Hence the HGA results show the usefulness of the developed module in the real-world environment.

Table 6. 19: Computational results from HGA for Pilkington's PLC [Case Study 2]

Problem Name	Prob. Size (N x M)	Makespan reported by Morshed (2006) and Noor (2007)	Makespan by HybH (Initial Solution)	% GAP
Honda CRV 1	5 x 4	346	346	0
Honda CRV 2	10 x 4	581	581	0
Honda CRV 3	25 x 4	1286	1286	0
Honda CRV 4	50 x 4	2461	2461	0
Honda CRV 5	75 x 4	3636	3636	0
Honda CRV 6	100 x 4	4811	4811	0
HHR (AGVR) Rover 400-1	5 x 5	364	364	0
HHR (AGVR) Rover 400-2	10 x 5	579	579	0
HHR (AGVR) Rover 400-3	25 x 5	1224	1224	0
HHR (AGVR) Rover 400-4	50 x 5	2299	2299	0
HHR (AGVR) Rover 400-5	75 x 5	3374	3374	0
HHR (AGVR) Rover 400-6	100 x 5	4449	4449	0
L319 heated-1	5 x 2	408	408	0
L319 heated-2	10 x 2	748	748	0
L319 heated-3	25 x 2	1768	1768	0
L319 heated-4	50 x 2	3468	3468	0
L319 heated-5	75 x 2	5168	5168	0
L319 heated-6	100 x 2	6868	6868	0
L319 non heated-1	5 x 2	324	324	0
L319 non heated-2	10 x 2	594	594	0
L319 non heated-3	25 x 2	1404	1404	0
L319 non heated-4	50 x 2	2754	2754	0
L319 non heated-5	75 x 2	4104	4104	0
L319 non heated-6	100 x 2	5454	5454	0
CB 40-1	5 x 3	321	321	0
CB 41-2	10 x 3	556	556	0
CB 42-3	25 x 3	1261	1261	0
CB 43-4	50 x 3	2436	2436	0
CB 44-5	75 x 3	3611	3611	0
CB 45-6	100 x 3	4786	4786	0

Table 6.20 shows MFPP FSSP results. The 1st and the 2nd columns show the problem name and size respectively and the 3rd Column shows the Makespan recorded in

literature by Altaf et al. (2010). The 4th and the 5th columns respectively show the Makespan achieved by HGA and the %GAP between the HGA and those from the results by Altaf et al. (2010). The %GAP is zero and once again the HGA proved to be an effective and viable scheduling model for real scheduling problems.

Table 6. 20: Computational results from HGA for MFPP [Case Study 3]

Problem Name	Size	Makespan Reported by Altaf et al., (2008)	Makespan by HGA	% GAP
MFPP	8 x 6	307.62	300.20	Improved

6.6 Summary of the Chapter

This chapter presents the results for the two developed novel heuristics (HybH and IBH) and the HGA. This chapter also covered in detailed the performance of the developed heuristics and HGA and showed that they performed extremely well for the benchmark JSSPs and FSSPs. The combined results of the job shop and flow shop benchmark problems are able to achieve the average %GAP of 0.08%. The HGA can be applied not only to JSSPs but also to other combinatorial scheduling problems such as FSSPs and batch-shop or process scheduling problems without any modifications. In most of the benchmark cases, the HGA obtained the optimum - only 6 cases out of 61 benchmark JSSPs and FSSPs could not be solved optimally, but where it achieved near-optimal values showing the feasibility of the schedules.

The HGA performance across the three different practical case studies (36 problems) has proved that the developed scheduling model can be applied to real-world scheduling problem for achieving optimal or near-optimal solutions. This shows the usefulness of the HGA in real-world scheduling problems. However, its performance may be improved further by incorporating some other evaluation and local search techniques.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Introduction

The main objective of the research was to develop a hybrid scheduling model based on AI techniques for JSSPs in order to find optimal or near-optimal solutions. In this research, novel heuristic rules have been developed. These rules were then combined with the GA in order to improve the GA's performance. The developed HGA was then applied to benchmark JSSPs for verification and validation. Details regarding the development of the heuristic rules, the HGA and their validation were described in Chapters 4, 5 and 6. This chapter describes the outcome of the techniques developed during this research. It also presents the future work recommendations both for the novel heuristics and the hybrid GA.

7.2 Research Achievement

The main objective of this research was to develop a hybrid scheduling model for solving JSSPs in order to find optimal or near optimal solution for the selected performance criteria of Makespan. The objectives of this research as outlined in Chapter 1, have successfully been achieved with the development, implementation, verification and validation of two new priority heuristic rules and their interfacing with GAs.

The scheduling problems and their solution approaches are thoroughly assessed before the development and implementation of the approaches. The developed approaches addressed the inherent difficulties of the scheduling problems due to which no heuristic can guarantee optimal or near optimal results for all problem sizes. The performance of IBH, HybH and HGA successfully addressed these issues to some extent by achieving optimal, near-optimal and better initial generation results and is therefore a significant contribution to performance improvement of the scheduling problems in general and JSSPs in particular (see Chapter 6). A summary of the research activities is shown in Figure 7.1

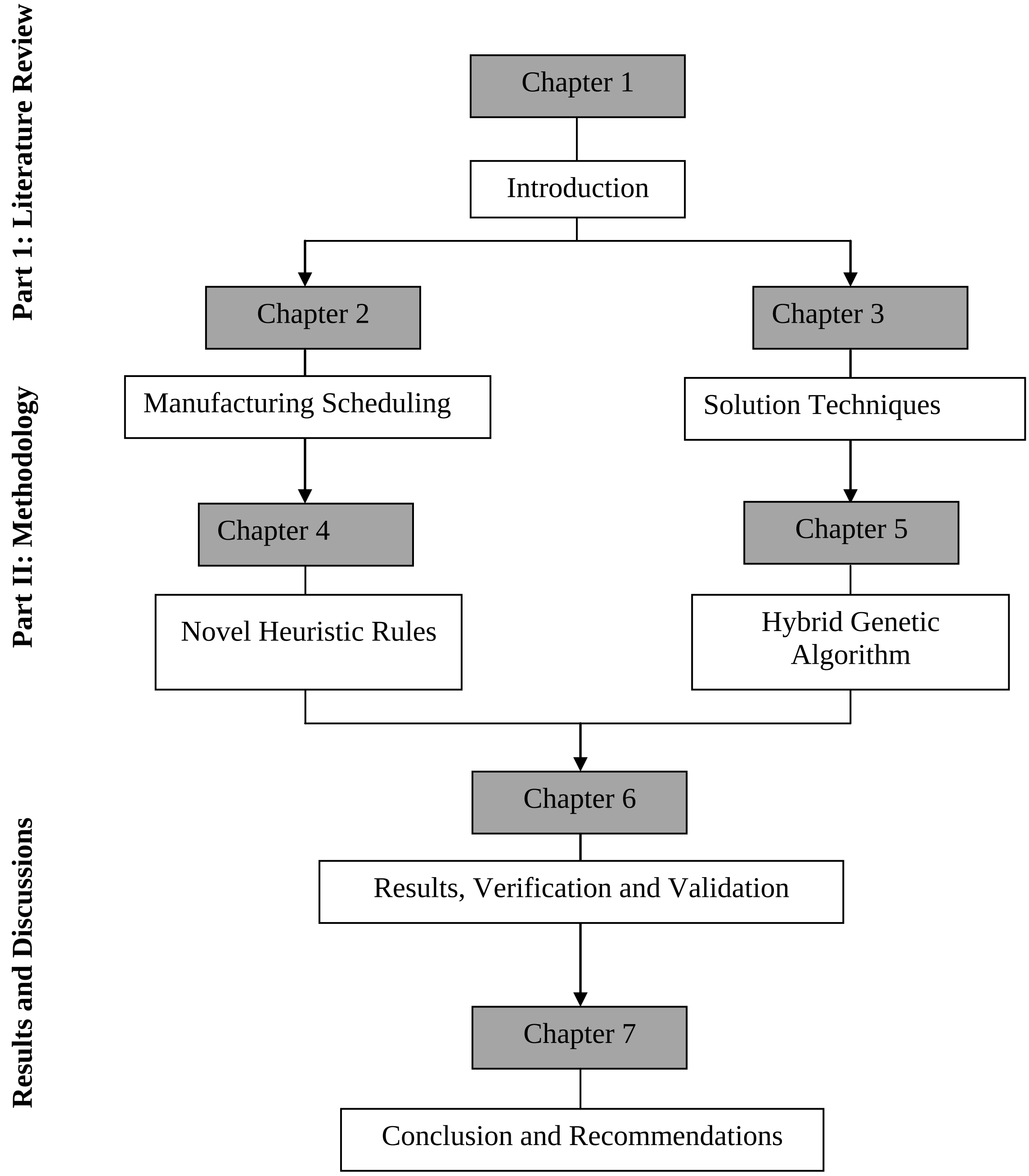


Figure 7. 1: Summary of research activities

Chapters 1, 2 and 3 of the research have proceeded from an introduction to the background of manufacturing scheduling a well known and common problem in all industries followed by an introductory literature review to manufacturing problems in general and JSSPs in particular. The techniques used to solve these problems were also reviewed to gain knowledge and fulfill the main objective: the development of hybrid technique using Artificial Intelligence (AI) to achieve optimal or near-optimal solutions. In Chapter 2, an introduction to different manufacturing environments in manufacturing systems was presented, followed by introduction, mathematical model, scheduling criteria and complexity issues of JSSPs. The sources and types of benchmark problems are presented in the chapter. In Chapter 3, details of the literature review and analysis revealed many solution approaches applied to JSSP over the past few decades and also showed that GAs are dominating in the list due to their search capabilities among all approaches.

However, GA fine tuning problems shifted researcher's focus to hybrid approaches mainly combining GA with other AI techniques or introducing heuristics to main GA loops such as local search heuristics. The review also indicated that the initial solution in GA or HGA can significantly effect the JSSP solutions. The fitter the initial solution the faster the GA will converge with a better solution. It is therefore recommended that a new heuristic rule-based systems must be developed which can provide stable results across the problem sizes and can be incorporated with AI techniques such as GA. Such heuristic rules and hybrid approaches would not only be applicable to JSSP but also to solve other complex combinatorial and real life problems.

7.3 Novel Heuristics for scheduling problems

In Chapter 4 of the research, the design and development process for two novel heuristic rules: the Index Based Heuristic (IBH) and the Hybrid Heuristics Rule (HybH) was presented. The proposed heuristic rules were applied to benchmark JSSPs from the literature and case studies in order to check the validity and effectiveness of the proposed heuristics (see Chapter 6).

7.3.1 Index Based Heuristic (IBH) Rule for scheduling problem

An IBH was developed for scheduling problems (see Chapter 4, Section 4.2). The developed IBH was tested against *minimum-Makespan* JSSPs. The heuristic calculates the indices, called as Index Values (IVal) of the candidate jobs and then assigns the jobs to the available machine in the ascending order of the index values, i.e., jobs with lower or shorter index values are assigned first. To minimize the idle time between jobs, a swap technique was introduced at a later stage, when the algorithm initially fails to achieve optimum value, after all candidate jobs had been assigned. The swap technique takes the candidate jobs for a machine and swaps them without violating the precedence constraint.

7.3.1.1 Performance of IBH

The proposed IBH overcame the deficiencies in the traditional heuristics and yielding solutions with greater %GAP from optimal results. The IBH performed well across all the test-bed benchmark problems (see Chapter 6, Section 6.2.2) and successfully achieved new optimal or near-optimal solutions for the JSSPs. For example, it had lesser % GAP value of 5.4% in comparison with that of the best traditional heuristics (LPT and MS rules) that have an overall mean *GAP* value of

10.5%. Hence, the IBH reduced the overall % *GAP* by 94.4%, which is again a significant increase in process efficiency.

7.3.2 Hybrid Heuristic (HybH) for scheduling problems

The Hybrid Heuristic (HybH) solution approach for scheduling problems is presented with the objective of optimizing the overall Makespan (C_{max}). The proposed HybH is a combination of the IBH and the Finished Job-Based (FJB) Heuristic. The HybH assigned candidate job's first operation on a machine using the IBH and the remaining operations on the basis the FJB.

7.3.2.1 Performance of HybH

The HybH results indicated that it performed well and consistently across the test-bed benchmark problems. For example, the overall mean % GAP taken across the LA-problems, HybH had a lesser % GAP value of 6% in comparison with best results from traditional heuristics (the LPT and the MS rules), which have an overall mean GAP value of 10.5%. The HybH reduced the overall % GAP by 77.9% in comparison to traditional heuristics on selected benchmark cases, which reflected a considerable gain in the efficiency (see section 6.2.1).

Some of the observations and conclusions regarding the individual models and recommendations for future work are presented in the following sections.

7.3.3 Future Recommendations regarding IBH and HybH

Both the heuristics, i.e., the IBH and the HybH, have shown encouraging results and are valid methodologies for scheduling optimization. The proposed heuristic rules overcame the deficiencies of the traditional heuristics for manufacturing scheduling and performed well across all the test-bed benchmark problems and successfully achieved optimal or near-optimal solutions for the same.

The evaluation process in the GA for a JSSP is a key step that determines the fitness of the objective function. In this research, only the HybH is used in the main GA loop in the evaluation process and initial solution of the benchmark problems (Chapter 5). However, the IBH was not used for the same purpose. In the future, the IBH may be applied to the evaluation process in combination with GA as well. Therefore, future work may focus on the hybridization of the IBH with other optimization techniques. The IBH may also be applied to some larger benchmark and real scheduling problems.

7.4 Hybrid Genetic Algorithm for scheduling problems

The hybrid GA-based approach towards solving scheduling problems was developed during this research. The GA uses the job-based chromosome representation, a multipoint crossover, mutation and permutation technique for the selection of chromosomes. The operational performance of the HGA was tested in detail using benchmark problems of various sizes and 36 industrial case studies from the literature with problem size ranging from 5x6 (five jobs and six machines) to 100x5, in order to gauge its capabilities, provide a reference for future research in this area and to fill the gap for parametric analysis for GAs. The performance of the HGA was satisfactory and obtained optimal solutions for almost all benchmark problems and industrial case studies.

7.4.1 Performance of HGA

The developed HGA was tested against 62 of *minimum-Makespan* benchmark JSSPs. It successfully achieved optimum results for 59 (out of 62) problems. The HGA was also tested on 36 case studies from literature, and it achieved the optimum results for 35 cases with % GAP and record improvement in one case study. This showed the usefulness of the HGA in real-world scheduling problems.

7.4.2 Recommendations for future work in HGA

(i) The initial solution can be generated by various methods such as heuristics and dispatching rules. Only the HybH is used in this research for producing the initial set of chromosomes due to the reason that it has performed well across a wide range of problems and produced better initial solutions, which may increase the chances of hitting the global optimum with lesser searching efforts, i.e., low computational costs. However, it may be fruitful to use the IBH and some other evaluation techniques in order to improve the performance.

(ii) Scheduling problems are a multi-objectives problems. The HGA in this research is limited to only the Makespan as an objective function. However, the HGA can provide a useful platform for future studies to treat scheduling as a multi-objective problem.

(iii) The present HGA is developed for a deterministic scheduling problem. However, in the future, this model may be extended to stochastic scheduling, where the arrival of a job will have some probability distribution.

(iv) The HGA may be coupled with local search techniques for fine-tuning of the HGA solutions in certain problems, which may enable the HGA to achieve optimal solutions that the current HGA could not achieve.

(v) There are many chromosome representations available in literature. However, a job-based representation of chromosomes is used in this research. In the future, other chromosome representations may be used in order to improve the performance.

(vi) The performance of the HGA also depends on the selection of the crossover and mutation operators. The operators that have been selected and used in this research need further investigation.

(vii) At present, the developed HGA is not user friendly due to weak GUI functions in MATLAB. In the future, a suitable front-end is needed to make it more user friendly.

(viii) The HGA is developed in MATLAB environment and the poor GUI interface with no built-in functions for charts, make it really hard to produce some fine colour schedule charts. However, a code for the Gantt chart is developed separately in MATLAB. Currently, these functions are good for small-sized scheduling problems. In the future, these functions may be used to produce better machine or job Gantt charts by adding more functions and options.

7.5 Reflection of the Research Work

The learning, research work, and writing process were a genuine learning experience. The research work is an addition to the knowledge of manufacturing scheduling. Although a number of algorithms and heuristics are available to address the scheduling problems; whether a job shop or a flow shop, but in this research work an effort is made to improve the performance criterion (Makespan), which is mainly considered for such scenarios. Considering the importance of the scheduling problem, both scenarios (job shop and flow shop) were considered and three new techniques (HGA, IBH, HybH) and *new process* for developing heuristic rules were developed and their results compared with the benchmark problems. In some cases (problems), the improvement is witnessed whilst in some cases the same results are achieved. Still, one feels, the large size and hard problems are far from being completely solved due to a large number of combinations and exponential complexities of computational time. Considering such large sized problems, this research work looks as an addition of a drop to the ocean of scheduling knowledge. There is definitely a room to improve the performance of AI techniques, local search algorithms and heuristics.

In this research, as a global search tool GA was selected. For local search, researchers have used many techniques such as Artificial Neural Networks (ANN), Fuzzy Logic (FL), Simulated Annealing (SA), Expert Systems (ES), Tabu Search (TS), Perti net, and Heuristic rules. All these techniques have been extensively applied to the problems with GA in order to improve the single objective. However, they can be combined either with each other or the developed new heuristic rules in order to check for any improvement in single or multi objectives. For example, development in ANN techniques such as Feedforward ANN and Hopfield ANN methods can be utilized in global or in local search in combination with GAs.

The two new heuristics that have been developed during the current research can also be used in combination with existing techniques such as ANN, FL, ES, TS, etc. and their strength and weaknesses can be gauged by applying them to large and hard problems.

7.6 Conclusion

The chapter has highlighted the discussion regarding the two news priority heuristic rules and the hybrid genetic algorithm. The developed algorithms focused on solving *minimum-Makespan* scheduling problems. The chapter also reviews the achievement of the objectives of the research as outlined in Chapter 1. Furthermore, the performance of the new developed algorithms, limitation and the recommendations for the future work have been discussed. As shown, the algorithms on the basis of their performance, offers an alternate reliable and a potential optimization technique for scheduling problems. The shortcomings of the Heuristics and the HGA are discussed and recommendations are made for future work.

.

References

Adam, N. R. and Surkis, J. (1980). "Priority Updates Intervals and Anomalies in Dynamic Ratio Type Jop Shop Scheduling Rules." Management Science **26**(12): 1227-1237.

Adams, J., Balas, E. and Zawack, D. (1988). "The shifting bottleneck procedure for job shop scheduling." Management Science **34**(3): 391-401.

Adams, J., Balas, E. and Zawack, D. (1998). "The shifting Bottleneck Procedure for Job Shop Scheduling, ." Management Science 34,391-401.

Akyol, D. E. (2004). "Application of neural networks to heuristic scheduling algorithms." Computers & Industrial Engineering **46**(4): 679-696.

Akyol, D. E. and Bayhan, G. M. (2007). "A review on evolution of production scheduling with neural networks." Computers & Industrial Engineering **53**(1): 95-122.

Akyol, D. E. and Bayhan, G. M. (2008). "Multi-machine earliness and tardiness scheduling problem: an interconnected neural network approach." International Journal of Advanced Manufacturing Technology **37**(5-6): 576-588.

Anderson, E. J. and Nyirenda, J. C. (1990). "2 New Rules for Minimze Tardiness in a job Shop." International Journal of Production Research **28**(12): 2277-2292.

Andersson, P. (2008). Reuse of manufacturing experience in product and process definitions Department of Applied Physics and Mechanical Engineering. Luleå, Luleå University of Technology, Sweden.

Applegate, D. and Cook, W. (1991). "A computational study of the job-shop scheduling problem." Informs Journal of Computing **3**(2): 149-156.

Askin, R. G. and Standridge, C. R. (1993). Modeling and Manufacturing System, John Wiley.

AWAD, E. M. (1995). Building expert systems : principles, procedures, and applications, New York, West Publishing Company.

Bai, S. (1998). "NSF Workshop on supply chain management in eletronic commerce, ." Final Report http://www.ise.ufl.edu/Supplychain/.

Baker, C. T. and Dzielinski, B. P. (1960). "Simulation of a Simplified Job Shop." Management Science **6**(3): 311-323.

Baker, J. E. (1987). Reducing Bias and Inefficiency in the Selection Algorithm. Proceedings of the Second International Conference on Genetic Algorithms and their Application New Jersey, Hillsdale.

Baker, K. R. (1974). Introduction to Sequencing and Scheduling, John Wiley & Sons Inc

Balas, E. (1965). "An additive algorithm for solving linear programs with zero-one variables." Operations Research **13**: 517-546.

Balas, E. (1967). "Discrete programming by the filter method." Operations Research **15**: 915-957.

Balas, E. (1970). "Machine Sequencing - Disjunctive Graph and Degree - Constrained Subgraphs." Naval Research Logistics Quarterly **17**(1): 1-&.

Baptiste, P. and Timkovsky, V. G. (2004). "Shortest path to nonpreemptive schedules of unit-time jobs on two identical parallel machines with minimum total completion time." Mathematical Methods of Operations Research **60**(1): 145-153.

Barker, J. R. and McMahon, G. B. (1985). "Scheduling The General Job Shop." Management Science **31**(5): 594-598.

Bean, J. (1994). "Genetic Algorithms and Random Keys for Sequencing and Optimization." Journal on Computing Spring **6**(2): 154-160.

Benavides, J. A. C. and Prado, J. C. (2002). "Creating an expert system for detailed scheduling." International Journal of Operations & Production Management **22**(7-8): 806-819.

Benders, J. F. (2005). "Partitioning procedures for solving mixed-variables programming problems." Computational Management Science **2**(1): 3-19.

Biegel, J. E. and Davern, J. J. (1990). "Genetic Algorithm and Job Shop Scheduling." Computers & Industrial Engineering **19**(1-4): 81-91.

Bierwirth, C. (1995). "A Generalized Permutation Approach to Job Shop Scheduling with GAs." Or Spektrum **17**(2-3): 87-92.

Bierwirth, C., Mattfeld, D. C. and Kopfer, I. (1996). On Permutation Representations for Scheduling Problems. PPSN'IV Parallel Problem Solving from Nature, , Berlin, Heidelberg, Germany, Springer-Verlag.

Black, M. (1937). "Vagueness. An Exercise in Logical Analysis." Philosophy of Science **4**(4): 427-455.

Blazewicz, J., Dror, M. and Weglarz, J. (1991). "Mathematical Programming Formulations for Machine Scheduling: A Survey." European Journal of Operational Research, Invited Review **51**(3): 283-300.

Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G. and Weglarz, J. (1996). "Job Shop Scheduling, Scheduling Computer and Manufacturing Processes, ." Springer, Germany: 16-20 and 265-317.

Bowman, E. H. (1959). "The Schedule-Sequencing Problem." Operations Research **7**(621-624).

Bozejko, W., Pempera, J. and Smutnicki, C. (2009). Parallel Simulated Annealing for the Job Shop Scheduling Problem. Computational Science - Iccs 2009, Part I. G. Allen, J. Nabrzyski, E. Seidel, J. Dongarra, G. D. VanAlbada and P. M. A. Sloot. Berlin, Springer-Verlag Berlin. **5544:** 631-640.

Braune, R., Wagner, S. and Affenzeller, M. (2005). On the Analysis of Crossover Schemes for Genetic Algorithms Applied to the Job Shop Scheduling Problem. Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI), International Institute of Informatics and Systemics, 2005.

Brucker, P. (1988). "An Efficient algorithm for the job Shop Problem with 2 jobs." Computing **40**(4): 353-359.

Brucker, P. (2007). Scheduling Algorithms, Springer-Verlag New York, Inc.

Bureerat, S. and Limtragool, J. (2008). "Structural topology optimisation using simulated annealing with multiresolution design variables." Finite Elements in Analysis and Design **44**(12–13): 738-747.

Canbolat, Y. B. and Gundogar, E. (2004). "Fuzzy priority rule for job shop scheduling." Journal of Intelligent Manufacturing **15**(4): 527-533.

Carlier, J. (1978). "Ordonnancements-A Constraints Disjunctives." Rairo-Recherche Operationnelle-Operations Research **12**(4): 333-351.

Chan, F. T. S., Chan, H. K. and Kazerooni, A. (2003). "Real time fuzzy scheduling rules in FMS." Journal of Intelligent Manufacturing **14**(3-4): 341-350.

Chang, P. T., Yao, M. J., Huang, S. F. and Chen, C. T. (2006). "A genetic algorithm for solving a fuzzy economic lot-size scheduling problem." International Journal of Production Economics **102**(2): 265-288.

Chang, Y. L., Sueyoshi, T. and Sullivan, R. S. (1996). "Ranking dispatching rules by data envelopment analysis in a job shop environment." Iie Transactions **28**(8): 631-642.

Chaudhry, I. A. and Drake, P. R. (2008). "Minimizing flow-time variance in a single-machine system using genetic algorithms." International Journal of Advanced Manufacturing Technology **39**(3-4): 355-366.

Chen, R. M. and Huang, Y. M. (2001). "Multiprocessor task assignment with fuzzy Hopfield neural network clustering technique." Neural Computing & Applications **10**(1): 12-21.

Chen, T. (2009). "Dynamic fuzzy-neural fluctuation smoothing rule for jobs scheduling in a wafer fabrication factory." Proceedings of the Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering **223**(I8): 1081-1094.

Cheng, K., Harrison, D. K. and Pan, P. Y. (1998). "Implementation of agile manufacturing - an AI and Internet based approach." Journal of Materials Processing Technology **76**(1-3): 96-101.

Cheng, R. W., Gen, M. and Tsujimura, Y. (1996). "A tutorial survey of job-shop scheduling problems using genetic algorithms .1. Representation." Computers & Industrial Engineering **30**(4): 983-997.

Cheng, R. W., Gen, M. and Tsujimura, Y. (1999). "A tutorial survey of job-shop scheduling problems using genetic algorithms: Part II. Hybrid genetic search strategies." Computers & Industrial Engineering **37**(1-2): 51-55.

Chiang, T. C. and Fu, L. C. (2004). Solving the FMS scheduling problem by critical ratio-based heuristics and the genetic algorithm, . IEEE Conference on Robotics and Automation.

Chiang, T. C. and Fu, L. C. (2007). "Using dispatching rules for job shop scheduling with due date-based objectives." International Journal of Production Research **45**(14): 3245-3262.

Coffman, E. G., Dereniowski, D. and Kubiak, W. (2012). "An efficient algorithm for finding ideal schedules." Acta Informatica **49**(1): 1-14.

Coffman, E. G., Sethuraman, J. and Timkovsky, V. G. (2003). "Ideal preemptive schedules on two processors." Acta Informatica **39**(8): 597-612.

Conway, R. W. (1964). "An Experimental Investigation of Priority Assignment in a Job Shop." Santa Monica, CA: RAND Corporation, 1964. (RM-3789-PR).

Conway, R. W., Maxwell, W. L. and Miller, L. W. (1967). Theory of Scheduling, Adison Wesley,Reading, MA.

Croce, F., Tadei, R. and Volta, G. (1995). "A genetic algorithm for the job shop problem." Complex Systems **22**: 15-24.

Cruz-Chavez, M. A. and Frausto-Solis, J. (2004). Simulated annealing with restart to job shop scheduling problem using upper bounds. Artificial Intelligence and Soft Computing - Icaisc 2004. L. Rutkowski, J. Siekmann, R. Tadeusiewicz and L. A. Zadeh. Berlin, Springer-Verlag Berlin. **3070:** 860-865.

Dantzig, G. (1963). Linear Programming and Extensions, Princeton University Press and the RAND Corporation.

Dantzig, G. B. and Wolfe, P. (1960). "Decomposition Principle for Linear Programs " Operations Research **8**: 101-111.

Davis, L. (1985). Job Shop Scheduling with Genetic Algorithms. Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc.**:** 136-140.

Davis, L. (1991). Handbook of Genetic Algorithms, Van Nostrand Reinhold.

De Jong, K. A. (1993). Genetic Algorithms are not function optimizers, Computer Science Department George Mason University Fairfax, VA 22030, USA.

Demirkol, E., Mehta, S. and Uzsoy, R. (1997). "A Computational Study of Shifting Bottleneck Procedures for Shop Scheduling Problems." Journal of Heuristics **3**(2): 111-137.

Devillers, J. (1996). Genetic Algorithms in Molecular Modeling, Elsevier.

Dominic, P. D. D., Kaliyamoorthy, S. and Kumar, M. S. (2004). "Efficient dispatching rules for dynamic job shop scheduling." International Journal of Advanced Manufacturing Technology **24**(1-2): 70-75.

Dorndorf, U. and Pesch, E. (1995). "Evolution Based Learning in A Job Shop Scheduling Environment." Computers & Operations Research **22**(1): 25-40.

Falkenauer, E. and Bouffoix, S. (1991). A genetic algorithm for job shop. . IEEE International Conference on Robotics and Automation, Sacrcment, California, USA.

Fang, H. L., Ross, P. and Corne, D. (1993). "A Promising Genetic Algorithm approach to Job-Shop Scheduling Rescheduling , and Open -Shop Scheduling Problem." Proceedings of the Fifth International Conference on Genetic Algorithms: 375-382.

Fargier, H. (1996). Fuzzy scheduling: Principles and experiments. New York, Wiley.

Fisher, H. and Thompson, G. L. (1963). Probabilistic learning combinations of local job shop scheduling rules. Industrial Scheduling. J. F. Muth and G. L. Thompson Eds. Englewood Cliffs, New Jersey., Prentice Hall. **15**.

French, S. (1982). Sequencing and Scheduling. Mathematics and its applications, Ellis Hor-wood Limited.

Gantt, H. (1919). "Efficiency and Democracy, ." Transactions of the American Society OF Mechanical Engineer **40**: 799-808.

Gao, J., Gen, M., Sun, L. and Zhao, X. (2007). "A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems." Computers & Industrial Engineering **53**(1): 149-162.

Gao, J. Q., He, G. X. and Wang, Y. S. (2009). "A new parallel genetic algorithm for solving multiobjective scheduling problems subjected to special process constraint." International Journal of Advanced Manufacturing Technology **43**(1-2): 151-160.

Garey, M. R. and Johnson, D. S. (1978). "Strong NP-Completeness Results - Motivation, Examples, and Implications." Journal of the Acm **25**(3): 499-508.

Garey, M. R. and Johnson, D. S. (1979). Computers and intractability: A guide to the theory of NP-completeness. New York, W.H. Freeman and Company.

Gen, M., Lin, L. and Cheng, R. (2008). Network Models and Optimization: Multiobjective Genetic Algorithm Approach, Springer-Verlag London Limited.

Gen, M., Tsujimura, Y., Kubota, E. and Ieee (1994). Solving Job Shop Schduling Problems by GA. New York, I E E E.

Giffler, B. and Thompson, G. L. (1960). "Algorithms for Solving Production-Scheduling Problems." Operations Research **8**(4): 487-503.

Giglio, R. J. and Wagner, H. M. (1964). "Approximate Solutions to the Three-Machine Scheduling Problem." Operations Research **12**(2): 305-324.

Goldberg, D. E. (1989). Genetic Algorithm in Search, Optimization and Machine Learning Addisson Wesley.

Goncalves, J. F., Mendes, J. and Resende, M. G. C. (2005). "A hybrid genetic algorithm for the job shop scheduling problem." European Journal of Operational Research **167**(1): 77-95.

Gonzalez, T. and Sahni, S. (1978). "Flow Shop and Job Shop Schedules - Complexities and Approximation " Operations Research **26**(1): 36-52.

Gordon, V. S. and Whitley, D. (1993). Serial and Parallel GAs as Function Optimizers.

Grabot, B. and Geneste, L. (1994). "Dispatching Rules in Scheduling - A Fuzzy Approach." International Journal of Production Research **32**(4): 903-915.

Graham, R. L., Lawler, E. L., Lenstra, J. K. and Kan, A. H. G. R. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. Annals of Discrete Mathematics. E. L. J. P.L. Hammer and B. H. Korte, Elsevier. **Volume 5:** 287-326.

Halshal, D. N. (1995). A Resource Scheduling Framework for Small Manufacturing Enterprises, University of Bradford, Uk. **PhD**.

Hart, E., Ross, P. and Corne, D. (2005). "Evolutionary Scheduling: A Review." Genetic Programming and Evolvable Machines **6**(2): 191-220.

Hefetz, N. and Adiri, I. (1982). "An Efficient Optimal Algorithm for the Two-Machines Unit-Time Jobshop Schedule-Length Problem." MATHEMATICS OF OPERATIONS RESEARCH **7**(3): 354-360.

Hitomi, K. (1996). Manufacturing Systems engineering – A unified approach to manufacturing technology, production management, and industrial economics, Taylor and Francis Ltd, London UK,.

Holland, J. (1975). Adaptation in Neural and Artificial Systems, The University of Micihigan Press, Ann Arbor.

Holthaus, O. and Rajendran, C. (1997). "Efficient dispatching rules for scheduling in a job shop." International Journal of Production Economics **48**(1): 87-105.

Holthaus, O. and Rajendran, C. (1997). "New dispatching rules for scheduling in a job shop - An experimental study." International Journal of Advanced Manufacturing Technology **13**(2): 148-153.

Hussain, A., Shakoor, Q., Abid, H., Omair, M. and Noor, S. (2010). Intelligent Scheduling System for Flow Shop and Job Shop Scheduling Problems in Discrete Manufacturing. Peshawar, NWFP University of Engineering Science and Technology.

Ishibuchi, H., Yamamoto, N., Misaki, S. and Tanaka, H. (1994). "Local Search Algorithm for Flow Shop Scheduling with Fuzzy Due-Dates." International Journal of Production Economics **33**(1-3): 53-66.

Ishii, H. and Tada, M. (1995). "single Machines Scheduling Problem with Fuzzy Precedence Relation." European Journal of Operational Research **87**(2): 284-288.

Jackson, J. R. (1956). "An extension of Johnson’s result on job lot scheduling. Naval Research Logistics Quarterly 3 (3), 201–203." Naval Research Logistics Quarterly **3**(3): 201-203.

Jain, A. S. and Meeran, S. (1998). "Job-shop scheduling using neural networks." International Journal of Production Research **36**(5): 1249-1272.

Jain, A. S. and Meeran, S. (1999). "Deterministic job-shop scheduling: Past, present and future." European Journal of Operational Research **113**(2): 390-434.

Jain, A. S. and Meeran, S. (2002). "A Multi-level Hybrid Framework Applied to the General Flow-shop Scheduling Problem." Computers and Operations Research, **29**(13): 873-1901.

Jamili, A., Shafia, M. A. and Tavakkoli-Moghaddam, R. (2011). "A hybrid algorithm based on particle swarm optimization and simulated annealing for a periodic job shop scheduling problem." International Journal of Advanced Manufacturing Technology **54**(1-4): 309-322.

Jayamohan, M. S. and Rajendran, C. (2004). "Development and analysis of cost-based dispatching rules for job shop scheduling." European Journal of Operational Research **157**(2): 307-321.

Johnson, D. S., Gutin, L. A., McGeoch, L. A., Yeo, A., Zhang, W. and Zverovich, A. (2002). Experimental analysis of heuristics for the ATSP. The traveling salesman problem and its variations, Dordrecht, Kluwer Academic Publishers.

Johnson, S. M. (1954). "Optimal two- and three-stage production schedules with setup times included." Naval Research Logistics Quarterly **1**(1): 61-68.

Jones, A. and Rabelo, J. C. (1998) "Survey of job shop scheduling techniques." NISTIR, National Institute of Standards and Technology, Gaithersburg, MD.

Kachitvichyanukul, V. and Sitthitham, S. (2011). "A two-stage genetic algorithm for multi-objective job shop scheduling problems." Journal of Intelligent Manufacturing **22**(3): 355-365.

Kantorovich, L. V. (1940). " A new method of solving some classes of extremal problems." Doklady Akad Sci USSR **28**: 211-214.

Kawai, T., Fujimoto, Y. and Ieee (2005). An efficient combination of dispatch rules for Job-Shop Scheduling Problem. New York, Ieee.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983). "optimization by Simulated Annealing " Science **220**(4598): 671-680.

Kis, T. (2003). Job-shop scheduling with processing alternatives. 18th EURO Summer-Winter-Institute on Meta-Heuristics in Combinatorial Optimization (ESWI XVIII), Switzerland, Elsevier Science Bv.

Kobayashi, S., Ono, I. and Yamamura, M. (1995). An Efficient Genetic Algorithm for Job Shop Scheduling Problems,. Proceedings of the Sixth International Conference on Genetic Algorithms, San Francisco, CA, USA, Morgan Kaufmann

Kolonko, M. (1999). "Some new results on simulated annealing applied to the job shop scheduling problem." European Journal of Operational Research **113**(1): 123-136.

Kubiak, W., Sethi, S. and Sriskandarajah, C. (1995). "An Efficient Algorithm for Job Shop Scheduling." Annals of Operations Research **57**: 203-216.

Kuroda, M. and Wang, Z. (1996). "Fuzzy job shop scheduling." International Journal of Production Economics **44**(1-2): 45-51.

Lai, P. J. and Wu, H. C. (2009). "Using Ant Colony Optimization to Minimze the Fuzzy Makespan and Total Weighted Fuzzy Completion Time in Flow Shop Scheduling Problem." International Journal of Uncertainty Fuzziness and Knowledge-Based Systems **17**(4): 559-584.

Lawrence, S. (1984). "Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement), ." Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg, 1984.

Lawrence, S. (1984). "Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques (Supplement)." Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg.

Lei, D. M. (2010). "A genetic algorithm for flexible job shop scheduling with fuzzy processing time." International Journal of Production Research **48**(10): 2995-3013.

Li, J. Q., Sun, K. B., Xu, D. H. and Li, H. X. (2010). "Single machine due date assignment scheduling problem with customer service level in fuzzy environment." Applied Soft Computing **10**(3): 849-858.

Liaw, C. F. (2000). "A hybrid genetic algorithm for the open shop scheduling problem." European Journal of Operational Research **124**(1): 28-42.

Lintern, R. R. H. a. G. (2006). Eliciting and representing the knowledge of experts. Cambridge handbook of expertise and expert performance. K. A. In Ericsson, Charness, N., Feltovich, P., & Hoffman, R. New York:, Cambridge University Press.**:** 203-222.

Lopez, P. and Roubellat, F., Eds. (2008). Production Scheduling. London, John Wiley.

Low, C. Y. and Yeh, Y. L. (2009). "Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated." Robotics and Computer-Integrated Manufacturing **25**(2): 314-322.

Luh, Y. L. and Guan, P. B. X. (1994). Fuzzy optimization-based scheduling of identical machines with possible breakdown. IEEE International Conference on Robotics and Automation.

Maccarthy, B. L. and Liu, J. (1993). "Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling." International Journal of Production Research **31**(1): 59-79.

Manne, A. S. (1960). "Job-Shop Scheduling Problem." Operations Research **8**: 219-223.

Maqsood, S., Hussain, I., Khan, M. K. and Wood, A. S. (2011). A Novel Index Based Heuristic for JSSP. 26th International Conference of CAD/CAM, Robotics & Factories of the Future. Kuala Lumpur, Malaysia.

Maqsood, S., Khan, M. K. and Wood, A. S. (2010). A Review of AI techniques for Manufacturing Scheduling. The 25th International conference of CAD/CAM, Robotics & Factories of the Future. Pretoria, South Africa.

Maqsood, S., Khan, M. K. and Wood, A. S. (2011). A novel heuristic for low batch manufacturing process scheduling optimization Computer Aided Process Engineering (CAPE) Forum. School of Engineering, Design & Technology. University of Bradford, UK.

Maqsood, S., Khan, M. K. and Wood, A. S. (2011). "A novel heuristic for low batch manufacturing process scheduling optimization with reference to process engineering." Chemical Product and Process Modeling **6**(2).

Maqsood, S., Noor, S., Khan, M. and Wood, A. (2011). Sensitivity analysis of genetic algorithms for job shop scheduling problems. 26th International Conference of CAD/CAM, Robotics & Factories of the Future. Kuala Lumpur, Malaysia.

Martin, B. (1996). A Time-Oriented Approach to Computing Optimal Schedules for the Job-Shop Scheduling Problem. School of Operations Research & Industrial Engineering. New York, Cornell University, USA. **PhD**.

Matta, M. E. (2009). "A genetic algorithm for the proportionate multiprocessor open shop." Computers & Operations Research **36**(9): 2601-2618.

McCulloch, W. S. and Pitts, W. (1943 &1990). "A Logical Calulus of the Ideas Immanent in Nervous Activity (Reprinted from Bulletin of Mathematical

BioPhysics, VOL 5, PG 115-133, 1943)." Bulletin of Mathematical Biology **52**(1-2): 99-115.
Mehrabad, M. S. and Pahlavani, A. (2009). "A fuzzy multi-objective programming for scheduling of weighted jobs on a single machine." International Journal of Advanced Manufacturing Technology **45**(1-2): 122-139.
Messenger, R. and Dove, R. (2012). Basic Genetic Algorithm pattern for use in self-organizing agile security. Security Technology (ICCST), 2012 IEEE International Carnahan Conference on.
Metaxiotis, K. S., Psarras, J. E. and Askounis, D. T. (2002). "GENESYS: an expert system for production scheduling." Industrial Management & Data Systems **102**(5-6): 309-317.
Minoux, M. (1986). Mathematical programming: Theory and algorithms. Chichester W. Sussex and New York, Wiley
Mohamed, N. M. Z. N. and Khan, M. K. (2011). "Knowledge Based System Implementation for Lean Process in Low Volume Automotive Manufacturing (LVAM) with Reference to Process Manufacturing." Chemical Product and Process Modeling **6**(2).
Momim, M. (1999). A knowledge-Based Approach for the Dynamic Scheduling and Sequencing in Manufacturing Cells, University of Bradford, UK. **PhD**.
Morshed, M. S. (2006). A hybrid model for job shop scheduling. Department of Mechanical and Manufacturing, University of Birmingham, UK. **PhD**.
Morton, T. E. and Pentico, D. W. (1993). Heuristic Scheduling Systems with Applications to Production Systems and Project Management. . New York, John Wiley and Sons.
Mosheiov, G. (2002). "Complexity analysis of job-shop scheduling with deteriorating jobs." Discrete Applied Mathematics **117**(1-3): 195-209.
Nababan, E. B., Abdullah, S., Hamdan, A. R. and Zakaria, M. S. (2008). A Branch and Bound Algorithm in Optimizing Job Shop Scheduling Problems.
Negnevitsky, M. (2002). Artificial Intelligence: A Guide to Intelligent Systems Essex, Addison Wesley.
Noor, S. (2007). Operational scheduling of traditional and flexible manufacturing system using genetic algorithms, artificial neural networks and simulation. School of Engineering Design and Technology. Bradford, University of Bradford. **PhD**.
Noor, S. and Khan, M. K. (2007). Hybrid scheduling system for flow shop and job shop scheduling problem. 23rd International Conference of CAD/CAM, Robotics & Factories of the Future Conference. Bogota, Colombia.
Norman, B. A. and Bean, J. C. (1996) "A random keys GA for JSSP."
Noronha, S. and Sarma, V. (1991). "Knowledge-Based Approaches for Scheduling Problems: A Survey." IEEE Transactions on Knowledge and Data Engineering **3**(2): 160-171.
Nowicki, E. and Smutnicki, C. (1996). "A fast taboo search algorithm for the job shop problem." Manage. Sci. **42**(6): 797-813.
O'Kane, J. F. (2000). "A knowledge-based system for reactive scheduling decision-making in FMS." Journal of Intelligent Manufacturing **11**(5): 461-474.
Oltean, M. (2004). Searching for a practical evidence of the no free lunch theorems. Biologically Inspired Approaches to Advanced Information Technology. A. J. Ijspeert, M. Murata and N. Wakamiya. **3141:** 472-483.
Ono, I., Yamamura, M., Kobayashi, S. and Ieee (1996). A genetic algorithm for job-shop scheduling problems using job-based order crossover. New York, IEEE.

Orvosh, D. and Davis, L. (1994). Using a genetic algorithm to optimize problems with Feasibility constrains. Proceedings of the First IEEE Conference on Evolutionary Computation,, Orlando, FL, IEEE Press.

Pan, J. C. H. and Huang, H. C. (2009). "A hybrid genetic algorithm for no-wait job shop scheduling problems." Expert Systems with Applications **36**(3): 5800-5806.

Pan, Y. P. and Shi, L. Y. (2006). "Branch-and-bound algorithms for solving hard instances of the one-machine sequencing problem." European Journal of Operational Research **168**(3): 1030-1039.

Panwalkar, S. S. and Iskander, W. (1977). "A Survey of Scheduling Rules." Operations Research **25**(1): 45-61.

Park, B. J., Choi, H. R. and Kim, H. S. (2003). "A hybrid genetic algorithm for the job shop scheduling problems." Computers & Industrial Engineering **45**(4): 597-613.

Park, Y. S., Kim, S. Y. and Lee, Y. H. (2000). "Scheduling jobs on parallel machines applying neural network and heuristic rules." Computers & Industrial Engineering **38**(1): 189-202.

Pesch, E. (1993). Machine Learning by Schedule Decomposition, ,, Faculty of Economics and Business Administration, University of Limburg, Maastricht.

Petrovic, S., Fayad, C., Petrovic, D., Burke, E. and Kendall, G. (2008). "Fuzzy job shop scheduling with lot-sizing." Annals of Operations Research **159**(1): 275-292.

Pezzella, F., Morganti, G. and Ciaschetti, G. (2008). "A genetic algorithm for the Flexible Job-shop Scheduling Problem." Computers & Operations Research **35**(10): 3202-3212.

Pham, D. T. and Pham, P. T. N. (1999). "Artificial intelligence in engineering." International Journal of Machine Tools & Manufacture **39**(6): 937-949.

Pinedo, M., Chao, X. and Pinedo, M. L., Eds. (1998). Operations scheduling with applications in manufacturing and services. Singapore, McGraw-Hill.

Pogorzelski, H. A. (1965). "Reviewed work(s): Remarks on Nicod's Axiom and on "Generalizing Deductio." The Journal of Symbolic Logic **30**(3): 376-377.

Ponnambalam, S. G., Jawahar, N. and Aravindan, P. (1999). "A simulated annealing algorithm for job shop scheduling." Production Planning & Control **10**(8): 767-777.

Priore, P., De La Fuente, D., Gomez, A. and Puente, J. (2001). "A review of machine learning in dynamic scheduling of flexible manufacturing systems." Ai Edam-Artificial Intelligence for Engineering Design Analysis and Manufacturing **15**(3): 251-263.

Rachamadugu, R. and Schriber, T. J. (1990). Performance of Dispatching Rules Under Perfect Sequencing Flexibility. 1990 Winter Simulation Conference (Wsc 90), New Orleans, La, I E E E.

Rajendran, C. and Holthaus, O. (1999). "A comparative study of dispatching rules in dynamic flowshops and jobshops." European Journal of Operational Research **116**(1): 156-170.

Ramasesh, R. (1990). "Dynamic job shop scheduling: A survey of simulation research." Omega **18**(1): 43-57.

Restrepo, I. M. and Balakrishnan, S. (2008). "Fuzzy-based methodology for multi-objective scheduling in a robot-centered flexible manufacturing cell." Journal of Intelligent Manufacturing **19**(4): 421-432.

Rich, E. and Knight, K. (1991). Artificial Inetlligence, MGraw-Hill, Inc.

Rothlauf, F., Goldberg, D. E. and Heinzl, A. (2002). "Network random keys - A tree representation scheme for genetic and evolutionary algorithms." Evolutionary Computation **10**(1): 75-97.

Roy, B. and Sussmann, B. (1964). "Les Problems d'Ordonnancement avec Contraintes Disjonctives." Note D. S. no. 9 bis, SEMA, Paris, France.

Sabuncuoglu, I. and Touhami, S. (2002). "Simulation metamodelling with neural networks: an experimental investigation." International Journal of Production Research **40**(11): 2483-2505.

Sadeh, N. M. and Nakakuki, Y. (1996). "Focused simulated annealing search: An application to job shop scheduling." Annals of Operations Research **63**: 77-103.

Sakawa, M. and Kubota, R. (2000). "Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms." European Journal of Operational Research **120**(2): 393-407.

Samanlioglu, F., Ferrell, W. G. and Kurz, M. E. (2008). "A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem." Computers & Industrial Engineering **55**(2): 439-449.

Shi, G. (1997). "A genetic algorithm applied to a classic job-shop scheduling problem." International Journal of Systems Science **28**(1): 25-32.

Shugang, L., Zhiming, W. and Xiaohong, P. (2005). "Job shop scheduling in real-time cases." International Journal of Advanced Manufacturing Technology **26**(7-8): 870-875.

Sidney, J. B. (1983). "Sequencing and scheduling-an introduction to the mathematics of the job-shop, by Simon French, Wiley, 1982, 245 pp. Price: $49.95." Networks **13**(2): 310-311.

Słowiński, R. and Hapke, M. (2000). "Scheduling under Fuzziness." Physica-Verlag: 308.

Smith, W. E. (1956). "Various optimizers for single stage production." Naval Research Logistics Quarterly **3**: 59-66.

Snyder, L. V. and Daskin, M. S. (2006). "A random-key genetic algorithm for the generalized traveling salesman problem." European Journal of Operational Research **174**(1): 38-53.

Solimanpur, M., Vrat, P. and Shankar, R. (2004). "A neuro-tabu search heuristic for the flow shop scheduling problem." Computers & Operations Research **31**(13): 2151-2164.

Soyuer, H., Kocamaz, M. and Kazancoglu, Y. (2007). "Scheduling jobs through multiple parallel channels using an expert system." Production Planning & Control **18**(1): 35-43.

Srinivas, V. (1971). "Hybrid Algorithm for One Machine Sequencing Problem to Minize Total Tardiness." Naval Research Logistics Quarterly **18**(3): 317-.

Steinhofel, K., Albrecht, A. and Wong, C. K. (2003). "An experimental analysis of local minima to improve neighbourhood search." Computers & Operations Research **30**(14): 2157-2173.

Storer, R. H., Wu, S. D. and Vaccari, R. (1992). "New search spaces for sequencing problems with application to job shop scheduling." Management Science **38**(10): 1495-1509.

Taillard, E. (1993). "Benchmarks for basic job shop scheduling problems." European Journal of Operational Research **64**(2): 278-285.

Tamaki, H. and Nishikawa, Y. (1992). A Paralleled GA Based on A Neighborhood Model and Its Application to the Job Shop Scheduling.

Tariq, A. (2008). Operational design of a Cellular Manufacturing System (CMS). Deptt. of Industrial and Manufacturing Islamabad, NUST, Pakistan. **PhD**.

Teti, R. and Kumara, S. R. T. (1997). "Intelligent computing methods for manufacturing systems." Cirp Annals 1997 - Manufacturing Technology, Volume 46/2/1997: 629-652.

Thornblad, K. and Patriksson, M. (2011). "A note on the complexity of flow-shop scheduling with deteriorating jobs." Discrete Applied Mathematics **159**(4): 251-253.

Timkovskiy, V. G. (1985). "On The Complexity of Scheduling An Arbitaray System " Soviet Journal of Computer and Systems Sciences **23**(5): 46-52.

Tseng, L. Y. and Lin, Y. T. (2009). "A hybrid genetic local search algorithm for the permutation flowshop scheduling problem." European Journal of Operational Research **198**(1): 84-92.

Vanlaarhoven, P. J. M., Aarts, E. H. L. and Lenstra, J. K. (1992). "Job Shop Scheduling Problem by Simulated Annealing." Operations Research **40**(1): 113-125.

Varadharajan, T. K. and Rajendran, C. (2005). "A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs." European Journal of Operational Research **167**(3): 772-795.

Varela, R., Vela, C. R., Puente, J. P. and Gomez, A. (2003). "A Knowledge Based Evolutionary Strategy for Scheduling Problems with Bottlenecks." European Journal of Operational Research **145**: 57-71.

Vela, C. R., Varela, R. and Gonzalez, M. A. (2010). "Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times." Journal of Heuristics **16**(2): 139-165.

Wang, L. and Zheng, D.-Z. (2001). "An effective hybrid optimization strategy for job-shop scheduling problems." Computers & Operations Research **28**(6): 585-596.

Wang, W. and Brunn, P. (1995). Production Scheduling and Neural Networks. Operations Research Proceedings, Germany, Springer.

Weckman, G. R., Ganduri, C. V. and Koonce, D. A. (2008). "A neural network job-shop scheduler." Journal of Intelligent Manufacturing **19**(2): 191-201.

White, J. A. (1987). Production Hand Book, John Wiley and Sons Inc.

Wu, T. H., Chang, C. C. and Chung, S. H. (2008). "A simulated annealing algorithm for manufacturing cell formation problems." Expert Systems with Applications **34**(3): 1609-1617.

Wu, Z. M. and Zhao, C. W. (2000). "Genetic algorithm approach to job shop scheduling and its use in real-time cases." International Journal of Computer Integrated Manufacturing **13**(5): 422-429.

Xu, X. A., Xu, Z. H. and Gu, X. S. (2011). "An asynchronous genetic local search algorithm for the permutation flowshop scheduling problem with total flowtime minimization." Expert Systems with Applications **38**(7): 7970-7979.

Yamada, T. and Nakano, R. (1991). Conventional Genetic Algorithm for JSSP. 4th Internation Conference on Genetic Algorithms and their Applications, , San Diego, USA.

Yamada, T. and Nakano, R. (1992). "A genetic algorithm application to large scale job shop problems." Parallel Problem Solving from Nature, 2: 281-290.

Yamada, T., Rosen, B. E., Nakano, R. and Ieee (1994). A Simulated Annealing Approach to Job Shop Scheduling Using Critical Block Transition Operators. New York, IEEE.

Yang, S. X. and Wang, D. W. (2000). "Constraint satisfaction adaptive neural network and heuristics combined approaches for generalized job-shop scheduling." Ieee Transactions on Neural Networks **11**(2): 474-486.

Yang, S. X. and Wang, D. W. (2001). "A new adaptive neural network and heuristics hybrid approach for job-shop scheduling." Computers & Operations Research **28**(10): 955-971.

Yang, S. X., Wang, D. W., Chai, T. Y. and Kendall, G. (2010). "An improved constraint satisfaction adaptive neural network for job-shop scheduling." Journal of Scheduling **13**(1): 17-38.

Yeh, J. Y. and Lin, W. S. (2007). "Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department." Expert Systems with Applications **32**(4): 1073-1083.

Yusof, R., Khalid, M. and San, T. C. (2010). "Solving Industrial Based Job-Shop Scheduling Problem by Distributed Micro-Genetic Algorithm with Local Search." Knowledge-Based and Intelligent Information and Engineering Systems, Pt I **6276**: 69-79.

Zadeh, L. A. (1965). "Fuzzy sets." Information and Control **8**(3): 338-353.

Zattar, I. C., Ferreira, J. C. E. and Botura, P. E. D. (2008). Reducing the Standard Deviation When Integrating Process Planning and Production Scheduling Through the Use of Expert Systems in an Agent-based Environment. Collaborative Productive and Service Life Cycle Management for a Sustainable World. R. Curran, S. Y. Chou and A. Trappey. Berlin, Springer-Verlag Berlin**:** 305-312.

Zhang, R. and Wu, C. (2010). "A hybrid approach to large-scale job shop scheduling." Applied Intelligence **32**(1): 47-59.

Zhang, R. and Wu, C. (2010). "A hybrid immune simulated annealing algorithm for the job shop scheduling problem." Applied Soft Computing **10**(1): 79-89.

Zhang, Y., Li, X. P. and Wang, Q. (2009). "Hybrid genetic algorithm for permutation flowshop scheduling problems with total flowtime minimization." European Journal of Operational Research **196**(3): 869-876.

Zhao, F. Q., Hong, Y., Yu, D. M. and Yang, Y. H. (2005). A hybrid approach based on artificial neural network and genetic algorithm for job-shop scheduling problem. International Conference on Neural Networks and Brain (ICNN&B 2005), Beijing, PEOPLES R CHINA, IEEE.

Zhou, H., Feng, Y. C. and Han, L. M. (2001). "The hybrid heuristic genetic algorithm for job shop scheduling." Computers & Industrial Engineering **40**(3): 191-200.

Zhou, H. and Shi, R. F. (2004). "A multi-objective optimization method based on genetic algorithm and local search with applications to scheduling." Management Sciences and Global Strategies in the 21st Century, Vols 1 and 2: 177-183.

APPENDIX A: SET OF BENCHMARK JSSP

This appendix contains a set of selected 50 JSSP test instances.
(Note: If the table splits, continue to next page).

These instances are contributed to the OR-Library by
Dirk C. Mattfeld (email dirk@uni-bremen.de) and
Rob J.M. Vaessens (email robv@win.tue.nl).

o **abz5-abz9** are from
J. Adams, E. Balas and D. Zawack (1988),
The shifting bottleneck procedure for job shop scheduling,
Management Science 34, 391-401.

o **ft06, ft10, and ft20** are from
H. Fisher, G.L. Thompson (1963),
Probabilistic learning combinations of local job-shop scheduling rules,
J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling,
Prentice Hall, Englewood Cliffs, New Jersey, 225-251.

o **la01-la40** are from
S. Lawrence (1984),
Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement),
Graduate School of Industrial Administration,
Carnegie-Mellon University, Pittsburgh, Pennsylvania.

o **orb01-orb05** are from
D. Applegate, W. Cook (1991),
A computational study of the job-shop scheduling instance,
ORSA Journal on Computing 3, 149-156.
(they were generated in Bonn in 1986)

o TA01,TA02,TA11,TA12,TA13,TA16,TA31,and TA37 are from
E. D. Taillard, (1994)
Parallel taboo search techniques for the job shop scheduling problem,
ORSA Journal on Computing 6, 108 117.
Results also reported in:
E. D. Taillard (1993),
Benchmarks for basic scheduling problems,
EJOR 64, 278-285.

++
++++++++++++++

Each instance consists of a line of description, a line containing the
number of jobs and the number of machines, and then one line for each job,
listing the machine number and processing time for each step of the job.
The machines are numbered starting with 0.

++++++++++++++++++++++++++++

instance abz5

+++++++++++++++++++++++++++++
Adams, Balas, and Zawack 10x10 instance (Table 1, instance 5)
10 10
4 88 8 68 6 94 5 99 1 67 2 89 9 77 7 99 0 86 3 92
5 72 3 50 6 69 4 75 2 94 8 66 0 92 1 82 7 94 9 63
9 83 8 61 0 83 1 65 6 64 5 85 7 78 4 85 2 55 3 77
7 94 2 68 1 61 4 99 3 54 6 75 5 66 0 76 9 63 8 67
3 69 4 88 9 82 8 95 0 99 2 67 6 95 5 68 7 67 1 86
1 99 4 81 5 64 6 66 8 80 2 80 7 69 9 62 3 79 0 88
7 50 1 86 4 97 3 96 0 95 8 97 2 66 5 99 6 52 9 71
4 98 6 73 3 82 2 51 1 71 5 94 7 85 0 62 8 95 9 79
0 94 6 71 3 81 7 85 1 66 2 90 4 76 5 58 8 93 9 97
3 50 0 59 1 82 8 67 7 56 9 96 6 58 4 81 5 59 2 96
+++++++++++++++++++++++++++++

instance abz6

+++++++++++++++++++++++++++++
Adams, and Zawack 10x10 instance (Table 1, instance 6)
10 10
7 62 8 24 5 25 3 84 4 47 6 38 2 82 0 93 9 24 1 66
5 47 2 97 8 92 9 22 1 93 4 29 7 56 3 80 0 78 6 67
1 45 7 46 6 22 2 26 9 38 0 69 4 40 3 33 8 75 5 96
4 85 8 76 5 68 9 88 3 36 6 75 2 56 1 35 0 77 7 85
8 60 9 20 7 25 3 63 4 81 0 52 1 30 5 98 6 54 2 86
3 87 9 73 5 51 2 95 4 65 1 86 6 22 8 58 0 80 7 65
5 81 2 53 7 57 6 71 9 81 0 43 4 26 8 54 3 58 1 69
4 20 6 86 5 21 8 79 9 62 2 34 0 27 1 81 7 30 3 46
9 68 6 66 5 98 8 86 7 66 0 56 3 82 1 95 4 47 2 78
0 30 3 50 7 34 2 58 1 77 5 34 8 84 4 40 9 46 6 44
+++++++++++++++++++++++++++++

instance abz7

+++++++++++++++++++++++++++++
Adams, Balas, and Zawack 15 x 20 instance (Table 1, instance 7)
20 15
2 24 3 12 9 17 4 27 0 21 6 25 8 27 7 26 1 30 5 31 11 18 14 16 13 39 10 19 12 26
6 30 3 15 12 20 11 19 1 24 13 15 10 28 2 36 5 26 7 15 0 11 8 23 14 20 9 26 4 28
6 35 0 22 13 23 7 32 2 20 3 12 12 19 10 23 9 17 1 14 5 16 11 29 8 16 4 22 14 22
9 20 6 29 1 19 7 14 12 33 4 30 0 32 5 21 11 29 10 24 14 25 2 29 3 13 8 20 13 18
11 23 13 20 1 28 6 32 7 16 5 18 8 24 9 23 3 24 10 34 2 24 0 24 14 28 12 15 4 18
8 24 11 19 14 21 1 33 7 34 6 35 5 40 10 36 3 23 2 26 4 15 9 28 13 38 12 13 0 25
13 27 3 30 6 21 8 19 12 12 4 27 2 39 9 13 14 12 5 36 10 21 11 17 1 29 0 17 7 33
5 27 4 19 6 29 9 20 3 21 10 40 8 14 14 39 13 39 2 27 1 36 12 12 11 37 7 22 0 13
13 32 11 29 8 24 3 27 5 40 4 21 9 26 0 27 14 27 6 16 2 21 10 13 7 28 12 28 1 32
12 35 1 11 5 39 14 18 7 23 0 34 3 24 13 11 8 30 11 31 4 15 10 15 2 28 9 26 6 33
10 28 5 37 12 29 1 31 7 25 8 13 14 14 4 20 3 27 9 25 13 31 11 14 6 25 2 39 0 36
0 22 11 25 5 28 13 35 4 31 8 21 9 20 14 19 2 29 7 32 10 18 1 18 3 11 12 17 6 15
12 39 5 32 2 36 8 14 3 28 13 37 0 38 6 20 7 19 11 12 14 22 1 36 4 15 9 32 10 16
8 28 1 29 14 40 12 23 4 34 5 33 6 27 10 17 0 20 7 28 11 21 2 21 13 20 9 33 3 27
9 21 14 34 3 30 12 38 0 11 11 16 2 14 5 14 1 34 8 33 4 23 13 40 10 12 6 23 7 27
9 13 14 40 7 36 4 17 0 13 5 33 8 25 13 24 10 23 3 36 2 29 1 18 11 13 6 33 12 13
3 25 5 15 2 28 12 40 7 39 1 31 8 35 6 31 11 36 4 12 10 33 14 19 9 16 13 27 0 21
12 22 10 14 0 12 2 20 5 12 1 18 11 17 8 39 14 31 3 31 7 32 9 20 13 29 4 13 6 26
5 18 10 30 7 38 14 22 13 15 11 20 9 16 3 17 1 12 2 13 12 40 6 17 8 30 4 38 0 13
9 31 8 39 12 27 1 14 5 33 3 31 11 22 13 36 0 16 7 11 14 14 4 29 6 28 2 22 10 17
+++++++++++++++++++++++++++++

instance abz8

+++++++++++++++++++++++++++++
Adams, Balas, and Zawack 15 x 20 instance (Table 1, instance 8)
20 15
0 19 9 33 2 32 13 18 10 39 8 34 6 25 4 36 11 40 12 33 1 31 14 30 3 34 5 26 7 13
9 11 10 22 14 19 5 12 4 25 6 38 0 29 7 39 13 19 11 22 1 23 3 20 2 40 12 19 8 26
3 25 8 17 11 24 13 40 10 32 14 16 5 39 9 19 0 24 1 39 4 17 2 35 7 38 6 20 12 31
14 22 3 36 2 34 12 17 4 30 13 12 1 13 6 25 9 12 7 18 10 31 0 39 5 40 8 26 11 37
12 32 14 15 1 35 7 13 8 32 11 23 6 22 4 21 0 38 2 38 3 40 10 31 5 11 13 37 9 16
10 23 12 38 8 11 14 27 9 11 6 25 5 14 4 12 2 27 11 26 7 29 3 28 13 21 0 20 1 30
6 39 8 38 0 15 12 27 10 22 9 27 2 32 4 40 3 12 13 20 14 21 11 22 5 17 7 38 1 27
11 11 13 24 10 38 8 15 9 19 14 13 5 30 0 26 2 29 6 33 12 21 1 15 3 21 4 28 7 33
8 20 6 17 5 26 3 34 9 23 0 16 2 18 4 35 12 24 10 16 11 26 7 12 14 13 13 27 1 19
1 18 7 37 14 27 9 40 5 40 6 17 8 22 3 17 10 30 0 38 4 21 12 32 11 24 13 24 2 30
11 19 0 22 13 36 6 18 5 22 3 17 14 35 10 34 7 23 8 19 2 29 1 22 12 17 4 33 9 39
6 32 3 22 12 24 5 13 4 13 1 11 0 11 13 25 8 13 2 15 10 33 11 17 14 16 9 38 7 24
14 16 13 16 1 37 8 25 2 26 3 11 9 34 4 14 0 20 6 36 12 12 5 29 10 25 7 32 11 12
8 20 10 24 11 27 9 38 5 34 12 39 7 33 4 37 2 31 13 15 14 34 3 33 6 26 1 36 0 14
8 31 0 17 9 13 1 21 10 17 7 19 13 14 3 40 5 32 11 25 2 34 14 23 6 13 12 40 4 26
8 38 12 17 3 14 13 17 4 12 1 35 6 35 0 19 10 36 7 19 9 29 2 31 5 26 11 35 14 37
14 20 3 16 0 33 10 14 5 27 7 31 8 16 6 31 12 28 9 37 4 37 2 29 11 38 1 30 13 36
11 18 3 37 14 16 6 15 8 14 12 11 13 32 5 12 1 11 10 29 7 19 4 12 9 18 2 26 0 39
11 11 2 11 12 22 9 35 14 20 7 31 4 19 3 39 5 28 6 33 10 34 1 38 0 20 13 17 8 28
2 12 12 25 5 23 8 21 6 27 9 30 14 23 11 39 3 26 13 34 7 17 1 24 4 12 0 19 10 36
+++++++++++++++++++++++++++++

instance abz9

+++++++++++++++++++++++++++++
Adams, Balas, and Zawack 15 x 20 instance (Table 1, instance 9)
20 15
6 14 5 21 8 13 4 11 1 11 14 35 13 20 11 17 10 18 12 11 2 23 3 13 0 15 7 11 9 35
1 35 5 31 0 13 3 26 6 14 9 17 7 38 12 20 10 19 13 12 8 16 4 34 11 15 14 12 2 14
0 30 4 35 2 40 10 35 6 30 14 23 8 29 13 37 7 38 3 40 9 26 12 11 1 40 11 36 5 17
7 40 5 18 4 12 8 23 0 23 9 14 13 16 12 14 10 23 3 12 6 16 14 32 1 40 11 25 2 29
2 35 3 15 12 31 11 28 6 32 4 30 10 27 7 29 0 38 13 11 1 23 14 17 5 27 9 37 8 29
5 33 3 33 6 19 12 40 10 19 0 33 13 26 2 31 11 28 7 36 4 38 1 21 14 25 9 40 8 35
13 25 0 32 11 33 12 18 4 32 6 28 5 15 3 35 9 14 2 34 7 23 10 32 1 17 14 26 8 19
2 16 12 33 9 34 11 30 13 40 8 12 14 26 5 26 6 15 3 21 1 40 4 32 0 14 7 30 10 35
2 17 10 16 14 20 6 24 8 26 3 36 12 22 0 14 13 11 9 20 7 23 1 29 11 23 4 15 5 40
4 27 9 37 3 40 11 14 13 25 7 30 0 34 2 11 5 15 12 32 1 36 10 12 14 28 8 31 6 23
13 25 0 22 3 27 8 14 5 25 6 20 14 18 7 14 1 19 2 17 4 27 9 22 12 22 11 27 10 21
14 34 10 15 0 22 3 29 13 34 6 40 7 17 2 32 12 20 5 39 4 31 11 16 1 37 8 33 9 13
6 12 12 27 4 17 2 24 8 11 5 19 14 11 3 17 9 25 1 11 11 31 13 33 7 31 10 12 0 22
5 22 14 15 0 16 8 32 7 20 4 22 9 11 13 19 1 30 12 33 6 29 11 18 3 34 10 32 2 18
5 27 3 26 10 28 6 37 4 18 12 12 11 11 13 26 7 27 9 40 14 19 1 24 2 18 0 12 8 34
8 15 5 28 9 25 6 32 1 13 7 38 11 11 2 34 4 25 0 20 10 32 3 23 12 14 14 16 13 20
1 15 4 13 8 37 3 14 10 22 5 24 12 26 7 22 9 34 14 22 11 19 13 32 0 29 2 13 6 35
7 36 5 33 13 28 9 20 10 30 4 33 14 29 0 34 3 22 11 12 6 30 8 12 1 35 2 13 12 35
14 26 11 31 5 35 2 38 13 19 10 35 4 27 8 29 3 39 9 13 6 14 7 26 0 17 1 22 12 15
1 36 7 34 11 33 8 17 14 38 6 39 5 16 3 27 13 29 2 16 0 16 4 19 9 40 12 35 10 39
+++++++++++++++++++++++++++++

instance ft06

+++++++++++++++++++++++++++++
Fisher and Thompson 6x6 instance, alternate name (mt06)
6 6
2 1 0 3 1 6 3 7 5 3 4 6
1 8 2 5 4 10 5 10 0 10 3 4
2 5 3 4 5 8 0 9 1 1 4 7
1 5 0 5 2 5 3 3 4 8 5 9
2 9 1 3 4 5 5 4 0 3 3 1
1 3 3 3 5 9 0 10 4 4 2 1
+++++++++++++++++++++++++++++

instance ft10

+++++++++++++++++++++++++++++
Fisher and Thompson 10x10 instance, alternate name (mt10)
10 10
0 29 1 78 2 9 3 36 4 49 5 11 6 62 7 56 8 44 9 21
0 43 2 90 4 75 9 11 3 69 1 28 6 46 5 46 7 72 8 30
1 91 0 85 3 39 2 74 8 90 5 10 7 12 6 89 9 45 4 33
1 81 2 95 0 71 4 99 6 9 8 52 7 85 3 98 9 22 5 43
2 14 0 6 1 22 5 61 3 26 4 69 8 21 7 49 9 72 6 53
2 84 1 2 5 52 3 95 8 48 9 72 0 47 6 65 4 6 7 25
1 46 0 37 3 61 2 13 6 32 5 21 9 32 8 89 7 30 4 55
2 31 0 86 1 46 5 74 4 32 6 88 8 19 9 48 7 36 3 79
0 76 1 69 3 76 5 51 2 85 9 11 6 40 7 89 4 26 8 74
1 85 0 13 2 61 6 7 8 64 9 76 5 47 3 52 4 90 7 45

++++++++++++++++++++++++++++

instance ft20

++++++++++++++++++++++++++++
Fisher and Thompson 20x5 instance, alternate name (mt20)
20 5
0 29 1 9 2 49 3 62 4 44
0 43 1 75 3 69 2 46 4 72
1 91 0 39 2 90 4 12 3 45
1 81 0 71 4 9 2 85 3 22
2 14 1 22 0 26 3 21 4 72
2 84 1 52 4 48 0 47 3 6
1 46 0 61 2 32 3 32 4 30
2 31 1 46 0 32 3 19 4 36
0 76 3 76 2 85 1 40 4 26
1 85 2 61 0 64 3 47 4 90
1 78 3 36 0 11 4 56 2 21
2 90 0 11 1 28 3 46 4 30
0 85 2 74 1 10 3 89 4 33
2 95 0 99 1 52 3 98 4 43
0 6 1 61 4 69 2 49 3 53
1 2 0 95 3 72 4 65 2 25
0 37 2 13 1 21 3 89 4 55
0 86 1 74 4 88 2 48 3 79
1 69 2 51 0 11 3 89 4 74
0 13 1 7 2 76 3 52 4 45
++++++++++++++++++++++++++++

instance la01

++++++++++++++++++++++++++++
Lawrence 10x5 instance (Table 3, instance 1); also called (setf1) or (F1)
10 5
1 21 0 53 4 95 3 55 2 34
0 21 3 52 4 16 2 26 1 71
3 39 4 98 1 42 2 31 0 12
1 77 0 55 4 79 2 66 3 77
0 83 3 34 2 64 1 19 4 37
1 54 2 43 4 79 0 92 3 62
3 69 4 77 1 87 2 87 0 93
2 38 0 60 1 41 3 24 4 83
3 17 1 49 4 25 0 44 2 98
4 77 3 79 2 43 1 75 0 96
++++++++++++++++++++++++++++

instance la02

++++++++++++++++++++++++++++
Lawrence 10x5 instance (Table 3, instance 2); also called (setf2) or (F2)
10 5
0 20 3 87 1 31 4 76 2 17
4 25 2 32 0 24 1 18 3 81
1 72 2 23 4 28 0 58 3 99
2 86 1 76 4 97 0 45 3 90
4 27 0 42 3 48 2 17 1 46
1 67 0 98 4 48 3 27 2 62
4 28 1 12 3 19 0 80 2 50
1 63 0 94 2 98 3 50 4 80
4 14 0 75 2 50 1 41 3 55
4 72 2 18 1 37 3 79 0 61
++++++++++++++++++++++++++++

instance la03

++++++++++++++++++++++++++++
Lawrence 10x5 instance (Table 3, instance 3); also called (setf3) or (F3)
10 5
1 23 2 45 0 82 4 84 3 38
2 21 1 29 0 18 4 41 3 50
2 38 3 54 4 16 0 52 1 52
4 37 0 54 2 74 1 62 3 57
4 57 0 81 1 61 3 68 2 30
4 81 0 79 1 89 2 89 3 11
3 33 2 20 0 91 4 20 1 66
4 24 1 84 0 32 2 55 3 8
4 56 0 7 3 54 2 64 1 39
4 40 1 83 0 19 2 8 3 7
++++++++++++++++++++++++++++

instance la04

++++++++++++++++++++++++++++
Lawrence 10x5 instance (Table 3, instance 4); also called (setf4) or (F4)
10 5
0 12 2 94 3 92 4 91 1 7
1 19 3 11 4 66 2 21 0 87
1 14 0 75 3 13 4 16 2 20
2 95 4 66 0 7 3 7 1 77
1 45 3 6 4 89 0 15 2 34
3 77 2 20 0 76 4 88 1 53
2 74 1 88 0 52 3 27 4 9
1 88 3 69 0 62 4 98 2 52
2 61 4 9 0 62 1 52 3 90
2 54 4 5 3 59 1 15 0 88
++++++++++++++++++++++++++++

instance la05

++++++++++++++++++++++++++++
Lawrence 10x5 instance (Table 3, instance 5); also called (setf5) or (F5)
10 5
1 72 0 87 4 95 2 66 3 60
4 5 3 35 0 48 2 39 1 54
1 46 3 20 2 21 0 97 4 55
0 59 3 19 4 46 1 34 2 37
4 23 2 73 3 25 1 24 0 28
3 28 0 45 4 5 1 78 2 83
0 53 3 71 1 37 4 29 2 12
4 12 2 87 3 33 1 55 0 38
2 49 3 83 1 40 0 48 4 7
2 65 3 17 0 90 4 27 1 23
++++++++++++++++++++++++++++

instance la06

++++++++++++++++++++++++++++
Lawrence 15x5 instance (Table 4, instance 1); also called (setg1) or (G1)
15 5
1 21 2 34 4 95 0 53 3 55
3 52 4 16 1 71 2 26 0 21
2 31 0 12 1 42 3 39 4 98
3 77 1 77 4 79 0 55 2 66
4 37 3 34 2 64 1 19 0 83
2 43 1 54 0 92 3 62 4 79
0 93 3 69 1 87 4 77 2 87
0 60 1 41 2 38 4 83 3 24
2 98 3 17 4 25 0 44 1 49
0 96 4 77 3 79 1 75 2 43
4 28 2 35 0 95 3 76 1 7
0 61 4 10 2 95 1 9 3 35
4 59 3 16 1 91 2 59 0 46
4 43 1 52 0 28 2 27 3 50
0 87 1 45 2 39 4 9 3 41
++++++++++++++++++++++++++++

instance la07

++++++++++++++++++++++++++++
Lawrence 15x5 instance (Table 4, instance 2); also called (setg2) or (G2)
15 5
0 47 4 57 1 71 3 96 2 14
0 75 1 60 4 22 3 79 2 65
3 32 0 33 2 69 1 31 4 58
0 44 1 34 4 51 3 58 2 47
3 29 1 44 0 62 2 17 4 8
1 15 2 40 0 97 4 38 3 66
2 58 1 39 0 57 4 20 3 50
2 57 3 32 4 87 0 63 1 21
4 56 0 84 2 90 1 85 3 61
4 15 0 20 1 67 3 30 2 70
4 84 0 82 1 23 2 45 3 38
3 50 2 21 0 18 4 41 1 29
4 16 1 52 0 52 2 38 3 54
4 37 0 54 3 57 2 74 1 62
4 57 1 61 0 81 2 30 3 68
++++++++++++++++++++++++++++

instance la08

++++++++++++++++++++++++++++
Lawrence 15x5 instance (Table 4, instance 3); also called (setg3) or (G3)
15 5
3 92 2 94 0 12 4 91 1 7
2 21 1 19 0 87 3 11 4 66
1 14 3 13 0 75 4 16 2 20
2 95 4 66 0 7 1 77 3 7
2 34 4 89 3 6 1 45 0 15
4 88 3 77 2 20 1 53 0 76
4 9 3 27 0 52 1 88 2 74
3 69 2 52 0 62 1 88 4 98
3 90 0 62 4 9 2 61 1 52
4 5 2 54 3 59 0 88 1 15
0 41 1 50 4 78 3 53 2 23
0 38 4 72 2 91 3 68 1 71
0 45 3 95 4 52 2 25 1 6
3 30 1 66 0 23 4 36 2 17
2 95 0 71 3 76 1 8 4 88
++++++++++++++++++++++++++++

instance la09

++++++++++++++++++++++++++++
Lawrence 15x5 instance (Table 4, instance 4); also called (setg4) or (G4)
15 5
1 66 3 85 2 84 0 62 4 19
3 59 1 64 2 46 4 13 0 25
4 88 3 80 1 73 2 53 0 41
0 14 1 67 2 57 3 74 4 47
0 84 4 64 2 41 3 84 1 78
0 63 3 28 1 46 2 26 4 52
3 10 2 17 4 73 1 11 0 64
2 67 1 97 3 95 4 38 0 85
2 95 4 46 0 59 1 65 3 93
2 43 4 85 3 32 1 85 0 60
4 49 3 41 2 61 0 66 1 90
1 17 0 23 3 70 4 99 2 49
4 40 3 73 0 73 1 98 2 68
3 57 1 9 2 7 0 13 4 98
0 37 1 85 2 17 4 79 3 41
++++++++++++++++++++++++++++

instance la10

++++++++++++++++++++++++++++
Lawrence 15x5 instance (Table 4, instance 5); also called (setg5) or (G5)
15 5
1 58 2 44 3 5 0 9 4 58
1 89 0 97 4 96 3 77 2 84
0 77 1 87 2 81 4 39 3 85
3 57 1 21 2 31 0 15 4 73
2 48 0 40 1 49 3 70 4 71
3 34 4 82 2 80 0 10 1 22
1 91 4 75 0 55 2 17 3 7
2 62 3 47 1 72 4 35 0 11
0 64 3 75 4 50 1 90 2 94
2 67 4 20 3 15 0 12 1 71
0 52 4 93 3 68 2 29 1 57
2 70 0 58 1 93 4 7 3 77
3 27 2 82 1 63 4 6 0 95
1 87 2 56 4 36 0 26 3 48
3 76 2 36 0 36 4 15 1 8
++++++++++++++++++++++++++++

instance la11

++++++++++++++++++++++++++++
Lawrence 20x5 instance (Table 5, instance 1); also called (seth1) or H1
20 5
2 34 1 21 0 53 3 55 4 95
0 21 3 52 1 71 4 16 2 26
0 12 1 42 2 31 4 98 3 39
2 66 3 77 4 79 0 55 1 77
0 83 4 37 3 34 1 19 2 64
4 79 2 43 0 92 3 62 1 54
0 93 4 77 2 87 1 87 3 69
4 83 3 24 1 41 2 38 0 60
4 25 1 49 0 44 2 98 3 17
0 96 1 75 2 43 4 77 3 79
0 95 3 76 1 7 4 28 2 35
4 10 2 95 0 61 1 9 3 35
1 91 2 59 4 59 0 46 3 16
2 27 1 52 4 43 0 28 3 50
4 9 0 87 3 41 2 39 1 45
1 54 0 20 4 43 3 14 2 71
4 33 1 28 3 26 0 78 2 37
1 89 0 33 2 8 3 66 4 42
4 84 0 69 2 94 1 74 3 27
4 81 2 45 1 78 3 69 0 96
++++++++++++++++++++++++++++

instance la12

++++++++++++++++++++++++++++
Lawrence 20x5 instance (Table 5, instance 2); also called (seth2) or H2
20 5
1 23 0 82 4 84 2 45 3 38
3 50 4 41 1 29 0 18 2 21
4 16 3 54 1 52 2 38 0 52
1 62 3 57 4 37 2 74 0 54
3 68 1 61 2 30 0 81 4 57
1 89 2 89 3 11 0 79 4 81
1 66 0 91 3 33 4 20 2 20
3 8 4 24 2 55 0 32 1 84
0 7 2 64 1 39 4 56 3 54
0 19 4 40 3 7 2 8 1 83
0 63 2 64 3 91 4 40 1 6
1 42 3 61 4 15 2 98 0 74
1 80 0 26 3 75 4 6 2 87
2 39 4 22 0 75 3 24 1 44
1 15 3 79 4 8 0 12 2 20
3 26 2 43 0 80 4 22 1 61
2 62 1 36 0 63 3 96 4 40
1 33 3 18 0 22 4 5 2 10
2 64 4 64 0 89 1 96 3 95
2 18 4 23 3 15 1 38 0 8
++++++++++++++++++++++++++++

instance la13

++++++++++++++++++++++++++++
Lawrence 20x5 instance (Table 5, instance 3); also called (seth3) or (H3)
20 5
3 60 0 87 1 72 4 95 2 66
1 54 0 48 2 39 3 35 4 5
3 20 1 46 0 97 2 21 4 55
2 37 0 59 3 19 1 34 4 46
2 73 3 25 1 24 0 28 4 23
1 78 3 28 2 83 0 45 4 5
3 71 1 37 2 12 4 29 0 53
4 12 3 33 1 55 2 87 0 38
0 48 1 40 2 49 3 83 4 7
0 90 4 27 2 65 3 17 1 23
0 62 3 85 1 66 2 84 4 19
3 59 2 46 4 13 1 64 0 25
2 53 1 73 3 80 4 88 0 41
2 57 4 47 0 14 1 67 3 74
2 41 4 64 3 84 1 78 0 84
4 52 3 28 2 26 0 63 1 46
1 11 0 64 3 10 4 73 2 17
4 38 3 95 0 85 1 97 2 67
3 93 1 65 2 95 0 59 4 46
0 60 1 85 2 43 4 85 3 32
++++++++++++++++++++++++++++

instance la14

++++++++++++++++++++++++++++
Lawrence 20x5 instance (Table 5, instance 4); also called (seth4) or (H4)
20 5
3 5 4 58 2 44 0 9 1 58
1 89 4 96 0 97 2 84 3 77
2 81 3 85 1 87 4 39 0 77
0 15 3 57 4 73 1 21 2 31
2 48 4 71 3 70 0 40 1 49
0 10 4 82 3 34 2 80 1 22
2 17 0 55 1 91 4 75 3 7
3 47 2 62 1 72 4 35 0 11
1 90 2 94 4 50 0 64 3 75
3 15 2 67 0 12 4 20 1 71
4 93 2 29 0 52 1 57 3 68
3 77 1 93 0 58 2 70 4 7
1 63 3 27 0 95 4 6 2 82
4 36 0 26 3 48 2 56 1 87
2 36 1 8 4 15 3 76 0 36
4 78 1 84 3 41 0 30 2 76
1 78 0 75 4 88 3 13 2 81
0 54 4 40 2 13 1 82 3 29
1 26 4 82 0 52 3 6 2 6
3 54 1 64 0 54 2 32 4 88
++++++++++++++++++++++++++++

instance la15

++++++++++++++++++++++++++++
Lawrence 20x5 instance (Table 5, instance 5); also called (seth5) or (H5)
20 5
0 6 2 40 1 81 3 37 4 19
2 40 3 32 0 55 4 81 1 9
1 46 4 65 2 70 3 55 0 77
2 21 4 65 0 64 3 25 1 15
2 85 0 40 1 44 3 24 4 37
0 89 4 29 1 83 3 31 2 84
4 59 3 38 1 80 2 30 0 8
0 80 2 56 1 77 4 41 3 97
4 56 0 91 3 50 2 71 1 17
1 40 0 88 4 59 2 7 3 80
0 45 1 29 2 8 4 77 3 58
2 36 0 54 3 96 1 9 4 10
0 28 2 73 1 98 3 92 4 87
0 70 3 86 2 27 1 99 4 96
1 95 0 59 4 56 3 85 2 41
1 81 2 92 4 32 0 52 3 39
1 7 4 22 2 12 0 88 3 60
3 45 0 93 2 69 4 49 1 27
0 21 1 84 2 61 3 68 4 26
1 82 2 33 4 71 0 99 3 44
++++++++++++++++++++++++++++

instance la16

++++++++++++++++++++++++++++
Lawrence 10x10 instance (Table 6, instance 1); also called (seta1) or (A1)
10 10
1 21 6 71 9 16 8 52 7 26 2 34 0 53 4 21 3 55 5 95
4 55 2 31 5 98 9 79 0 12 7 66 1 42 8 77 6 77 3 39
3 34 2 64 8 62 1 19 4 92 9 79 7 43 6 54 0 83 5 37
1 87 3 69 2 87 7 38 8 24 9 83 6 41 0 93 5 77 4 60
2 98 0 44 5 25 6 75 7 43 1 49 4 96 9 77 3 17 8 79
2 35 3 76 5 28 9 10 4 61 6 9 0 95 8 35 1 7 7 95
3 16 2 59 0 46 1 91 9 43 8 50 6 52 5 59 4 28 7 27
1 45 0 87 3 41 4 20 6 54 9 43 8 14 5 9 2 39 7 71
4 33 2 37 8 66 5 33 3 26 7 8 1 28 6 89 9 42 0 78
8 69 9 81 2 94 4 96 3 27 0 69 7 45 6 78 1 74 5 84
++++++++++++++++++++++++++++

instance la17

++++++++++++++++++++++++++++
Lawrence 10x10 instance (Table 6, instance 2); also called (seta2) or (A2)
10 10
4 18 7 21 9 41 2 45 3 38 8 50 5 84 6 29 1 23 0 82
8 57 5 16 1 52 7 74 2 38 3 54 6 62 9 37 4 54 0 52
2 30 4 79 3 68 1 61 8 11 6 89 7 89 0 81 9 81 5 57
0 91 8 8 3 33 7 55 5 20 2 20 4 32 6 84 1 66 9 24
9 40 0 7 4 19 8 7 6 83 2 64 5 56 3 54 7 8 1 39
3 91 2 64 5 40 0 63 7 98 4 74 8 61 1 6 6 42 9 15
1 80 7 39 8 24 3 75 4 75 5 6 6 44 0 26 2 87 9 22
1 15 7 43 2 20 0 12 8 26 6 61 3 79 9 22 5 8 4 80
2 62 3 96 4 22 9 5 0 63 6 33 7 10 8 18 1 36 5 40
1 96 0 89 5 64 3 95 9 23 7 18 8 15 2 64 6 38 4 8
++++++++++++++++++++++++++++

instance la18

++++++++++++++++++++++++++++
Lawrence 10x10 instance (Table 6, instance 3); also called (seta3) or (A3)
10 10
6 54 0 87 4 48 3 60 7 39 8 35 1 72 5 95 2 66 9 5
3 20 9 46 6 34 5 55 0 97 8 19 4 59 2 21 7 37 1 46
4 45 1 24 8 28 0 28 7 83 6 78 5 23 3 25 9 5 2 73
9 12 1 37 4 38 3 71 8 33 2 12 6 55 0 53 7 87 5 29
3 83 2 49 6 23 9 27 7 65 0 48 4 90 5 7 1 40 8 17
1 66 4 25 0 62 2 84 9 13 6 64 7 46 8 59 5 19 3 85

1 73 3 80 0 41 2 53 9 47 7 57 8 74 4 14 6 67 5 88
5 64 3 84 6 46 1 78 0 84 7 26 8 28 9 52 2 41 4 63
1 11 0 64 7 67 4 85 3 10 5 73 9 38 8 95 6 97 2 17
4 60 8 32 2 95 3 93 1 65 6 85 7 43 9 85 5 46 0 59
+++++++++++++++++++++++++++++

instance la19

+++++++++++++++++++++++++++++
Lawrence 10x10 instance (Table 6, instance 4); also called (seta4) or (A4)
10 10
2 44 3 5 5 58 4 97 0 9 7 84 8 77 9 96 1 58 6 89
4 15 7 31 1 87 8 57 0 77 3 85 2 81 5 39 9 73 6 21
9 82 6 22 4 10 3 70 1 49 0 40 8 34 2 48 7 80 5 71
1 91 2 17 7 62 5 75 8 47 4 11 3 7 6 72 9 35 0 55
6 71 1 90 3 75 0 64 2 94 8 15 4 12 7 67 9 20 5 50
7 70 5 93 8 77 2 29 4 58 6 93 3 68 1 57 9 7 0 52
6 87 1 63 4 26 5 6 2 82 3 27 7 56 8 48 9 36 0 95
0 36 5 15 8 41 9 78 3 76 6 84 4 30 7 76 2 36 1 8
5 88 2 81 3 13 6 82 4 54 7 13 8 29 9 40 1 78 0 75
9 88 4 54 6 64 7 32 0 52 2 6 8 54 5 82 3 6 1 26
+++++++++++++++++++++++++++++

instance la20

+++++++++++++++++++++++++++++
Lawrence 10x10 instance (Table 6, instance 5); also called (seta5) or (A5)
10 10
6 9 1 81 4 55 2 40 8 32 3 37 0 6 5 19 9 81 7 40
7 21 2 70 9 65 4 64 1 46 5 65 8 25 0 77 3 55 6 15
2 85 5 37 0 40 3 24 1 44 6 83 4 89 8 31 7 84 9 29
4 80 6 77 7 56 0 8 2 30 5 59 3 38 1 80 9 41 8 97
0 91 6 40 4 88 1 17 2 71 3 50 9 59 8 80 5 56 7 7
2 8 6 9 3 58 5 77 1 29 8 96 0 45 9 10 4 54 7 36
4 70 3 92 1 98 5 87 6 99 7 27 8 86 9 96 0 28 2 73
1 95 7 92 3 85 4 52 6 81 9 32 8 39 0 59 2 41 5 56
3 60 8 45 0 88 2 12 1 7 5 22 4 93 9 49 7 69 6 27
0 21 2 61 3 68 5 26 6 82 9 71 8 44 4 99 7 33 1 84
+++++++++++++++++++++++++++++

instance la21

+++++++++++++++++++++++++++++
Lawrence 15x10 instance (Table 7, instance 1); also called (setb1) or (B1)
15 10
2 34 3 55 5 95 9 16 4 21 6 71 0 53 8 52 1 21 7 26
3 39 2 31 0 12 1 42 9 79 8 77 6 77 5 98 4 55 7 66
1 19 0 83 3 34 4 92 6 54 9 79 8 62 5 37 2 64 7 43
4 60 2 87 8 24 5 77 3 69 7 38 1 87 6 41 9 83 0 93
8 79 9 77 2 98 4 96 3 17 0 44 7 43 6 75 1 49 5 25
8 35 7 95 6 9 9 10 2 35 1 7 5 28 4 61 0 95 3 76
4 28 5 59 3 16 9 43 0 46 8 50 6 52 7 27 2 59 1 91
5 9 4 20 2 39 6 54 1 45 7 71 0 87 3 41 9 43 8 14
1 28 5 33 0 78 3 26 2 37 7 8 8 66 6 89 9 42 4 33
2 94 5 84 6 78 9 81 1 74 3 27 8 69 0 69 7 45 4 96
1 31 4 24 0 20 2 17 9 25 8 81 5 76 3 87 7 32 6 18
5 28 9 97 0 58 4 45 6 76 3 99 2 23 1 72 8 90 7 86
5 27 9 48 8 27 7 62 4 98 6 67 3 48 0 42 1 46 2 17
1 12 8 50 0 80 2 50 9 80 3 19 5 28 6 63 4 94 7 98
4 61 3 55 6 37 5 14 2 50 8 79 1 41 9 72 7 18 0 75
+++++++++++++++++++++++++++++

instance la22

+++++++++++++++++++++++++++++
Lawrence 15x10 instance (Table 7, instance 2); also called (setb2) or (B2)
15 10
9 66 5 91 4 87 2 94 7 21 3 92 1 7 0 12 8 11 6 19
3 13 2 20 4 7 1 14 9 66 0 75 6 77 5 16 7 95 8 7
8 77 7 20 2 34 0 15 9 88 5 89 6 53 3 6 1 45 4 76
3 27 2 74 6 88 4 62 7 52 8 69 5 9 9 98 0 52 1 88
4 88 6 15 1 52 2 61 7 54 0 62 8 59 5 9 3 90 9 5
6 71 0 41 4 38 3 53 7 91 8 68 1 50 5 78 2 23 9 72
3 95 9 36 6 66 5 52 0 45 8 30 4 23 2 25 7 17 1 6
4 65 1 8 8 85 0 71 7 65 6 28 5 88 3 76 9 27 2 95
9 37 1 37 4 28 3 51 8 86 2 9 6 55 0 73 7 51 5 90
3 39 2 15 6 83 9 44 7 53 0 16 4 46 5 24 1 25 8 82
1 72 4 48 0 87 2 66 9 5 6 54 7 39 8 35 5 95 3 60
1 46 3 20 0 97 2 21 9 46 7 37 8 19 4 59 6 34 5 55
5 23 3 25 6 78 1 24 0 28 7 83 8 28 9 5 2 73 4 45
1 37 0 53 7 87 4 38 3 71 5 29 9 12 8 33 6 55 2 12
4 90 8 17 2 49 3 83 1 40 6 23 7 65 9 27 5 7 0 48
+++++++++++++++++++++++++++++

instance la23

+++++++++++++++++++++++++++++
Lawrence 15x10 instance (Table 7, instance 3); also called (setb3) or (B3)
15 10
7 84 5 58 8 77 2 44 4 97 6 89 3 5 1 58 9 96 0 9
6 21 1 87 4 15 5 39 2 81 3 85 7 31 8 57 9 73 0 77
0 40 5 71 8 34 9 82 3 70 6 22 4 10 7 80 2 48 1 49
5 75 2 17 3 7 6 72 4 11 7 62 8 47 9 35 1 91 0 55
9 20 4 12 6 71 7 67 0 64 2 94 8 15 5 50 3 75 1 90
6 93 5 93 1 57 7 70 8 77 4 58 0 52 2 29 9 7 3 68
7 56 0 95 8 48 4 26 2 82 1 63 9 36 3 27 6 87 5 6
3 76 5 15 9 78 1 8 8 41 2 36 4 30 6 84 0 36 7 76
0 75 7 13 2 81 8 29 4 54 6 82 5 88 1 78 9 40 3 13
2 6 1 26 7 32 6 64 4 54 0 52 5 82 3 6 9 88 8 54
8 62 2 67 5 32 0 62 7 69 3 61 1 35 4 72 9 5 6 93
2 78 9 90 0 85 1 72 8 64 6 63 3 11 7 82 5 88 4 7
4 28 9 11 7 50 6 88 0 44 5 31 2 27 1 66 8 49 3 35
2 14 5 39 6 56 4 62 3 97 9 66 7 69 1 7 8 47 0 76
1 18 8 93 7 58 6 47 3 69 9 57 2 41 5 53 4 79 0 64
+++++++++++++++++++++++++++++

instance la24

+++++++++++++++++++++++++++++
Lawrence 15x10 instance (Table 7, instance 4); also called (setb4) or (B4)
15 10
7 8 9 75 0 72 6 74 4 30 8 43 2 38 5 98 1 26 3 19
6 19 8 73 3 43 0 23 1 85 4 39 5 13 9 26 2 67 7 9
1 50 3 93 5 80 4 7 0 55 2 61 6 57 8 72 9 42 7 46
1 68 7 43 4 99 6 60 5 68 0 91 8 11 3 96 9 11 2 72
7 84 2 34 8 40 5 7 1 70 6 74 3 12 0 43 9 69 4 30
8 60 0 49 4 59 5 72 9 63 1 69 7 99 6 45 3 27 2 9
6 71 2 91 8 65 1 90 9 98 4 8 7 50 0 75 5 37 3 17
8 62 7 90 5 98 3 31 2 91 4 38 9 72 1 9 0 72 6 49
4 35 0 39 9 74 5 25 7 47 3 52 2 63 8 21 6 35 1 80
9 58 0 5 3 50 8 52 1 88 6 20 2 68 5 24 4 53 7 57
7 99 3 91 4 33 5 19 2 18 6 38 0 24 9 35 1 49 8 9
0 68 3 60 2 77 7 10 8 60 5 15 9 72 1 18 6 90 4 18
9 79 1 60 3 56 6 91 2 40 8 86 7 72 0 80 5 89 4 51
4 10 2 92 5 23 6 46 8 40 7 72 3 6 1 23 0 95 9 34
2 24 5 29 9 49 8 55 0 47 6 77 3 77 7 8 1 28 4 48
+++++++++++++++++++++++++++++

instance la25

+++++++++++++++++++++++++++++
Lawrence 15x10 instance (Table 7, instance 5); also called (setb5) or (B5)
15 10
8 14 4 75 3 12 2 38 0 76 5 97 9 12 1 29 7 44 6 66
5 38 3 82 2 85 4 58 6 87 9 89 0 43 1 80 7 69 8 92
9 5 1 84 0 43 6 48 4 8 7 7 3 41 5 61 8 66 2 14
2 42 1 8 0 96 5 19 4 59 7 97 9 73 8 43 3 74 6 41
6 55 2 70 3 75 8 42 4 37 7 23 1 48 5 5 9 38 0 7
8 9 2 72 7 31 0 79 5 73 3 95 4 25 6 43 9 60 1 56
0 97 2 64 3 78 5 21 4 94 9 31 8 53 6 16 7 86 1 7
3 86 7 85 9 63 0 61 2 65 4 30 5 32 1 33 8 44 6 59
2 44 3 16 4 11 6 45 1 30 9 84 8 93 0 60 5 61 7 90
7 36 8 31 4 47 6 52 0 32 5 11 2 28 9 35 3 20 1 49
8 20 6 49 7 74 4 10 5 17 3 34 0 85 2 77 9 68 1 84
1 85 5 7 8 71 6 59 4 76 0 17 3 29 2 17 7 48 9 13
2 15 6 87 7 11 1 39 4 39 8 43 0 19 3 32 9 16 5 64
6 32 2 92 5 33 8 82 1 83 7 57 9 99 4 91 3 99 0 8
4 88 7 7 8 27 1 38 3 91 2 69 6 21 9 62 5 39 0 48
+++++++++++++++++++++++++++++

instance la26

+++++++++++++++++++++++++++++
Lawrence 20x10 instance (Table 8, instance 1); also called (setc1) or (C1)
20 10
8 52 7 26 6 71 9 16 2 34 1 21 5 95 4 21 0 53 3 55
4 55 5 98 3 39 9 79 0 12 8 77 6 77 7 66 2 31 1 42
5 37 4 92 2 64 6 54 1 19 7 43 0 83 3 34 9 79 8 62
1 87 5 77 0 93 3 69 2 87 7 38 8 24 6 41 9 83 4 60
2 98 5 25 6 75 9 77 1 49 3 17 8 79 0 44 7 43 4 96
1 7 4 61 0 95 2 35 9 10 8 35 5 28 3 76 7 95 6 9
5 59 9 43 0 46 4 28 6 52 3 16 2 59 1 91 8 50 7 27
5 9 9 43 8 14 7 71 4 20 6 54 3 41 0 87 1 45 2 39
1 28 8 66 0 78 2 37 9 42 3 26 5 33 6 89 4 33 7 8
4 96 3 27 6 78 5 84 2 94 8 69 1 74 9 81 7 45 0 69
4 24 7 32 9 25 2 17 3 87 8 81 5 76 6 18 1 31 0 20
8 90 5 28 1 72 7 86 2 23 3 99 6 76 9 97 4 45 0 58
2 17 4 98 3 48 1 46 8 27 6 67 7 62 0 42 9 48 5 27
0 80 8 50 3 19 7 98 5 28 2 50 4 94 6 63 1 12 9 80
9 72 0 75 4 61 8 79 6 37 2 50 5 14 3 55 7 18 1 41
3 96 2 14 5 57 0 47 7 65 4 75 8 79 1 71 6 60 9 22
1 31 7 47 8 58 3 32 4 44 5 58 6 34 0 33 2 69 9 51
1 44 7 40 2 17 0 62 8 66 6 15 3 29 9 38 5 8 4 97
2 58 3 50 4 63 9 87 0 57 6 21 7 57 8 32 1 39 5 20
1 85 0 84 5 56 3 61 9 15 7 70 8 30 2 90 6 67 4 20
+++++++++++++++++++++++++++++

instance la27

+++++++++++++++++++++++++++++
Lawrence 20x10 instance (Table 8, instance 2); also called (setc2) or (C2)
20 10
3 60 4 48 5 95 0 87 1 72 9 5 8 35 7 39 6 54 2 66
7 37 6 34 0 97 5 55 2 21 3 20 4 59 9 46 8 19 1 46
4 45 2 73 1 24 8 28 0 28 3 25 5 23 7 83 9 5 6 78
0 53 2 12 9 12 1 37 8 33 3 71 6 55 5 29 7 87 4 38
4 90 2 49 9 27 7 65 5 7 6 23 0 48 3 83 8 17 1 40
3 85 4 25 2 84 6 64 9 13 1 66 7 46 8 59 0 62 5 19
5 88 6 67 4 14 0 41 1 73 7 57 2 53 3 80 9 47 8 74
1 78 5 64 4 63 6 46 3 84 0 84 8 28 9 52 7 26 2 41
1 11 0 64 6 97 9 38 2 17 4 85 5 73 3 10 8 95 7 67
3 93 2 95 7 43 1 65 8 32 0 59 6 85 5 46 9 85 4 60
2 61 3 41 5 49 4 23 0 66 7 49 8 70 9 99 1 90 6 17
4 13 7 7 1 98 8 57 0 73 3 73 2 68 5 40 9 98 6 9
9 86 6 76 4 14 3 41 1 85 0 37 8 19 2 17 7 54 5 79
1 40 2 53 7 97 5 87 8 96 4 84 3 16 6 66 9 52 0 95
6 33 1 33 3 87 0 18 2 55 8 13 4 77 7 60 9 42 5 74
7 92 5 91 8 79 2 54 4 69 6 79 3 33 1 61 9 39 0 16
6 82 1 41 4 28 5 64 2 78 3 76 7 6 8 49 9 47 0 58
0 52 5 42 8 24 9 91 3 47 6 88 4 91 7 52 2 28 1 35
5 82 2 76 3 86 6 93 4 84 7 38 8 95 9 37 1 21 0 23
9 77 4 8 6 42 7 64 0 70 2 45 8 45 5 28 3 67 1 86

+++++++++++++++++++++++++++++

instance la28

+++++++++++++++++++++++++++++
Lawrence 20x10 instance (Table 8, instance 3); also called (setc3) or (C3)
20 10
8 32 1 81 4 55 7 40 0 6 5 19 9 81 3 37 2 40 6 9
2 70 3 55 7 21 4 64 1 46 8 25 9 65 0 77 5 65 6 15
7 84 4 89 3 24 1 44 2 85 8 31 9 29 6 83 5 37 0 40
4 80 5 59 0 8 2 30 6 77 3 38 1 80 7 56 9 41 8 97
6 40 2 71 0 91 7 7 9 59 8 80 3 50 5 56 1 17 4 88
7 36 9 10 0 45 6 9 4 54 8 96 2 8 5 77 1 29 3 58
6 99 8 86 3 92 0 28 1 98 4 70 5 87 9 96 2 73 7 27
1 95 3 85 5 56 4 52 0 59 2 41 6 81 8 39 9 32 7 92
1 7 7 69 4 93 6 27 5 22 0 88 8 45 3 60 9 49 2 12
7 33 2 61 8 44 5 26 1 84 6 82 3 68 0 21 9 71 4 99
8 43 0 72 4 30 5 98 9 75 1 26 7 8 6 74 3 19 2 38
6 19 2 67 8 73 1 85 9 26 4 39 7 9 0 23 5 13 3 43
8 72 7 46 5 80 3 93 2 61 4 7 9 42 1 50 0 55 6 57
4 99 0 91 9 11 5 68 7 43 3 96 2 72 8 11 6 60 1 68
9 69 0 43 3 12 8 40 1 70 6 74 2 34 5 7 4 30 7 84
7 99 3 27 4 59 5 72 2 9 6 45 0 49 9 63 1 69 8 60
0 75 3 17 2 91 7 50 8 65 5 37 9 98 1 90 6 71 4 8
9 72 1 9 3 31 6 49 2 91 8 62 7 90 0 72 5 98 4 38
4 35 2 63 5 25 6 35 8 21 7 47 3 52 1 80 0 39 9 74
2 68 5 24 9 58 8 52 0 5 6 20 3 50 7 57 1 88 4 53
+++++++++++++++++++++++++++++

instance la29

+++++++++++++++++++++++++++++
Lawrence 20x10 instance (Table 8, instance 4); also called (setc4) or (C4)
20 10
8 14 2 38 7 44 0 76 5 97 3 12 4 75 6 66 9 12 1 29
0 43 2 85 3 82 5 38 4 58 9 89 8 92 6 87 7 69 1 80
3 41 7 7 9 5 0 43 2 14 4 8 5 61 1 84 8 66 6 48
2 42 3 74 4 59 6 41 1 8 9 73 8 43 0 96 5 19 7 97
7 23 8 42 4 37 6 55 0 7 5 5 2 70 9 38 3 75 1 48
8 9 6 43 7 31 4 25 5 73 3 95 0 79 2 72 9 60 1 56
1 7 5 21 8 53 6 16 4 94 0 97 3 78 2 64 7 86 9 31
2 65 6 59 7 85 1 33 4 30 8 44 0 61 3 86 9 63 5 32
6 45 2 44 5 61 8 93 1 30 7 90 9 84 4 11 3 16 0 60
4 47 7 36 8 31 1 49 3 20 2 28 6 52 9 35 5 11 0 32
2 77 4 10 9 68 5 17 0 85 1 84 8 20 6 49 7 74 3 34
0 17 5 7 1 85 3 29 2 17 4 76 6 59 8 71 9 13 7 48
6 87 4 39 8 43 7 11 2 15 3 32 5 64 0 19 1 39 9 16
5 33 3 99 6 32 4 91 8 82 2 92 9 99 7 57 1 83 0 8
3 91 5 39 2 69 8 27 7 7 6 21 1 38 9 62 4 88 0 48
2 67 7 80 3 24 0 88 4 18 1 44 8 45 9 64 5 80 6 38
9 59 3 72 6 47 4 40 7 21 5 43 0 51 8 52 1 24 2 15
3 70 2 31 6 20 8 76 1 40 7 43 0 32 5 88 9 5 4 77
4 47 5 64 9 85 3 49 7 58 1 26 0 32 8 80 2 14 6 94
5 59 2 96 0 5 7 79 8 34 4 75 3 26 6 9 9 23 1 11
+++++++++++++++++++++++++++++

instance la30

+++++++++++++++++++++++++++++
Lawrence 20x10 instance (Table 8, instance 5); also called (setc5) or (C5)
20 10
6 32 3 16 1 33 8 12 7 70 4 10 9 75 0 82 5 88 2 20
8 39 4 81 3 91 5 56 9 69 1 45 6 59 0 86 2 36 7 68
3 84 2 57 7 41 5 73 4 81 0 88 8 38 9 17 6 83 1 5
4 20 5 6 2 15 8 19 1 30 0 94 6 45 7 17 3 18 9 88
9 24 6 49 5 16 4 11 3 60 7 5 8 63 1 25 2 15 0 45
1 86 8 50 2 77 6 54 9 48 0 93 3 32 7 92 5 45 4 71
5 86 6 90 3 78 9 88 2 57 0 32 7 57 8 86 4 71 1 39
2 59 3 18 9 31 4 41 7 20 5 83 8 65 0 54 6 94 1 69
3 47 4 79 6 76 0 59 1 72 2 8 9 30 5 73 7 57 8 84
0 59 2 89 4 10 7 45 3 8 5 54 6 88 8 20 9 7 1 62
5 63 6 9 4 77 3 37 2 5 8 13 9 79 1 24 7 10 0 82
0 74 1 32 2 61 7 53 4 92 9 20 8 10 3 5 6 45 5 23
2 85 9 51 0 61 5 99 4 37 6 94 1 98 8 65 3 33 7 75
0 51 3 24 5 8 6 30 7 12 8 23 2 7 4 17 9 35 1 81
1 71 5 42 8 68 2 31 6 29 3 63 4 65 9 70 7 27 0 93
1 28 5 38 4 51 7 70 2 33 8 78 9 45 3 90 6 54 0 72
0 18 2 90 4 25 6 92 8 85 5 35 7 29 1 81 9 80 3 59
5 67 2 96 1 38 4 86 0 97 3 94 7 86 6 35 9 82 8 45
2 92 8 51 4 59 6 52 5 8 9 70 1 75 3 54 7 60 0 33
3 98 7 80 5 78 0 82 2 7 9 89 1 69 4 51 8 79 6 62
+++++++++++++++++++++++++++++

instance la31

+++++++++++++++++++++++++++++
Lawrence 30x10 instance (Table 9, instance 1); also called (setd1) or (D1)
30 10
4 21 7 26 9 16 2 34 3 55 8 52 5 95 6 71 1 21 0 53
8 77 5 98 1 42 7 66 2 31 3 39 6 77 9 79 4 55 0 12
2 64 4 92 3 34 1 19 8 62 6 54 7 43 0 83 9 79 5 37
0 93 8 24 3 69 7 38 5 77 2 87 4 60 6 41 1 87 9 83
9 77 0 44 4 96 8 79 6 75 2 98 5 25 3 17 7 43 1 49
3 76 2 35 5 28 0 95 7 95 4 61 8 35 1 7 6 9 9 10
1 91 7 27 8 50 3 16 4 28 5 59 6 52 0 46 2 59 9 43
1 45 7 71 2 39 0 87 8 14 6 54 3 41 9 43 5 9 4 20
2 37 3 26 4 33 9 42 0 78 6 89 7 8 8 66 1 28 5 33
1 74 0 69 5 84 3 27 9 81 7 45 8 69 2 94 6 78 4 96
5 76 7 32 6 18 0 20 3 87 2 17 9 25 4 24 1 31 8 81
9 97 8 90 5 28 7 86 0 58 1 72 2 23 6 76 3 99 4 45
9 48 5 27 6 67 7 62 4 98 0 42 1 46 8 27 3 48 2 17
9 80 3 19 5 28 1 12 4 94 6 63 7 98 8 50 0 80 2 50
2 50 1 41 4 61 8 79 5 14 9 72 7 18 3 55 6 37 0 75
9 22 5 57 4 75 2 14 7 65 3 96 1 71 0 47 8 79 6 60
3 32 2 69 4 44 1 31 9 51 0 33 6 34 5 58 7 47 8 58
8 66 7 40 2 17 0 62 9 38 5 8 6 15 3 29 1 44 4 97
3 50 2 58 6 21 4 63 7 57 8 32 5 20 9 87 0 57 1 39
4 20 6 67 1 85 2 90 7 70 0 84 8 30 5 56 3 61 9 15
6 29 0 82 4 18 3 38 7 21 8 50 1 23 5 84 2 45 9 41
3 54 9 37 6 62 5 16 0 52 8 57 4 54 2 38 7 74 1 52
4 79 1 61 8 11 0 81 7 89 6 89 5 57 3 68 9 81 2 30
9 24 1 66 4 32 3 33 8 8 2 20 6 84 0 91 7 55 5 20
3 54 2 64 6 83 9 40 7 8 0 7 4 19 5 56 1 39 8 7
1 6 4 74 0 63 2 64 9 15 6 42 7 98 8 61 5 40 3 91
1 80 3 75 0 26 2 87 9 22 7 39 8 24 4 75 6 44 5 6
5 8 3 79 6 61 1 15 0 12 7 43 8 26 9 22 2 20 4 80
1 36 0 63 7 10 4 22 3 96 5 40 9 5 8 18 6 33 2 62
4 8 8 15 2 64 3 95 1 96 6 38 7 18 9 23 5 64 0 89
+++++++++++++++++++++++++++++

instance la32

+++++++++++++++++++++++++++++
Lawrence 30x10 instance (Table 9, instance 2); also called (setd2) or (D2)
30 10
6 89 1 58 4 97 2 44 8 77 3 5 0 9 5 58 9 96 7 84
7 31 2 81 9 73 4 15 1 87 5 39 8 57 0 77 3 85 6 21
2 48 5 71 0 40 3 70 1 49 6 22 4 10 8 34 7 80 9 82
4 11 6 72 7 62 0 55 2 17 5 75 3 7 1 91 9 35 8 47
0 64 6 71 4 12 1 90 2 94 3 75 9 20 8 15 5 50 7 67
2 29 6 93 3 68 5 93 1 57 8 77 0 52 9 7 4 58 7 70
4 26 3 27 1 63 5 6 6 87 7 56 8 48 9 36 0 95 2 82
1 8 7 76 3 76 4 30 6 84 9 78 8 41 0 36 2 36 5 15
3 13 8 29 0 75 2 81 1 78 5 88 4 54 9 40 7 13 6 82
0 52 2 6 3 6 5 82 6 64 9 88 8 54 4 54 7 32 1 26
8 62 1 35 4 72 7 69 0 62 5 32 9 5 3 61 2 67 6 93
2 78 3 11 7 82 4 7 1 72 8 64 9 90 0 85 5 88 6 63
7 50 4 28 3 35 1 66 2 27 8 49 9 11 6 88 5 31 0 44
4 62 5 39 0 76 2 14 6 56 3 97 1 7 7 69 9 66 8 47
6 47 2 41 0 64 7 58 9 57 8 93 3 69 5 53 1 18 4 79
7 76 9 81 0 76 6 61 4 77 8 26 2 74 5 22 1 58 3 78
6 30 8 72 3 43 0 65 1 16 4 92 5 95 9 29 2 99 7 64
1 35 3 74 5 16 4 85 0 7 2 81 6 86 8 61 9 35 7 34
1 97 7 43 4 72 6 88 5 17 0 43 8 94 3 64 9 22 2 42
7 99 2 84 8 99 5 98 1 20 6 31 3 74 0 92 9 23 4 89
8 32 0 6 4 55 5 19 9 81 1 81 7 40 6 9 3 37 2 40
6 15 2 70 8 25 1 46 9 65 4 64 7 21 0 77 5 65 3 55
8 31 7 84 5 37 3 24 2 85 4 89 9 29 1 44 0 40 6 83
4 80 0 8 9 41 5 59 7 56 3 38 2 30 8 97 6 77 1 80
9 59 0 91 3 50 8 80 1 17 6 40 2 71 5 56 4 88 7 7
7 36 3 58 4 54 5 77 2 8 6 9 0 45 9 10 1 29 8 96
0 28 3 92 2 73 7 27 8 86 5 87 9 96 1 98 6 99 4 70
9 32 1 95 3 85 6 81 2 41 8 39 7 92 0 59 5 56 4 52
4 93 2 12 5 22 6 27 8 45 7 69 3 60 1 7 0 88 9 49
2 61 5 26 9 71 8 44 0 21 6 82 3 68 7 33 1 84 4 99
+++++++++++++++++++++++++++++

instance la33

+++++++++++++++++++++++++++++
Lawrence 30x10 instance (Table 9, instance 3); also called (setd3) or (D3)
30 10
2 38 4 75 9 12 5 97 0 76 1 29 8 14 6 66 7 44 3 12
0 43 5 38 1 80 3 82 2 85 4 58 6 87 8 92 9 89 7 69
6 48 4 8 8 66 7 7 2 14 3 41 5 61 0 43 1 84 9 5
5 19 3 74 6 41 4 59 8 43 2 42 9 73 7 97 1 8 0 96
3 75 5 5 2 70 8 42 7 23 6 55 1 48 9 38 4 37 0 7
2 72 7 31 3 95 0 79 4 25 1 56 8 9 9 60 5 73 6 43
9 31 3 78 6 16 4 94 7 86 5 21 0 97 8 53 1 7 2 64
3 86 2 65 6 59 8 44 1 33 7 85 0 61 5 32 9 63 4 30
4 11 5 61 9 84 3 16 7 90 1 30 0 60 8 93 2 44 6 45
5 11 2 28 0 32 7 36 8 31 4 47 3 20 6 52 9 35 1 49
5 17 3 34 6 49 1 84 0 85 8 20 7 74 9 68 4 10 2 77
8 71 5 7 3 29 1 85 4 76 6 59 2 17 0 17 9 13 7 48
1 39 9 16 4 39 6 87 7 11 3 32 2 15 0 19 5 64 8 43
5 33 8 82 2 92 1 83 6 32 3 99 9 99 4 91 0 8 7 57
7 7 0 48 9 62 4 88 6 21 5 39 8 27 3 91 1 38 2 69
9 64 8 45 3 24 7 80 2 67 4 18 6 38 0 88 5 80 1 44
2 15 3 72 4 40 7 21 8 52 0 51 9 59 1 24 6 47 5 43
4 77 7 43 1 40 2 31 8 76 6 20 5 88 3 70 9 5 0 32
2 14 7 58 9 85 5 64 1 26 6 94 0 32 3 49 8 80 4 47
9 23 1 11 8 34 4 75 7 79 3 26 2 96 0 5 6 9 5 59
0 75 2 20 8 10 3 66 6 43 7 37 1 9 9 83 4 68 5 52
8 54 1 26 4 79 7 88 6 84 0 6 2 54 9 59 3 28 5 42
4 56 9 29 3 36 0 40 6 86 8 68 2 69 7 23 5 62 1 16
7 53 1 5 6 17 9 59 2 59 8 78 3 64 0 82 4 13 5 12
9 7 6 62 7 90 5 83 1 85 3 69 0 16 4 81 2 58 8 66
7 24 2 65 1 69 5 42 9 82 6 82 0 83 3 46 8 72 4 33
1 10 8 27 7 43 5 20 4 71 9 65 2 73 6 99 0 24 3 64
9 35 3 92 0 38 5 35 7 30 8 45 2 8 4 82 1 34 6 21
5 23 7 84 9 7 4 85 8 60 1 15 2 52 6 94 3 83 0 6
2 70 6 29 8 27 9 80 4 6 7 39 1 79 0 28 3 66 5 66
+++++++++++++++++++++++++++++

instance la34

+++++++++++++++++++++++++++++
Lawrence 30x10 instance (Table 9, instance 4); also called (setd4) or (D4)
30 10
2 51 7 59 1 35 5 73 9 65 0 27 6 13 3 81 8 32 4 74
4 64 7 33 5 75 2 33 8 10 0 28 3 38 6 53 9 49 1 55
6 83 1 23 2 72 3 7 9 72 0 6 4 39 5 52 8 90 7 21

3 82 1 23 2 93 4 78 6 88 7 53 9 28 8 65 5 21 0 61
4 41 6 12 9 12 3 77 1 70 7 24 0 81 5 73 2 62 8 6
4 98 3 28 6 42 9 72 0 15 8 15 5 94 2 33 1 51 7 99
0 32 8 22 9 96 4 15 6 78 3 31 5 7 1 94 2 23 7 86
7 93 2 97 3 43 5 73 0 24 8 68 9 88 1 42 4 35 6 72
2 14 0 44 8 13 5 67 1 63 3 49 7 5 4 17 6 85 9 66
7 82 9 15 3 72 4 26 0 8 1 68 6 21 8 45 2 99 5 27
4 93 6 23 0 51 8 54 3 49 1 96 2 56 9 36 5 53 7 52
8 60 0 14 4 70 9 55 1 23 5 83 3 38 2 24 7 37 6 48
0 62 7 15 8 69 9 23 1 82 6 26 4 45 5 33 3 12 2 37
6 72 1 9 7 15 5 28 8 92 9 12 0 59 3 64 4 87 2 73
0 50 1 14 7 90 5 46 3 71 4 48 2 80 9 61 8 24 6 44
0 22 9 94 5 16 3 73 2 54 8 54 4 46 1 97 6 61 7 75
9 55 3 67 6 77 4 30 7 6 1 32 8 47 5 93 2 6 0 40
1 30 3 98 7 79 0 22 6 79 2 7 8 36 9 36 5 9 4 92
8 37 7 72 2 52 4 31 1 82 9 54 5 7 6 82 3 73 0 49
1 73 3 83 7 45 2 76 4 43 9 29 0 35 5 92 8 39 6 28
2 58 0 26 1 48 8 52 7 34 6 96 5 70 4 98 3 80 9 94
1 70 8 23 5 26 4 14 6 90 2 93 3 21 0 42 7 18 9 36
4 28 6 76 7 25 0 17 1 84 2 67 8 87 3 43 9 88 5 84
7 30 3 91 8 52 4 80 0 21 5 8 9 37 2 15 6 12 1 92
1 28 4 7 7 46 6 92 2 77 3 15 9 69 8 54 0 47 5 39
9 50 5 44 2 64 8 38 4 93 6 33 7 75 0 41 1 24 3 5
7 94 0 17 6 87 2 21 8 92 9 28 1 61 4 63 3 34 5 77
3 72 8 98 9 5 4 28 2 9 5 95 6 64 1 43 0 50 7 96
0 85 2 85 8 39 1 98 7 24 3 71 5 60 4 55 9 22 6 35
3 78 6 49 2 46 1 11 0 90 5 20 9 34 7 6 4 70 8 74
+++++++++++++++++++++++++++++

instance la35

+++++++++++++++++++++++++++++
Lawrence 30x10 instance (Table 9, instance 5); also called (setd5) or (D5)
30 10
0 66 2 84 3 26 7 29 9 94 6 98 8 7 5 98 1 45 4 43
3 32 0 97 6 55 2 88 8 93 9 88 1 20 4 50 7 17 5 5
4 43 3 68 8 47 9 68 1 57 6 20 5 81 2 60 7 94 0 62
1 57 5 40 0 78 6 9 2 49 9 17 3 32 4 30 8 87 7 77
0 52 4 30 3 48 5 48 1 26 9 17 6 93 8 97 7 49 2 89
7 95 0 33 1 5 6 17 5 70 3 57 4 34 2 61 8 62 9 39
7 97 5 92 1 31 8 5 2 79 4 5 3 67 0 5 9 78 6 60
2 79 4 6 7 20 8 45 6 34 3 24 9 26 5 68 1 16 0 46
7 58 9 50 2 19 8 93 6 49 3 25 5 85 4 50 0 93 1 26
9 81 6 71 5 7 1 39 2 16 8 42 0 71 4 84 3 56 7 99
8 9 0 86 9 6 3 71 6 97 5 85 4 16 2 42 7 81 1 81
4 72 3 24 0 30 8 56 2 43 1 61 7 82 6 40 5 59 9 43
9 43 1 13 6 70 7 93 0 95 8 12 4 15 2 78 5 97 3 14
0 14 6 26 1 71 3 46 8 80 5 31 4 37 9 27 7 92 2 67
2 12 0 43 5 96 6 7 3 45 7 20 1 13 9 29 4 60 8 33
1 78 5 50 6 84 0 42 8 84 4 30 9 76 2 57 7 87 3 59
4 49 7 50 1 15 8 13 0 93 6 50 9 32 5 59 3 10 2 35
1 25 0 47 7 60 8 33 4 53 5 37 9 73 2 22 3 87 6 79
0 84 6 83 1 71 5 68 9 89 8 11 3 60 4 50 2 33 7 97
1 14 0 38 6 88 5 5 4 77 7 92 8 24 2 73 9 52 3 71
7 62 9 19 6 38 3 15 8 64 2 64 4 8 1 61 0 19 5 33
2 33 5 46 4 74 0 56 6 84 9 83 8 19 7 8 3 32 1 97
4 50 3 71 6 50 2 97 9 8 0 17 7 19 8 92 5 54 1 52
8 32 1 79 3 97 5 38 9 49 4 76 6 76 0 56 2 78 7 54
5 13 3 5 2 25 0 86 1 95 9 28 6 78 8 24 7 10 4 39
7 48 2 59 0 20 9 7 5 31 6 97 1 89 4 32 3 25 8 41
5 87 0 18 9 48 2 43 1 30 6 97 7 47 8 65 3 69 4 27
6 71 5 20 8 20 1 78 3 39 0 17 7 50 2 44 9 42 4 38
0 50 9 42 3 72 5 7 1 77 7 58 4 78 2 89 6 70 8 36
3 32 9 95 2 13 0 73 6 97 8 24 4 49 5 57 1 68 7 94
+++++++++++++++++++++++++++++

instance la36

+++++++++++++++++++++++++++++
Lawrence 15x15 instance (Table 10, instance 1); also called (seti1) or (I1)
15 15
4 21 3 55 6 71 14 98 10 12 2 34 9 16 1 21 0 53 7 26 8 52 5 95 12 31 11 42 13 39
11 54 4 83 1 77 7 64 8 34 14 79 12 43 0 55 3 77 6 19 9 37 5 79 10 92 13 62 2 66
9 83 5 77 2 87 7 38 4 60 12 98 0 93 13 17 6 41 10 44 3 69 11 49 8 24 1 87 14 25
5 77 0 96 9 28 6 7 4 95 13 35 7 35 8 76 11 9 12 95 2 43 1 75 10 61 14 10 3 79
10 87 4 28 8 50 2 59 0 46 11 45 14 9 9 43 6 52 7 27 1 91 13 41 3 16 5 59 12 39
0 20 2 71 4 78 13 66 3 14 12 8 14 42 6 28 1 54 9 33 11 89 8 26 7 37 10 33 5 43
8 69 4 96 12 17 0 69 7 45 11 31 6 78 10 20 3 27 13 87 1 74 5 84 14 76 2 94 9 81
4 58 13 90 11 76 3 81 7 23 9 28 1 18 2 32 12 86 8 99 14 97 0 24 10 45 6 72 5 25
5 27 1 46 6 67 8 27 13 19 10 80 2 17 3 48 7 62 11 12 14 28 4 98 0 42 9 48 12 50
11 37 5 80 4 75 8 55 7 50 0 94 9 14 6 41 14 72 3 50 10 61 13 79 2 98 12 18 1 63
7 65 3 96 0 47 4 75 12 69 14 58 10 33 1 71 9 22 13 32 5 57 8 79 2 14 11 31 6 60
1 34 2 47 3 58 5 51 4 62 6 44 9 8 7 17 10 97 8 29 11 15 13 66 12 40 0 44 14 38
3 50 7 57 13 61 5 20 11 85 12 90 2 58 4 63 10 84 1 39 9 87 6 21 14 56 8 32 0 57
9 84 7 45 5 15 14 41 10 18 4 82 11 29 2 70 1 67 3 30 13 50 6 23 0 20 12 21 8 38
9 37 10 81 11 61 14 57 8 57 0 52 7 74 6 62 12 30 1 52 2 38 13 68 4 54 3 54 5 16
+++++++++++++++++++++++++++++

instance la37

+++++++++++++++++++++++++++++
Lawrence 15x15 instance (Table 10, instance 2); also called (seti2) or (I2)
15 15
5 19 6 64 11 73 9 13 2 84 14 88 3 85 10 41 12 53 13 80 1 66 7 46 8 59 4 25 0 62
1 67 3 74 7 41 2 57 14 52 0 14 9 64 8 84 6 78 5 47 13 28 4 84 10 63 12 26 11 46
6 97 8 95 0 64 9 38 10 59 12 95 2 17 11 65 13 93 3 10 5 73 1 11 4 85 14 46 7 67
10 23 12 49 3 32 4 66 2 43 0 60 8 41 7 61 13 70 9 49 11 17 6 90 1 85 14 99 5 85
9 98 8 57 3 73 6 9 0 73 7 7 1 98 4 13 13 41 5 40 11 85 10 37 2 68 14 79 12 17
11 66 7 53 5 86 6 40 0 14 3 19 13 96 4 95 2 54 10 84 12 97 8 16 14 52 1 76 9 87
4 77 2 55 9 42 5 74 14 91 13 33 10 16 12 54 0 18 3 87 7 60 8 13 6 33 1 33 11 61
6 41 5 39 11 82 9 64 14 47 10 28 7 78 13 49 1 79 4 58 2 92 3 79 12 6 0 69 8 76
11 21 5 42 9 91 2 28 0 52 6 88 12 76 13 86 10 23 1 35 7 52 4 91 3 47 14 82 8 24
11 42 1 93 3 95 13 45 9 28 14 77 0 84 10 8 7 45 4 70 5 37 6 86 12 64 8 67 2 38
4 97 12 81 1 58 7 84 5 58 0 9 11 87 3 5 2 44 13 85 6 89 10 77 9 96 14 39 8 77
12 80 1 21 10 10 5 73 8 70 6 49 2 31 13 34 4 40 11 22 0 15 14 82 3 57 9 71 7 48
2 17 7 62 5 75 9 35 1 91 14 50 3 7 10 64 13 75 12 94 0 55 6 72 8 47 4 11 11 90
11 93 6 57 1 71 12 70 9 93 5 20 3 15 13 77 10 58 0 12 2 67 8 68 14 7 7 29 4 52
13 76 3 27 4 26 9 36 11 8 10 36 0 95 8 48 2 82 6 87 5 6 1 63 7 56 12 36 14 15
+++++++++++++++++++++++++++++

instance la38

+++++++++++++++++++++++++++++
Lawrence 15x15 instance (Table 10, instance 3); also called (seti3) or (I3)
15 15
1 26 12 67 0 72 6 74 14 13 8 43 4 30 3 19 10 23 11 85 5 98 13 43 2 38 7 8 9 75
14 42 0 39 4 55 12 46 1 19 8 93 9 80 5 26 10 7 6 50 11 57 3 73 2 9 7 61 13 72
3 96 4 99 12 34 6 60 7 43 14 7 13 12 8 11 11 70 10 43 0 91 1 68 9 11 5 68 2 72
14 63 11 45 4 49 1 74 8 27 0 30 9 72 7 9 12 99 13 60 5 69 6 69 2 84 3 40 10 59
2 91 0 75 9 98 3 17 10 72 13 31 11 9 14 98 7 50 5 37 4 8 8 65 1 90 12 91 6 71
11 35 6 80 4 39 3 62 14 74 5 72 10 35 9 25 1 49 8 52 7 63 2 90 13 21 12 47 0 38
14 19 7 57 10 24 13 91 3 50 0 5 11 49 12 18 9 58 5 24 8 52 1 88 2 68 6 20 4 53
7 77 14 72 5 35 11 90 4 68 6 18 3 9 0 33 8 60 10 18 12 10 13 60 1 38 2 99 9 15
13 6 8 86 2 40 9 79 12 92 11 23 5 89 10 95 6 91 7 72 0 80 1 60 3 56 4 51 14 23
1 46 6 28 5 34 11 77 4 47 0 10 14 49 8 77 10 48 7 24 12 8 2 72 13 55 9 29 3 40
10 22 4 89 12 79 0 7 9 15 1 6 11 30 6 38 5 11 8 52 3 20 7 5 14 9 2 20 13 28
5 73 14 56 2 37 3 22 13 25 6 58 1 8 7 93 4 88 8 17 12 9 11 69 10 71 9 85 0 55
9 85 14 58 3 46 8 64 2 49 6 37 1 33 4 30 5 26 0 20 13 74 10 77 12 99 11 56 7 21
10 17 3 24 4 89 5 15 11 60 1 42 8 98 2 64 13 92 0 63 7 52 12 54 6 75 14 23 9 38
3 8 5 17 11 56 7 93 14 26 9 62 6 7 10 88 0 97 1 7 2 43 8 29 13 35 12 87 4 57
+++++++++++++++++++++++++++++

instance la39

+++++++++++++++++++++++++++++
Lawrence 15x15 instance (Table 10, instance 4); also called (seti4) or (I4)
15 15
10 51 14 43 7 80 4 18 6 38 3 24 2 67 12 15 11 24 13 72 8 45 5 80 9 64 1 44 0 88
6 40 9 88 10 77 5 59 11 20 3 52 8 70 0 40 4 32 13 76 12 43 7 31 2 21 14 5 1 47
0 32 3 49 10 5 5 64 7 58 8 80 6 94 11 11 1 26 13 26 14 59 9 85 4 47 12 96 2 14
5 23 6 9 0 75 12 37 11 43 2 79 4 75 3 34 7 20 13 10 14 83 10 68 9 52 8 66 1 9
12 69 9 59 3 28 14 62 13 36 1 26 6 84 11 16 8 54 5 42 2 54 0 6 10 40 7 88 4 79
13 78 12 53 11 17 5 29 4 82 2 23 9 12 8 64 1 86 7 59 6 5 3 68 14 59 10 13 0 56
10 83 13 46 9 7 12 65 11 69 6 62 0 16 2 58 8 66 5 83 7 90 14 42 4 81 3 69 1 85
7 73 10 71 8 64 6 10 9 20 11 99 4 24 14 65 5 82 3 72 12 43 1 82 13 27 2 24 0 33
4 82 1 34 3 92 2 8 0 38 8 45 6 21 5 35 12 52 9 35 11 15 14 23 10 6 13 83 7 30
2 84 5 7 9 66 10 6 4 28 13 27 6 79 7 70 0 85 1 94 3 60 14 80 12 39 8 66 11 29
3 44 6 58 13 14 8 65 1 72 5 14 12 52 4 21 9 25 0 5 11 51 7 61 14 55 10 42 2 36

14 43 10 72 5 78 11 12 12 17 0 46 9 27 6 51 2 63 1 79 8 79 7 91 4 49 13 26 3 93
7 49 0 49 4 71 5 78 9 44 10 41 12 91 13 84 8 91 6 21 11 47 14 28 3 61 2 70 1 93
3 25 4 85 0 66 2 45 10 95 12 21 8 84 5 24 9 53 7 67 6 91 11 11 13 32 1 30 14 89
3 92 7 93 0 99 1 40 10 37 12 69 5 66 6 57 14 22 9 44 8 73 13 97 11 18 2 69 4 41
+++++++++++++++++++++++++++++

instance la40

+++++++++++++++++++++++++++++
Lawrence 15x15 instance (Table 10, instance 5); also called (seti5) or (I5)
15 15
9 65 10 28 4 74 12 33 2 51 14 75 5 73 8 32 6 13 3 81 1 35 7 59 13 38 11 55 0 27
0 64 1 53 11 83 2 33 4 6 9 52 14 72 8 7 13 90 12 21 6 23 3 10 10 39 5 49 7 72
14 73 3 82 1 23 12 62 6 88 5 21 8 65 11 70 7 53 10 81 2 93 13 77 0 61 9 28 4 78
1 12 6 51 7 33 4 15 14 72 10 98 9 94 5 12 11 42 2 24 13 15 8 28 3 6 12 99 0 41
12 97 5 7 9 96 4 15 14 73 13 43 0 32 8 22 11 42 1 94 2 23 7 86 6 78 10 24 3 31
1 72 5 88 2 93 13 13 4 44 14 66 6 63 7 14 9 67 10 17 11 85 0 35 3 68 12 5 8 49
9 15 7 82 6 21 14 53 3 72 13 49 2 99 4 26 12 56 8 45 1 68 10 51 0 8 5 27 11 96
3 54 7 24 4 14 8 38 5 36 2 52 14 55 12 37 11 48 0 93 13 60 10 70 1 23 6 23 9 83
3 12 8 69 6 26 9 23 14 28 1 82 5 33 4 45 13 64 7 15 11 9 12 73 10 59 2 37 0 62
0 87 5 12 7 80 4 50 10 48 12 90 1 72 13 24 6 14 8 71 11 44 9 46 2 15 14 61 3 92
2 54 0 22 6 61 4 46 3 73 5 16 12 6 9 94 14 93 13 67 8 54 7 75 11 32 10 40 1 97
10 92 14 36 4 22 9 9 3 47 1 77 12 79 13 36 6 30 8 98 11 79 7 7 5 55 2 6 0 30
0 49 13 83 3 73 6 82 1 82 14 92 11 73 4 31 10 35 9 54 5 7 8 37 7 72 2 52 12 76
10 98 12 34 13 52 4 26 1 28 3 39 8 80 5 29 9 70 0 43 6 48 7 58 2 45 14 94 11 96
1 70 10 17 6 90 12 67 4 14 8 23 3 21 7 18 13 43 11 84 5 26 9 36 2 93 14 84 0 42
+++++++++++++++++++++++++++++

instance orb01

+++++++++++++++++++++++++++++
trivial 10x10 instance from Bill Cook (BIC2)
10 10
0 72 1 64 2 55 3 31 4 53 5 95 6 11 7 52 8 6 9 84
0 61 3 27 4 88 2 78 1 49 5 83 8 91 6 74 7 29 9 87
0 86 3 32 1 35 2 37 5 18 4 48 6 91 7 52 9 60 8 30
0 8 1 82 4 27 3 99 6 74 5 9 2 33 9 20 7 59 8 98
1 50 0 94 5 43 3 62 4 55 7 48 2 5 8 36 9 47 6 36
0 53 6 30 2 7 3 12 1 68 8 87 4 28 9 70 7 45 5 7
2 29 3 96 0 99 1 14 4 34 7 14 5 7 6 76 8 57 9 76
2 90 0 19 3 87 4 51 1 84 5 45 9 84 6 58 7 81 8 96
2 97 1 99 4 93 0 38 7 13 5 96 3 40 9 64 6 32 8 45
2 44 0 60 8 29 3 5 6 74 1 85 4 34 7 95 9 51 5 47
+++++++++++++++++++++++++++++

instance orb02

+++++++++++++++++++++++++++++
doomed 10x10 instance from Monika (MON2)
10 10
0 72 1 54 2 33 3 86 4 75 5 16 6 96 7 7 8 99 9 76
0 16 3 88 4 48 8 52 9 60 6 29 7 18 5 89 2 80 1 76
0 47 7 11 3 14 2 56 6 16 4 83 1 10 5 61 8 24 9 58
0 49 1 31 3 17 8 50 5 63 2 35 4 65 7 23 6 50 9 29
0 55 6 6 1 28 3 96 5 86 2 99 9 14 7 70 8 64 4 24
4 46 0 23 6 70 8 19 2 54 3 22 9 85 7 87 5 79 1 93
4 76 3 60 0 76 9 98 2 76 1 50 8 86 7 14 6 27 5 57
4 93 6 27 9 57 3 87 8 86 2 54 7 24 5 49 0 20 1 47
2 28 6 11 8 78 7 85 4 63 9 81 3 10 1 9 5 46 0 32
2 22 9 76 5 89 8 13 6 88 3 10 7 75 4 98 1 78 0 17
+++++++++++++++++++++++++++++

instance orb03

+++++++++++++++++++++++++++++
deadlier 10x10 instance from Bruce Gamble (BRG1)
10 10
0 96 1 69 2 25 3 5 4 55 5 15 6 88 7 11 8 17 9 82
0 11 1 48 2 67 3 38 4 18 7 24 6 62 5 92 9 96 8 81
2 67 1 63 0 93 4 85 3 25 5 72 6 51 7 81 8 58 9 15
2 30 1 35 0 27 4 82 3 44 7 92 6 25 5 49 9 28 8 77
1 53 0 83 4 73 3 26 2 77 6 33 5 92 9 99 8 38 7 38
1 20 0 44 4 81 3 88 2 66 6 70 5 91 9 37 8 55 7 96
1 21 2 93 4 22 0 56 3 34 6 40 7 53 9 46 5 29 8 63
1 32 2 63 4 36 0 26 3 17 5 85 7 15 8 55 9 16 6 82
0 73 2 46 3 89 4 24 1 99 6 92 7 7 9 51 5 19 8 14
0 52 2 20 3 70 4 98 1 23 5 15 7 81 8 71 9 24 6 81
+++++++++++++++++++++++++++++

instance orb04

+++++++++++++++++++++++++++++
deadly 10x10 instance from Bruce Shepherd (BRS1)
10 10
0 8 1 10 2 35 3 44 4 15 5 92 6 70 7 89 8 50 9 12
0 63 8 39 3 80 5 22 2 88 1 39 9 85 6 27 7 74 4 69
0 52 6 22 1 33 3 68 8 27 2 68 5 25 4 34 7 24 9 84
0 31 1 85 4 55 8 80 5 58 7 11 6 69 9 56 3 73 2 25
0 97 5 98 9 87 8 47 7 77 4 90 3 98 2 80 1 39 6 40
1 97 5 68 0 44 9 67 2 44 8 85 3 78 6 90 7 33 4 81
0 34 3 76 8 48 7 61 9 11 2 36 4 33 6 98 1 7 5 44
0 44 9 5 4 85 1 51 5 58 7 79 2 95 6 48 3 86 8 73
0 24 1 63 9 48 7 77 8 73 6 74 4 63 5 17 2 93 3 84
0 51 2 5 4 40 9 60 1 46 5 58 8 54 3 72 6 29 7 94
+++++++++++++++++++++++++++++

instance orb05

+++++++++++++++++++++++++++++
10x10 instance from George Steiner (GES1)
10 10
9 11 8 93 0 48 7 76 6 13 5 71 3 59 2 90 4 10 1 65
8 52 9 76 0 84 7 73 5 56 4 10 6 26 2 43 3 39 1 49
9 28 8 44 7 26 6 66 4 68 5 74 3 27 2 14 1 6 0 21
0 18 1 58 3 62 2 46 6 25 4 6 5 60 7 28 8 80 9 30
0 78 1 47 7 29 5 16 4 29 6 57 3 78 2 87 8 39 9 73
9 66 8 51 3 12 7 64 5 67 4 15 6 66 2 26 1 20 0 98
8 23 9 76 6 45 7 75 5 24 3 18 4 83 2 15 1 88 0 17
9 56 8 83 7 80 6 16 4 31 5 93 3 30 2 29 1 66 0 28
9 79 8 69 2 82 4 16 5 62 3 41 6 91 7 35 0 34 1 75
0 5 1 19 2 20 3 12 4 94 5 60 6 99 7 31 8 96 9 63
+++++++++++++++++++++++++++++

Instance TA01 (15x15)
+++++++++++++++++++++++++++++
Times
94 66 10 53 26 15 65 82 10 27 93 92 96 70 83
74 31 88 51 57 78 8 7 91 79 18 51 18 99 33
4 82 40 86 50 54 21 6 54 68 82 20 39 35 68
73 23 30 30 53 94 58 93 32 91 30 56 27 92 9
78 23 21 60 36 29 95 99 79 76 93 42 52 42 96
29 61 88 70 16 31 65 83 78 26 50 87 62 14 30
18 75 20 4 91 68 19 54 85 73 43 24 37 87 66
32 52 9 49 61 35 99 62 6 62 7 80 3 57 7
85 30 96 91 13 87 82 83 78 56 85 8 66 88 15
5 59 30 60 41 17 66 89 78 88 69 45 82 6 13
90 27 1 8 91 80 89 49 32 28 90 93 6 35 73
47 43 75 8 51 3 84 34 28 60 69 45 67 58 87
65 62 97 20 31 33 33 77 50 80 48 90 75 96 44
28 21 51 75 17 89 59 56 63 18 17 30 16 7 35
57 16 42 34 37 26 68 73 5 8 12 87 83 20 97
Machines
7 13 5 8 4 3 11 12 9 15 10 14 6 1 2
5 6 8 15 14 9 12 10 7 11 1 4 13 2 3
2 9 10 13 7 12 14 6 1 3 8 11 5 4 15
6 3 10 7 11 1 14 5 8 15 12 9 13 2 4
8 9 7 11 5 10 3 15 13 6 2 14 12 1 4
6 4 13 14 12 5 15 8 3 2 11 1 10 7 9
13 4 8 9 15 7 2 12 5 6 3 11 1 14 10
12 6 1 8 13 14 15 2 3 9 5 4 10 7 11
11 12 7 15 1 2 3 6 13 5 9 8 10 14 4
7 12 10 3 9 1 14 4 11 8 2 13 15 5 6
5 8 14 1 6 13 7 9 15 11 4 2 12 10 3
3 15 1 13 7 11 8 6 9 10 14 2 4 12 5
6 9 11 3 4 7 10 1 14 5 2 12 13 8 15
9 15 5 14 6 7 10 2 13 8 12 11 4 3 1
11 9 13 7 5 2 14 15 12 1 8 4 3 10 6
+++++++++++++++++++++++++++++

instance TA02 (15x15)
+++++++++++++++++++++++++++++

Times
86 60 10 59 65 94 71 25 98 49 43 8 90 21 73
68 28 38 36 93 35 37 28 62 86 65 11 20 82 23
33 67 96 91 83 81 60 88 20 62 22 79 38 40 82
13 14 73 88 24 16 78 70 53 68 73 90 58 7 4
93 52 63 13 19 41 71 59 19 60 85 99 73 95 19
62 60 93 16 10 72 88 69 58 41 46 63 76 83 62
50 68 90 34 44 5 8 25 70 53 78 92 62 85 70
60 64 92 44 63 91 21 1 96 19 59 12 41 11 94
93 46 51 37 91 90 63 40 68 13 16 83 49 24 23
5 35 21 14 66 3 6 98 63 64 76 94 17 62 37
35 42 62 68 73 27 52 39 41 25 9 34 50 41 98
23 32 35 10 29 68 20 8 58 62 39 32 8 33 91
28 31 3 28 66 59 24 45 81 8 44 42 2 23 53
11 93 27 59 62 23 23 7 77 64 60 97 36 53 72
36 98 38 24 84 47 72 1 91 85 68 42 20 30 30
Machines
10 15 5 14 11 4 8 9 1 6 2 3 13 7 12
11 9 12 15 4 14 10 8 5 3 7 2 6 13 1
8 1 7 6 15 14 3 12 5 13 2 10 4 11 9
10 12 15 1 2 9 6 11 13 5 14 4 7 8 3
12 5 14 4 9 2 11 13 3 15 7 8 1 10 6
6 3 2 11 1 5 9 15 7 4 10 8 12 13 14
6 11 14 1 10 9 2 12 15 8 13 3 7 5 4
13 1 10 4 14 7 6 8 3 15 12 9 11 2 5
12 11 6 14 2 10 9 8 4 7 1 3 15 13 5

3 15 4 11 7 2 1 14 12 5 6 9 8 13 10
12 15 14 6 5 10 2 7 13 1 3 9 11 4 8
13 4 11 9 5 8 14 12 15 2 3 1 6 7 10
9 14 6 1 12 10 5 13 2 11 7 3 8 15 4
3 6 5 4 10 2 12 14 8 7 11 15 1 9 13
2 11 5 3 1 8 7 10 12 13 6 15 4 14 9
++++++++++++++++++++++++++++++

instance TA03 (15x15)

++++++++++++++++++++++++++++++
Times
69 81 81 62 80 3 38 62 54 66 88 82 3 12 88
83 51 47 15 89 76 52 18 22 85 26 30 5 89 22
62 47 93 54 38 78 71 96 19 33 44 71 90 9 21
33 82 80 30 96 31 11 26 41 55 12 10 92 3 75
36 49 10 43 69 72 19 65 37 57 32 11 73 89 12
83 32 6 13 87 94 36 76 46 30 56 62 32 52 72
29 78 21 27 17 43 14 15 16 49 72 19 99 38 64
12 74 4 3 15 62 50 38 49 25 18 55 5 71 27
69 13 33 47 86 31 97 48 25 40 94 22 61 59 16
27 4 35 80 49 46 84 46 96 72 18 23 96 74 23
36 17 81 67 47 5 51 23 82 35 96 7 54 92 38
78 58 62 43 1 56 76 49 80 26 79 9 24 24 42
38 86 38 38 83 36 11 17 99 14 57 64 58 96 17
10 86 93 63 61 62 75 90 40 77 8 27 96 69 64
73 12 14 71 3 47 84 84 53 58 95 87 90 68 75
Machines
8 12 9 4 13 2 14 1 15 7 10 5 3 11 6
13 2 12 10 7 4 3 5 6 9 14 15 11 1 8
2 3 10 1 4 6 9 5 15 11 13 14 8 7 12
14 11 7 3 15 8 5 12 1 6 10 4 9 2 13
2 9 5 15 7 6 4 3 10 11 14 8 12 1 13
6 15 3 13 11 2 12 5 7 10 1 14 9 4 8
6 3 1 2 9 15 12 11 8 10 7 13 5 14 4
5 8 11 2 10 9 3 15 12 4 6 7 14 13 1
15 12 1 10 11 6 4 13 9 14 7 2 8 3 5
10 1 4 11 13 14 6 2 7 15 9 12 3 8 5
8 3 2 13 4 15 5 7 6 10 9 14 11 1 12
1 9 15 13 10 6 7 11 8 12 4 5 2 14 3
9 13 11 12 15 4 7 2 5 6 1 10 14 3 8
14 3 12 1 15 11 4 2 13 5 6 7 8 10 9
2 14 1 12 3 11 5 9 4 6 8 7 10 13 15
++++++++++++++++++++++++++++++

instance TA11 (20x15)

++++++++++++++++++++++++++++++
Times
25 75 75 76 38 62 38 59 14 13 46 31 57 92 3
67 5 11 11 40 34 77 42 35 96 22 55 21 29 16
22 98 8 35 59 31 13 46 52 22 18 19 64 29 70
99 42 2 35 11 92 88 97 21 56 17 43 27 19 23
50 5 59 71 47 39 82 35 12 2 39 42 52 65 35
48 57 5 2 60 64 86 3 51 26 34 39 45 63 54
40 43 50 71 46 99 67 34 6 95 67 54 29 30 60
59 3 85 6 46 49 5 82 18 71 48 79 62 65 76
65 55 81 15 32 52 97 69 82 89 69 87 22 71 63
70 74 52 94 14 81 24 14 32 39 67 59 18 77 50
18 6 96 53 35 99 39 18 14 90 64 81 89 48 80
44 75 12 13 74 59 71 75 30 93 26 30 84 91 93
39 56 13 29 55 69 26 7 55 48 22 46 50 96 17
57 14 8 13 95 53 78 24 92 90 68 87 43 75 94
93 92 18 28 27 40 56 83 51 15 97 48 53 78 39
47 34 42 28 11 11 30 14 10 4 20 92 19 59 28
69 82 64 40 27 82 27 43 56 17 18 20 98 43 68
84 26 87 61 95 23 88 89 49 84 12 51 3 44 20
43 54 18 72 70 28 20 22 59 36 85 13 73 29 45
7 97 4 22 74 45 62 95 66 14 40 23 79 34 8
Machines
4 12 15 2 11 3 5 8 1 13 6 10 7 14 9
6 1 4 9 5 2 13 15 7 8 11 3 10 14 12
3 4 15 1 10 13 6 5 8 11 9 12 14 2 7
9 11 2 14 4 5 15 10 3 6 12 8 1 7 13
15 9 2 3 11 10 13 5 7 6 1 14 4 12 8
4 11 2 6 7 1 9 8 12 14 3 15 13 10 5
3 11 2 13 9 1 8 7 15 14 5 4 6 10 12
2 1 3 5 8 14 12 4 13 6 7 15 10 9 11
5 6 10 11 8 7 3 2 13 4 14 1 9 15 12
2 5 4 11 15 1 7 14 12 9 6 13 8 10 3
4 11 2 1 10 9 15 7 5 8 3 13 6 12 14
3 8 7 9 4 6 15 5 2 1 10 11 14 12 13
1 8 15 9 13 11 10 4 7 2 5 3 12 14 6
13 4 10 5 2 1 11 7 6 3 15 14 8 9 12
4 15 7 6 14 10 2 1 13 8 3 5 11 9 12
6 15 7 13 9 3 5 10 12 14 4 2 8 1 11
4 8 11 15 1 9 2 12 6 14 5 13 7 10 3
11 9 3 12 14 7 15 4 10 8 5 6 13 1 2
4 3 13 14 2 7 15 6 5 9 10 12 1 11 8
12 15 6 7 11 10 14 2 5 9 1 4 13 3 8
++++++++++++++++++++++++++++++

instance TA12 (20x15)

++++++++++++++++++++++++++++++
Times
55 66 48 59 8 21 64 7 80 5 59 8 91 11 81
86 76 40 76 9 23 80 51 46 48 68 51 15 5 82
84 97 26 70 33 31 20 39 42 33 70 84 23 54 55
60 82 14 36 22 21 3 11 82 92 52 85 77 3 89
83 33 15 36 96 99 81 24 59 89 11 13 26 91 87
51 20 89 99 95 41 7 67 77 45 74 91 87 1 55
35 71 47 34 77 68 85 27 2 99 9 18 28 33 92
76 58 37 28 80 96 97 92 84 68 1 86 33 66 20
17 11 18 90 57 95 17 33 61 49 36 38 62 73 25
82 84 87 44 96 64 68 57 65 89 42 77 43 76 38
54 66 8 48 84 15 93 94 57 16 64 13 62 63 53
21 70 42 29 83 5 16 76 67 46 67 83 46 29 26
96 42 49 54 58 8 41 14 35 9 74 16 50 69 45
69 90 17 18 45 48 31 29 27 85 71 92 20 11 86
41 24 82 50 24 75 34 80 71 54 5 42 8 35 93
63 4 85 53 61 54 16 18 5 43 24 88 67 79 41
17 37 56 70 56 24 95 12 96 27 55 36 41 65 23
79 6 89 69 16 56 81 98 12 19 88 3 36 67 74
38 76 47 21 80 97 35 45 74 92 98 54 91 79 46
34 56 26 62 82 38 89 33 50 62 39 63 88 13 42
Machines
3 6 2 9 4 5 15 8 11 14 12 10 13 7 1
15 9 13 5 12 4 7 10 1 11 2 14 3 8 6
8 13 2 9 3 11 4 12 14 15 6 7 1 10 5
2 9 6 12 8 7 4 3 5 10 13 14 1 15 11
9 5 13 2 4 15 3 10 14 7 6 11 12 8 1
3 15 11 8 4 1 2 14 10 7 12 13 5 6 9
1 5 14 2 9 11 12 7 10 3 6 13 8 4 15
13 3 12 10 9 11 14 5 6 15 7 4 2 8 1
8 4 5 6 14 1 12 11 10 2 9 7 13 15 3
8 15 13 7 6 10 11 1 4 5 3 2 9 14 12
1 10 14 8 7 4 12 9 11 5 3 2 15 13 6
15 11 4 13 6 8 5 7 2 3 1 14 10 12 9
12 11 6 2 4 14 10 8 9 15 1 3 7 13 5
8 7 14 11 4 2 6 12 5 9 1 13 10 3 15
3 5 15 6 1 8 11 13 2 10 4 9 7 14 12
6 15 13 8 4 14 1 10 11 7 2 3 9 5 12
15 9 2 11 12 13 8 1 4 3 10 14 6 7 5
6 3 11 4 13 5 7 12 1 14 9 15 8 2 10
1 10 11 5 3 7 14 2 4 15 8 9 6 13 12
3 13 10 9 5 7 11 15 8 6 2 12 4 1 14
++++++++++++++++++++++++++++++

instance TA13 (20x15)

++++++++++++++++++++++++++++++
Times
91 17 4 63 67 30 87 80 95 14 17 22 1 85 41
77 77 9 77 24 8 64 6 12 13 71 76 95 8 6
92 3 12 27 58 66 99 33 7 78 96 30 54 23 88
19 45 65 24 30 30 49 32 78 31 3 25 9 2 22
84 61 35 44 37 16 97 85 51 26 13 76 41 2 96
85 55 2 65 52 97 81 8 22 59 95 52 85 64 13
64 94 4 13 98 26 32 20 97 28 63 2 23 14 62
56 98 56 28 1 96 27 38 41 94 77 63 63 81 6
63 98 64 37 89 96 88 13 72 28 57 99 11 8 96
17 71 80 33 87 82 44 14 85 2 60 72 27 63 66
47 42 61 17 65 5 96 47 9 20 10 11 86 90 65
66 91 8 37 99 90 16 89 17 98 87 8 40 33 37
99 2 22 12 13 62 30 44 25 56 10 44 25 39 65
35 62 52 84 30 2 50 69 64 54 45 38 90 70 37
73 40 16 21 50 10 46 2 48 16 58 37 12 30 82
76 40 21 91 48 6 91 75 79 51 51 81 70 65 19
49 5 59 40 74 70 84 47 25 86 75 26 51 32 15
11 18 6 60 83 64 85 21 52 49 30 56 31 25 31
83 42 11 64 44 90 8 35 72 67 72 55 43 88 35
19 53 80 89 21 34 56 89 50 28 15 27 74 83 79
Machines
13 12 9 10 8 14 1 11 3 5 6 7 4 15 2
3 6 1 11 5 2 14 12 4 15 7 9 10 13 8
4 3 7 10 6 13 1 14 8 11 15 2 5 9 12
10 15 6 8 14 11 13 3 4 9 2 7 12 1 5
4 9 7 2 14 1 10 6 15 3 13 8 5 12 11
4 9 14 12 13 2 5 15 7 6 10 1 8 11 3
12 3 1 7 5 2 6 13 10 8 4 9 14 11 15
13 4 5 12 9 2 6 1 14 7 11 10 3 15 8
6 9 8 1 13 14 2 7 5 15 11 10 4 3 12
7 4 10 9 3 2 13 8 6 15 1 5 12 14 11
7 1 5 15 9 14 13 12 11 2 10 8 6 4 3
1 7 5 4 9 3 12 2 10 6 14 13 11 15 8
3 10 6 13 4 15 5 14 12 8 2 11 9 7 1
12 8 11 5 9 15 7 4 10 6 2 14 3 13 1
9 14 11 13 1 10 3 7 12 2 4 15 6 8 5
15 8 12 7 11 10 3 1 2 13 9 4 5 14 6
9 14 8 10 12 13 6 5 3 4 15 7 1 11 2
9 15 6 12 2 1 8 4 13 3 10 11 14 7 5
9 7 2 12 8 13 15 11 1 10 4 5 6 14 3
14 2 5 1 6 8 4 7 13 9 12 3 11 15 10
++++++++++++++++++++++++++++++

instance TA21 (20x20)

++++++++++++++++++++++++++++++
Times
64 57 81 98 59 87 93 62 20 14 85 45 47 9 94 9 15 66 1 94
39 96 88 83 77 58 83 3 78 68 64 97 33 25 47 44 7 60 42 91
96 66 88 60 22 92 62 14 89 39 94 66 10 53 26 15 65 82 10 27
93 92 96 70 83 74 31 88 51 57 78 8 7 91 79 18 51 18 99 33
4 82 40 86 50 54 21 6 54 68 82 20 39 35 68 73 23 30 30 53
94 58 93 32 91 30 56 27 92 9 78 23 21 60 36 29 95 99 79 76
93 42 52 42 96 29 61 88 70 16 31 65 83 78 26 50 87 62 14 30
18 75 20 4 91 68 19 54 85 73 43 24 37 87 66 32 52 9 49 61
35 99 62 6 62 7 80 3 57 7 85 30 96 91 13 87 82 83 78 56
85 8 66 88 15 5 59 30 60 41 17 66 89 78 88 69 45 82 6 13
90 27 1 8 91 80 89 49 32 28 90 93 6 35 73 47 43 75 8 51

3 84 34 28 60 69 45 67 58 87 65 62 97 20 31 33 33 77 50 80
48 90 75 96 44 28 21 51 75 17 89 59 56 63 18 17 30 16 7 35
57 16 42 34 37 26 68 73 5 8 12 87 83 20 97 20 85 61 9 36
63 11 45 10 33 5 41 47 9 74 33 35 78 12 22 44 8 97 10 86
33 60 21 96 69 34 94 15 23 84 16 55 50 5 59 35 12 57 11 51
72 42 4 62 15 27 16 34 8 50 85 12 48 5 25 40 81 46 67 25
83 92 25 40 21 4 43 38 60 24 3 28 86 68 55 91 97 19 73 20
28 81 46 98 46 29 96 12 71 32 64 39 16 97 99 49 75 7 79 80
71 9 11 8 4 47 93 82 6 49 7 24 92 13 86 80 34 75 35 29
Machines
7 2 16 3 20 14 17 19 4 13 15 6 11 8 9 10 1 18 12 5
9 7 11 10 20 2 1 18 3 12 8 14 19 16 5 15 6 4 17 13
2 4 5 13 1 15 8 20 11 12 6 19 14 16 17 9 10 3 18 7
19 16 11 2 20 10 3 13 5 1 14 17 15 12 4 6 9 18 7 8
3 11 13 16 8 15 18 6 9 2 14 1 4 17 5 10 12 19 7 20
1 17 11 10 18 9 6 4 20 12 16 3 15 5 13 14 2 7 8 19
12 1 2 20 10 11 9 18 19 17 5 4 13 8 7 16 6 15 3 14
17 4 9 11 19 8 2 15 20 3 13 7 6 12 18 5 14 10 16 1
18 19 16 12 11 13 15 4 2 1 17 10 7 20 8 9 5 3 14 6
10 6 4 3 12 5 17 15 8 14 2 19 9 16 13 7 18 20 1 11
6 10 19 1 4 17 7 2 3 12 16 20 8 9 5 11 13 18 15 14
12 6 10 5 4 16 18 8 19 20 14 2 7 13 11 1 15 17 3 9
16 4 5 8 19 14 20 10 3 11 12 15 7 18 13 17 1 9 2 6
14 11 17 9 5 2 3 20 15 13 19 18 1 6 10 8 7 4 12 16
1 6 5 20 15 9 2 17 16 11 13 3 12 7 8 19 14 4 18 10
12 17 18 11 16 13 1 6 5 9 4 8 14 2 19 10 20 7 15 3
8 9 6 14 20 17 12 2 15 19 11 7 4 16 5 3 1 10 18 13
18 11 16 6 5 15 9 19 14 7 20 4 2 10 12 3 13 17 8 1
15 4 17 1 11 19 5 2 6 20 13 10 12 16 14 8 18 9 7 3
14 11 4 15 2 9 1 17 20 5 7 10 12 6 3 8 18 13 16 19
+++++++++++++++++++++++++++++

instance TA22 (20x20)

+++++++++++++++++++++++++++++
Times
94 61 12 68 40 84 30 16 34 92 53 55 61 67 30 88 12 20 16 51
22 75 29 87 47 48 21 46 77 35 10 92 9 75 40 89 86 33 2 1
32 8 99 14 41 53 97 19 39 20 91 54 97 79 21 22 93 67 17 84
13 43 97 41 4 35 6 93 32 35 2 54 77 9 97 10 45 81 76 37
26 70 33 58 38 77 86 53 47 20 71 69 95 4 23 89 87 20 67 65
86 73 93 26 98 37 67 87 33 6 68 16 12 5 33 87 96 46 87 89
3 34 2 96 67 37 30 50 84 27 37 89 92 68 20 80 76 74 11 38
60 97 42 73 28 69 90 44 27 54 24 36 82 13 33 80 44 99 80 82
79 62 31 27 72 12 4 4 11 35 83 57 19 80 20 16 96 24 64 93
61 86 46 58 2 19 46 50 79 84 14 16 76 89 85 86 60 44 28 63
10 44 26 61 92 30 19 27 22 86 22 62 75 10 78 3 97 88 10 46
21 51 3 94 82 26 83 57 86 61 80 81 25 5 75 38 16 20 50 52
17 86 6 49 74 82 86 26 80 46 94 7 27 26 97 14 27 3 12 82
46 21 1 99 83 22 2 42 61 79 17 67 61 72 49 91 38 28 34 14
50 49 40 63 5 80 70 3 62 43 58 39 52 68 71 86 61 53 1 97
53 51 25 16 91 93 37 61 41 49 20 24 58 8 72 30 15 86 31 40
72 77 34 45 83 85 19 5 77 75 61 89 77 44 32 86 40 23 35 57
33 16 60 70 67 37 42 24 75 1 22 32 21 3 69 77 53 64 34 15
58 55 68 5 20 88 91 79 55 16 53 84 1 66 14 83 1 96 54 30
80 81 9 49 32 19 92 65 88 64 4 68 79 21 84 92 66 51 83 96
Machines
4 3 20 12 11 14 17 16 2 1 13 18 19 10 6 8 15 5 9 7
15 13 5 18 8 3 11 16 9 10 4 7 17 14 12 2 1 6 20 19
19 18 1 14 8 2 12 20 6 4 3 17 16 5 9 15 10 13 11 7
18 11 9 13 6 17 15 12 7 2 10 5 20 1 19 16 4 3 14 8
9 13 2 11 14 16 6 18 10 8 19 4 5 1 3 15 20 12 17 7
7 1 10 17 15 19 18 11 3 14 12 8 20 2 9 4 5 16 13 6
13 2 7 15 12 16 4 11 10 20 6 18 3 5 9 19 8 1 14 17
6 13 20 2 9 14 19 10 11 3 18 4 16 7 17 8 1 5 12 15
3 9 4 11 17 16 1 18 8 14 6 15 13 10 20 7 19 12 5 2
18 3 8 20 12 17 14 11 16 2 6 7 9 10 4 13 19 15 1 5
8 13 9 12 17 16 5 4 15 10 20 3 14 18 7 2 11 19 6 1
5 18 12 17 16 19 15 20 7 8 6 11 14 1 4 10 3 9 2 13
9 17 13 3 6 16 11 18 8 14 5 2 1 4 20 10 19 15 12 7
2 9 19 12 18 10 11 17 20 6 13 5 4 7 8 3 15 1 14 16
14 8 5 1 20 3 15 4 12 18 17 11 2 10 13 7 16 9 6 19
16 1 19 17 9 5 6 15 8 18 2 4 11 20 7 3 14 13 10 12
4 12 3 10 16 2 11 8 20 6 9 14 18 7 1 17 15 13 19 5
18 7 12 6 17 5 13 10 19 1 11 15 8 4 16 2 9 20 3 14
6 7 9 13 3 10 17 14 19 2 8 18 1 16 20 4 12 5 15 11
4 16 11 5 19 2 17 10 13 8 12 1 3 20 14 15 18 6 7 9
+++++++++++++++++++++++++++++

instance TA22 (20x20)

+++++++++++++++++++++++++++++
Times
33 8 81 68 28 91 91 74 7 7 19 50 65 53 9 90 69 50 58 13
69 10 58 10 91 5 37 9 93 94 46 55 99 28 95 94 4 51 59 10
79 70 35 82 35 84 34 87 91 69 12 31 94 65 13 16 39 46 4 74
50 40 81 47 96 67 94 53 22 17 23 24 66 15 56 84 79 25 13 72
7 81 62 50 91 77 32 10 78 78 21 78 21 10 88 23 92 34 88 48
66 71 55 25 43 24 87 59 90 63 90 22 6 50 9 18 19 52 83 66
66 39 10 80 55 38 29 41 63 32 91 27 72 71 61 35 17 26 42 64
11 33 84 12 18 57 43 24 77 85 62 49 5 46 93 85 92 30 64 77
38 30 31 25 90 79 3 52 87 30 87 4 57 43 55 21 30 1 72 75
9 49 91 39 40 59 20 27 67 22 2 47 91 11 70 97 78 69 17 40
57 32 67 26 23 55 14 77 77 82 34 1 64 90 37 47 27 54 3 94
25 33 12 27 32 49 35 5 73 3 28 54 45 32 53 99 85 86 13 99
64 77 82 32 75 32 68 16 63 81 31 58 73 12 25 64 98 72 47 84
17 98 99 39 73 82 1 43 48 62 44 50 44 72 89 45 44 21 79 60
87 63 8 20 88 88 77 88 46 30 44 42 84 41 74 52 25 87 43 77
39 93 44 23 75 7 60 45 71 49 3 68 56 20 35 8 79 21 48 43
75 92 83 48 7 99 43 94 6 34 48 60 33 16 34 99 83 11 80 43
97 80 2 37 31 37 58 11 24 84 10 30 97 89 47 37 73 11 90 54
1 97 68 8 7 72 38 50 42 32 54 94 31 52 76 20 29 56 36 16
29 31 49 91 7 37 86 75 21 46 47 1 16 29 47 81 52 44 95 79
Machines
8 5 1 11 18 19 17 16 4 12 6 7 10 20 13 9 15 14 3 2
4 20 18 15 12 2 3 8 17 14 10 6 9 1 19 11 7 5 16 13
16 1 5 9 3 15 19 13 8 6 7 17 18 12 11 14 20 10 2 4
18 8 13 10 1 14 2 15 3 17 11 19 9 5 7 16 20 6 4 12
20 7 6 16 14 5 3 19 4 17 12 15 2 18 9 10 8 11 13 1
15 1 6 19 12 3 13 8 7 14 5 20 16 2 4 11 17 18 10 9
9 15 19 8 7 4 2 12 18 11 16 20 5 3 17 14 10 1 6 13
18 19 14 16 6 9 20 17 1 3 8 4 2 10 12 15 7 11 5 13
12 1 10 19 2 20 17 13 7 11 8 15 16 9 5 14 18 6 3 4
1 9 19 5 14 15 2 8 20 16 17 11 4 7 12 6 13 18 10 3
17 7 4 9 11 2 20 18 1 14 3 5 6 13 10 16 8 19 12 15
5 8 17 11 10 1 19 2 20 9 15 3 14 12 7 6 13 16 4 18
13 14 3 19 10 9 1 11 8 4 16 18 7 17 15 6 5 12 2 20
18 19 6 9 3 5 10 16 8 11 4 13 14 17 15 2 1 7 12 20
13 8 11 3 19 5 20 15 14 4 7 2 6 17 16 9 10 1 12 18
1 4 16 2 17 14 15 8 19 11 6 12 7 20 3 18 9 10 13 5
4 16 2 5 3 12 14 18 19 15 17 1 7 9 13 6 10 20 8 11
3 18 11 2 20 13 7 9 16 15 6 10 4 19 12 14 1 5 17 8
12 4 20 2 16 15 9 7 5 17 1 6 14 11 18 3 19 13 8 10
3 18 10 11 4 6 2 1 17 15 16 13 12 9 19 5 14 20 7 8

APPENDIX B: LIST OF BENCHMARK FSSP

This appendix contains a set of 8 (CAR0l to CAR08) problems from Carlicr (1978) - CAR problems. (Note: If the table splits, continue to next page).
+++++++++++++++++++++++++++++

instance CAR01

+++++++++++++++++++++++++++++
11x5 instance from JACQUES CARLIER (Instance CAR1)
11x5

0 375 1 12 2 142 3 245 4 412
0 632 1 452 2 758 3 278.4 398
0 12 1 876 2 124 3 534 4 765
0 460 1 542 2 523 3 120 4 499
0 528 1 101 2 789 3 124 4 999
0 796 1 245 2 632 3 375 4 123
0 532 1 230 2 543 3 896 4 452
0 14 1 124 2 214 3 543 4 785
0 257 1 527 2 753 3 210 4 463
0 896 1 896 2 214 3 258 4 259
0 532 1 302 2 501 3 765 4 988

+++++++++++++++++++++++++++++

instance CAR02

+++++++++++++++++++++++++++++
13x4 instance from JACQUES CARLIER (Instance CAR2)
13x4

0 654 1 147 2 345 3 447
0 321 1 520 2 789 3 702
0 12 1 147 2 630 3 255
0 345 1 586 2 214 3 866
0 678 1 532 2 275 3 332
0 963 1 145 2 302 3 225
0 25 1 24 2 142 3 589
0 874 1 517 2 24 3 996
0 114 1 896 2 520 3 541
0 785 1 543 2 336 3 234
0 203 1 210 2 699 3 784
0 696 1 784 2 855 3 512
0 302 1 512 2 221 3 345

+++++++++++++++++++++++++++++

instance CAR03

+++++++++++++++++++++++++++++
12x5 instance from JACQUES CARLIER (Instance CAR3)
12x5

0 456 1 537 2 123 3 214 4 234
0 789 1 854 2 225 3 528 4 123
0 876 1 632 2 588 3 896 4 456
0 543 1 145 2 669 3 325 4 789
0 210 1 785 2 966 3 147 4 876
0 123 1 214 2 332 3 856 4 543
0 456 1 752 2 144 3 321 4 210
0 789 1 143 2 755 3 427 4 123
0 876 1 698 2 322 3 546 4 456
0 543 1 532 2 100 3 321 4 789
0 210 1 145 2 114 3 401 4 876
0 124 1 247 2 753 3 214 4 543

+++++++++++++++++++++++++++++

instance CAR04

+++++++++++++++++++++++++++++
14x4 instance from JACQUES CARLIER (Instance CAR4)
14x4

0 456 1 856 2 963 3 696
0 789 1 930 2 21 3 320
0 630 1 214 2 475 3 142
0 214 1 257 2 320 3 753
0 573 1 896 2 124 3 214
0 218 1 532 2 752 3 528
0 653 1 142 2 147 3 653
0 214 1 547 2 532 3 214
0 204 1 865 2 145 3 527
0 785 1 321 2 763 3 536
0 696 1 124 2 214 3 214
0 532 1 12 2 257 3 528
0 12 1 345 2 854 3 888
0 457 1 678 2 123 3 999

+++++++++++++++++++++++++++++

instance CAR05

+++++++++++++++++++++++++++++
10x6 instance from JACQUES CARLIER (Instance CAR5)
10x6

0 333 1 991 2 996 3 123 4 145 5 234
0 333 1 111 2 663 3 456 4 785 5 532
0 252 1 222 2 222 3 789 4 214 5 586
0 222 1 204 2 114 3 876 4 752 5 532
0 255 1 477 2 123 3 543 4 143 5 142
0 555 1 566 2 456 3 210 4 698 5 573
0 558 1 899 2 789 3 124 4 532 5 12
0 888 1 965 2 876 3 537 4 145 5 14
0 889 1 588 2 543 3 854 4 247 5 527
0 999 1 889 2 210 3 632 4 451 5 856

+++++++++++++++++++++++++++++

instance CAR06

+++++++++++++++++++++++++++++
8x9 instance from JACQUES CARLIER (Instance CAR6)
8x9

0 887 1 447 2 234 3 159 4 201 5 555 6 463 7 456 8 753
0 799 1 779 2 567 3 267 4 478 5 444 6 123 7 789 8 21
0 999 1 999 2 852 3 483 4 520 5 120 6 456 7 630 8 427
0 666 1 666 2 140 3 753 4 145 5 142 6 789 7 258 8 520
0 663 1 25 2 222 3 420 4 699 5 578 6 876 7 741 8 142
0 333 1 558 2 558 3 159 4 875 5 965 6 543 7 36 8 534
0 222 1 886 2 965 3 25 4 633 5 412 6 210 7 985 8 157
0 114 1 541 2 412 3 863 4 222 5 25 6 123 7 214 8 896

+++++++++++++++++++++++++++++

instance CAR07

+++++++++++++++++++++++++++++
7x7 instance from JACQUES CARLIER (Instance CAR7)
7x7

0 692 1 310 2 832 3 630 4 258 5 147 6 255
0 581 1 582 2 14 3 214 4 147 5 753 6 806
0 475 1 475 2 785 3 578 4 852 526 699
0 23 1 196 2 696 3 214 4 586 5 356 6 877
0 158 1 325 2 530 3 785 4 325 5 565 6 412
0 796 1 874 2 214 3 236 4 896 5 898 6 302
0 542 1 205 2 578 3 963 4 325 5 800 6 120

+++++++++++++++++++++++++++++

instance CAR07

+++++++++++++++++++++++++++++
8x8 instance from JACQUES CARLIER (Instance CAR8)
8x8

0 456 1 654 2 852 3 145 4 632 5 425 6 214 7 654
0 789 1 123 2 369 3 678 4 581 5 396 6 123 7 789
0 654 1 123 2 632 3 965 4 475 5 325 6 456 7 654
0 321 1 456 2 581 3 421 4 32 5 147 6 789 7 123
0 456 1 789 2 472 3 365 4 536 5 852 6 654 7 123
0 789 1 654 2 586 3 824 4 325 5 12 6 321 7 456
0 654 1 321 2 320 3 758 4 863 5 452 6 456 7 789
0 789 1 147 2 120 3 639 4 21 5 863 6 789 7 654

APPENDIX C: CASE STUDY 1 - RESULT SASM PROBLEMS

This appendix contains a results statistics for case studies.

Column 1: Shows current job number which seizes the machines Job Number

Column 2: Shows operation number of jobs in process in column 1

Column 3: Shows Arrival Time of job in column 1

Column 4: Shows Waiting Time for a job to be loaded on the machine

Column 5: Shows start time of process

Column 6: Shows Processing Time of job on machines

Column 7: Shows Machine Idle Time

Column 8: Shows Finish time

Column 9: Shows Next Machine on which finished job to be processed

(Note: If the table splits, continue to next page).

Cmax = 505
MC1 =
1 1 0 0 0 10 0 10 2
5 1 0 10 10 15 0 25 2
6 1 0 25 25 16 0 41 2
8 2 28 13 41 16 0 57 3
3 2 54 3 57 0 0 57 5
2 2 63 0 63 0 6 63 3
7 2 108 0 108 0 45 108 5
4 6 405 0 405 0 297 405 0
MC2 =
8 1 0 0 0 28 0 28 1
2 1 0 28 28 35 0 63 1
1 2 10 53 63 32 0 95 3
5 2 25 70 95 30 0 125 3
4 2 38 87 125 23 0 148 4
6 2 41 107 148 25 0 173 3
3 6 150 23 173 0 0 173 0
7 6 184 0 184 0 11 184 0
MC3 =
4 1 0 0 0 38 0 38 2
8 3 57 0 57 22 19 79 4
2 3 63 16 79 29 0 108 4
1 3 95 13 108 21 0 129 4
3 4 110 19 129 0 0 129 6
5 3 125 4 129 31 0 160 4
7 4 163 0 163 0 3 163 6
6 3 173 0 173 17 10 190 4
MC4 =
3 1 0 0 0 54 0 54 1
7 1 0 54 54 54 0 108 1
8 4 79 29 108 55 0 163 5
2 4 108 55 163 51 0 214 5
1 4 129 85 214 40 0 254 5
4 3 148 106 254 65 0 319 5
5 4 160 159 319 54 0 373 5
6 4 190 183 373 68 0 441 5
MC5 =
3 3 57 0 57 53 57 110 3
7 3 108 2 110 53 0 163 3
8 5 163 0 163 59 0 222 6
2 5 214 8 222 48 0 270 6
1 5 254 16 270 53 0 323 6
4 4 319 4 323 61 0 384 6
5 5 373 11 384 53 0 437 6
6 5 441 0 441 64 4 505 6
MC6 =
3 5 129 0 129 21 129 150 2
7 5 163 0 163 21 13 184 2
8 6 222 0 222 20 38 242 0
2 6 270 0 270 20 28 290 0
1 6 323 0 323 23 33 346 0
4 5 384 0 384 21 38 405 1
5 6 437 0 437 18 32 455 0
6 6 505 0 505 0 50 505 0
SASM JSSP 02

Cmax = 514
MC1 =
4 2 74 0 74 15 74 89 3
1 2 81 8 89 12 0 101 2
3 4 83 18 101 10 0 111 2
6 4 147 0 147 15 36 162 5
5 5 191 0 191 12 29 203 4
2 5 263 0 263 13 60 276 4
MC2 =
6 1 0 0 0 30 0 30 4
2 1 0 30 30 20 0 50 3
4 1 0 50 50 24 0 74 1
5 2 56 18 74 27 0 101 5
1 3 101 0 101 18 0 119 4
3 5 111 8 119 23 0 142 5
MC3 =
3 1 0 0 0 21 0 21 4
5 1 0 21 21 35 0 56 2
1 1 0 56 56 25 0 81 1
2 2 50 31 81 32 0 113 5
4 3 89 24 113 40 0 153 4
6 6 360 0 360 35 207 395 0
MC4 =
3 2 21 0 21 43 21 64 6
6 2 30 34 64 65 0 129 6
1 4 119 10 129 56 0 185 6
4 4 153 32 185 61 0 246 5
5 6 203 43 246 55 0 301 0
2 6 276 25 301 46 0 347 0
MC5 =
5 3 101 0 101 71 101 172 6
2 3 113 59 172 67 0 239 6
3 6 142 97 239 55 0 294 0
6 5 162 132 294 66 0 360 3
1 6 214 146 360 65 0 425 0
4 5 246 179 425 68 0 493 6
MC6 =
3 3 64 0 64 19 64 83 1
6 3 129 0 129 18 46 147 1
5 4 172 0 172 19 25 191 1
1 5 185 6 191 23 0 214 5
2 4 239 0 239 24 25 263 1
4 6 493 0 493 21 230 514 0
SASM JSSP 03

Cmax = 425
MC1 =
1 2 65 0 65 10 65 75 2
4 2 87 0 87 9 12 96 3
3 4 98 0 98 10 2 108 2
6 4 135 0 135 10 27 145 5
2 5 170 0 170 14 25 184 4
5 5 222 0 222 11 38 233 4
MC2 =
6 1 0 0 0 38 0 38 4
2 1 0 38 38 15 0 53 3
4 1 0 53 53 34 0 87 1
5 2 44 43 87 35 0 122 5
1 3 75 47 122 32 0 154 4
3 5 108 46 154 30 0 184 5
MC3 =
3 1 0 0 0 21 0 21 4
5 1 0 21 21 23 0 44 2
1 1 0 44 44 21 0 65 1
2 2 53 12 65 8 0 73 5
4 3 96 0 96 19 23 115 4
6 6 262 0 262 43 147 305 0
MC4 =
3 2 21 0 21 55 21 76 6
6 2 38 38 76 41 0 117 6
4 4 115 2 117 50 0 167 5
1 4 154 13 167 40 0 207 6
2 6 184 23 207 45 0 252 0
5 6 233 19 252 48 0 300 0
MC5 =
2 3 73 0 73 61 73 134 6
5 3 122 12 134 63 0 197 6
6 5 145 52 197 65 0 262 3
4 5 167 95 262 52 0 314 6
3 6 184 130 314 58 0 372 0
1 6 245 127 372 53 0 425 0
MC6 =
3 3 76 0 76 22 76 98 1
6 3 117 0 117 18 19 135 1
2 4 134 1 135 35 0 170 1
5 4 197 0 197 25 27 222 1
1 5 207 15 222 23 0 245 5
4 6 314 0 314 20 69 334 0
SASM FSSP 01

Cmax = 425

```
MC1 =
 1  2  65   0  65  10  65  75   2
 4  2  87   0  87   9  12  96   3
 3  4  98   0  98  10   2  108   2
 6  4  135   0  135  10  27  145   5
 2  5  170   0  170  14  25  184   4
 5  5  222   0  222  11  38  233   4
MC2 =
 6  1  0  0  0  38  0  38   4
 2  1  0  38  38  15  0  53   3
 4  1  0  53  53  34  0  87   1
 5  2  44  43  87  35  0  122   5
 1  3  75  47  122  32  0  154   4
 3  5  108  46  154  30  0  184   5
MC3 =
 3  1  0  0  0  21  0  21   4
 5  1  0  21  21  23  0  44   2
 1  1  0  44  44  21  0  65   1
 2  2  53  12  65   8  0  73   5
 4  3  96   0  96  19  23  115   4
 6  6  262   0  262  43  147  305   0
MC4 =
 3  2  21   0  21  55  21  76   6
 6  2  38  38  76  41  0  117   6
 4  4  115   2  117  50  0  167   5
 1  4  154  13  167  40  0  207   6
 2  6  184  23  207  45  0  252   0
 5  6  233  19  252  48  0  300   0
MC5 =
 2  3  73   0  73  61  73  134   6
 5  3  122  12  134  63  0  197   6
 6  5  145  52  197  65  0  262   3
 4  5  167  95  262  52  0  314   6
 3  6  184  130  314  58  0  372   0
 1  6  245  127  372  53  0  425   0
MC6 =
 3  3  76   0  76  22  76  98   1
 6  3  117   0  117  18  19  135   1
 2  4  134   1  135  35  0  170   1
 5  4  197   0  197  25  27  222   1
 1  5  207  15  222  23  0  245   5
 4  6  314   0  314  20  69  334   0
SASM FSSP 02
```

```
Cmax =  519
MC1 =
 6  1  0  0  0  10  0  10   2
 4  1  0  10  10   9  0  19   2
 3  1  0  19  19  10  0  29   2
 1  1  0  29  29  10  0  39   2
 5  1  0  39  39  11  0  50   2
 2  1  0  50  50  14  0  64   2
MC2 =
 6  2  10   0  10  38  10  48   3
 4  2  19  29  48  34  0  82   3
 3  2  29  53  82  30  0  112   3
 1  2  39  73  112  32  0  144   3
 5  2  50  94  144  35  0  179   3
 2  2  64  115  179  15  0  194   3
MC3 =
 6  3  48   0  48  43  48  91   4
 4  3  82   9  91  19  0  110   4
 3  3  112   0  112  21   2  133   4
 1  3  144   0  144  21  11  165   4
 5  3  179   0  179  23  14  202   4
 2  3  194   8  202   8  0  210   4
MC4 =
 6  4  91   0  91  41  91  132   5
 4  4  110  22  132  50  0  182   5
 3  4  133  49  182  55  0  237   5
 1  4  165  72  237  40  0  277   5
 5  4  202  75  277  48  0  325   5
 2  4  210  115  325  45  0  370   5
MC5 =
 6  5  132   0  132  65  132  197   6
 4  5  182  15  197  52  0  249   6
 3  5  237  12  249  58  0  307   6
 1  5  277  30  307  53  0  360   6
 5  5  325  35  360  63  0  423   6
 2  5  370  53  423  61  0  484   6
MC6 =
 6  6  197   0  197  18  197  215   0
 4  6  249   0  249  20  34  269   0
 3  6  307   0  307  22  38  329   0
 1  6  360   0  360  23  31  383   0
 5  6  423   0  423  25  40  448   0
 2  6  484   0  484  35  36  519   0
```

APPENDIX D: CASE STUDY 2 - PILKINGTON PLC PROBLEMS

This appendix contains a results statistics for case studies.

Column 1: Shows current job number which seizes the machines Job Number

Column 2: Shows operation number of jobs in process in column 1

Column 3: Shows Arrival Time of job in column 1

Column 4: Shows Waiting Time for a job to be loaded on the machine

Column 5: Shows start time of process

Column 6: Shows Processing Time of job on machines

Column 7: Shows Machine Idle Time

Column 8: Shows Finish time

Column 9: Shows Next Machine on which finished job to be processed

(Note: If the table splits, continue to next page).

Honda CRV 01

Cmax = 346

MC1 =

1	1	0	0	0	43	0	43	2
2	1	0	43	43	43	0	86	2
3	1	0	86	86	43	0	129	2
4	1	0	129	129	43	0	172	2
5	1	0	172	172	43	0	215	2

MC2 =

1	2	43	0	43	34	43	77	3
2	2	86	0	86	34	9	120	3
3	2	129	0	129	34	9	163	3
4	2	172	0	172	34	9	206	3
5	2	215	0	215	34	9	249	3

MC3 =

1	3	77	0	77	34	77	111	4
2	3	120	0	120	34	9	154	4
3	3	163	0	163	34	9	197	4
4	3	206	0	206	34	9	240	4
5	3	249	0	249	34	9	283	4

MC4 =

1	4	111	0	111	47	111	158	0
2	4	154	4	158	47	0	205	0
3	4	197	8	205	47	0	252	0
4	4	240	12	252	47	0	299	0
5	4	283	16	299	47	0	346	0

Honda CRV 02

Cmax = 581

MC1 =

1	1	0	0	0	43	0	43	2
2	1	0	43	43	43	0	86	2
3	1	0	86	86	43	0	129	2
4	1	0	129	129	43	0	172	2
5	1	0	172	172	43	0	215	2
6	1	0	215	215	43	0	258	2
7	1	0	258	258	43	0	301	2
8	1	0	301	301	43	0	344	2
9	1	0	344	344	43	0	387	2
10	1	0	387	387	43	0	430	2

MC2 =

1	2	43	0	43	34	43	77	3
2	2	86	0	86	34	9	120	3
3	2	129	0	129	34	9	163	3
4	2	172	0	172	34	9	206	3
5	2	215	0	215	34	9	249	3
6	2	258	0	258	34	9	292	3
7	2	301	0	301	34	9	335	3
8	2	344	0	344	34	9	378	3
9	2	387	0	387	34	9	421	3
10	2	430	0	430	34	9	464	3

MC3 =

1	3	77	0	77	34	77	111	4
2	3	120	0	120	34	9	154	4
3	3	163	0	163	34	9	197	4
4	3	206	0	206	34	9	240	4
5	3	249	0	249	34	9	283	4
6	3	292	0	292	34	9	326	4
7	3	335	0	335	34	9	369	4
8	3	378	0	378	34	9	412	4
9	3	421	0	421	34	9	455	4
10	3	464	0	464	34	9	498	4

MC4=

1	4	111	0	111	47	111	158	0
2	4	154	4	158	47	0	205	0
3	4	197	8	205	47	0	252	0
4	4	240	12	252	47	0	299	0
5	4	283	16	299	47	0	346	0
6	4	326	20	346	47	0	393	0
7	4	369	24	393	47	0	440	0
8	4	412	28	440	47	0	487	0
9	4	455	32	487	47	0	534	0
10	4	498	36	534	47	0	581	0

Honda CRV 03

Cmax = 1286

MC1 =

1	1	0	0	0	43	0	43	2
2	1	0	43	43	43	0	86	2
3	1	0	86	86	43	0	129	2
4	1	0	129	129	43	0	172	2
5	1	0	172	172	43	0	215	2
6	1	0	215	215	43	0	258	2
7	1	0	258	258	43	0	301	2
8	1	0	301	301	43	0	344	2
9	1	0	344	344	43	0	387	2
10	1	0	387	387	43	0	430	2
11	1	0	430	430	43	0	473	2
12	1	0	473	473	43	0	516	2
13	1	0	516	516	43	0	559	2
14	1	0	559	559	43	0	602	2
15	1	0	602	602	43	0	645	2
16	1	0	645	645	43	0	688	2
17	1	0	688	688	43	0	731	2
18	1	0	731	731	43	0	774	2
19	1	0	774	774	43	0	817	2
20	1	0	817	817	43	0	860	2
21	1	0	860	860	43	0	903	2
22	1	0	903	903	43	0	946	2
23	1	0	946	946	43	0	989	2
24	1	0	989	989	43	0	1032	2
25	1	0	1032	1032	43	0	1075	2

MC2 =

1	2	43	0	43	34	43	77	3
2	2	86	0	86	34	9	120	3
3	2	129	0	129	34	9	163	3
4	2	172	0	172	34	9	206	3
5	2	215	0	215	34	9	249	3
6	2	258	0	258	34	9	292	3
7	2	301	0	301	34	9	335	3
8	2	344	0	344	34	9	378	3
9	2	387	0	387	34	9	421	3
10	2	430	0	430	34	9	464	3
11	2	473	0	473	34	9	507	3
12	2	516	0	516	34	9	550	3
13	2	559	0	559	34	9	593	3
14	2	602	0	602	34	9	636	3
15	2	645	0	645	34	9	679	3
16	2	688	0	688	34	9	722	3
17	2	731	0	731	34	9	765	3
18	2	774	0	774	34	9	808	3
19	2	817	0	817	34	9	851	3
20	2	860	0	860	34	9	894	3
21	2	903	0	903	34	9	937	3
22	2	946	0	946	34	9	980	3
23	2	989	0	989	34	9	1023	3
24	2	1032	0	1032	34	9	1066	3
25	2	1075	0	1075	34	9	1109	3

MC3 =

1	3	77	0	77	34	77	111	4
2	3	120	0	120	34	9	154	4
3	3	163	0	163	34	9	197	4
4	3	206	0	206	34	9	240	4
5	3	249	0	249	34	9	283	4
6	3	292	0	292	34	9	326	4
7	3	335	0	335	34	9	369	4
8	3	378	0	378	34	9	412	4
9	3	421	0	421	34	9	455	4
10	3	464	0	464	34	9	498	4
11	3	507	0	507	34	9	541	4
12	3	550	0	550	34	9	584	4
13	3	593	0	593	34	9	627	4
14	3	636	0	636	34	9	670	4
15	3	679	0	679	34	9	713	4
16	3	722	0	722	34	9	756	4
17	3	765	0	765	34	9	799	4
18	3	808	0	808	34	9	842	4
19	3	851	0	851	34	9	885	4
20	3	894	0	894	34	9	928	4
21	3	937	0	937	34	9	971	4
22	3	980	0	980	34	9	1014	4
23	3	1023	0	1023	34	9	1057	4
24	3	1066	0	1066	34	9	1100	4
25	3	1109	0	1109	34	9	1143	4

MC4 =

1	4	111	0	111	47	111	158	0
2	4	154	4	158	47	0	205	0
3	4	197	8	205	47	0	252	0
4	4	240	12	252	47	0	299	0
5	4	283	16	299	47	0	346	0
6	4	326	20	346	47	0	393	0
7	4	369	24	393	47	0	440	0
8	4	412	28	440	47	0	487	0
9	4	455	32	487	47	0	534	0
10	4	498	36	534	47	0	581	0
11	4	541	40	581	47	0	628	0
12	4	584	44	628	47	0	675	0
13	4	627	48	675	47	0	722	0
14	4	670	52	722	47	0	769	0
15	4	713	56	769	47	0	816	0
16	4	756	60	816	47	0	863	0
17	4	799	64	863	47	0	910	0
18	4	842	68	910	47	0	957	0
19	4	885	72	957	47	0	1004	0
20	4	928	76	1004	47	0	1051	0
21	4	971	80	1051	47	0	1098	0
22	4	1014	84	1098	47	0	1145	0
23	4	1057	88	1145	47	0	1192	0
24	4	1100	92	1192	47	0	1239	0
25	4	1143	96	1239	47	0	1286	0

Honda CRV 04

Cmax = 2461

MC1 =

1	1	0	0	0	43	0	43	2
2	1	0	43	43	43	0	86	2

3	1	0	86	86	43	0	129	2
4	1	0	129	129	43	0	172	2
5	1	0	172	172	43	0	215	2
6	1	0	215	215	43	0	258	2
7	1	0	258	258	43	0	301	2
8	1	0	301	301	43	0	344	2
9	1	0	344	344	43	0	387	2
10	1	0	387	387	43	0	430	2
11	1	0	430	430	43	0	473	2
12	1	0	473	473	43	0	516	2
13	1	0	516	516	43	0	559	2
14	1	0	559	559	43	0	602	2
15	1	0	602	602	43	0	645	2
16	1	0	645	645	43	0	688	2
17	1	0	688	688	43	0	731	2
18	1	0	731	731	43	0	774	2
19	1	0	774	774	43	0	817	2
20	1	0	817	817	43	0	860	2
21	1	0	860	860	43	0	903	2
22	1	0	903	903	43	0	946	2
23	1	0	946	946	43	0	989	2
24	1	0	989	989	43	0	1032	2
25	1	0	1032	1032	43	0	1075	2
26	1	0	1075	1075	43	0	1118	2
27	1	0	1118	1118	43	0	1161	2
28	1	0	1161	1161	43	0	1204	2
29	1	0	1204	1204	43	0	1247	2
30	1	0	1247	1247	43	0	1290	2
31	1	0	1290	1290	43	0	1333	2
32	1	0	1333	1333	43	0	1376	2
33	1	0	1376	1376	43	0	1419	2
34	1	0	1419	1419	43	0	1462	2
35	1	0	1462	1462	43	0	1505	2
36	1	0	1505	1505	43	0	1548	2
37	1	0	1548	1548	43	0	1591	2
38	1	0	1591	1591	43	0	1634	2
39	1	0	1634	1634	43	0	1677	2
40	1	0	1677	1677	43	0	1720	2
41	1	0	1720	1720	43	0	1763	2
42	1	0	1763	1763	43	0	1806	2
43	1	0	1806	1806	43	0	1849	2
44	1	0	1849	1849	43	0	1892	2
45	1	0	1892	1892	43	0	1935	2
46	1	0	1935	1935	43	0	1978	2
47	1	0	1978	1978	43	0	2021	2
48	1	0	2021	2021	43	0	2064	2
49	1	0	2064	2064	43	0	2107	2
50	1	0	2107	2107	43	0	2150	2

MC2 =

1	2	43	0	43	34	43	77	3
2	2	86	0	86	34	9	120	3
3	2	129	0	129	34	9	163	3
4	2	172	0	172	34	9	206	3
5	2	215	0	215	34	9	249	3
6	2	258	0	258	34	9	292	3
7	2	301	0	301	34	9	335	3
8	2	344	0	344	34	9	378	3
9	2	387	0	387	34	9	421	3
10	2	430	0	430	34	9	464	3
11	2	473	0	473	34	9	507	3
12	2	516	0	516	34	9	550	3
13	2	559	0	559	34	9	593	3
14	2	602	0	602	34	9	636	3
15	2	645	0	645	34	9	679	3
16	2	688	0	688	34	9	722	3
17	2	731	0	731	34	9	765	3
18	2	774	0	774	34	9	808	3
19	2	817	0	817	34	9	851	3
20	2	860	0	860	34	9	894	3
21	2	903	0	903	34	9	937	3
22	2	946	0	946	34	9	980	3
23	2	989	0	989	34	9	1023	3
24	2	1032	0	1032	34	9	1066	3
25	2	1075	0	1075	34	9	1109	3
26	2	1118	0	1118	34	9	1152	3
27	2	1161	0	1161	34	9	1195	3
28	2	1204	0	1204	34	9	1238	3
29	2	1247	0	1247	34	9	1281	3
30	2	1290	0	1290	34	9	1324	3
31	2	1333	0	1333	34	9	1367	3
32	2	1376	0	1376	34	9	1410	3
33	2	1419	0	1419	34	9	1453	3
34	2	1462	0	1462	34	9	1496	3
35	2	1505	0	1505	34	9	1539	3
36	2	1548	0	1548	34	9	1582	3
37	2	1591	0	1591	34	9	1625	3
38	2	1634	0	1634	34	9	1668	3
39	2	1677	0	1677	34	9	1711	3
40	2	1720	0	1720	34	9	1754	3
41	2	1763	0	1763	34	9	1797	3
42	2	1806	0	1806	34	9	1840	3
43	2	1849	0	1849	34	9	1883	3
44	2	1892	0	1892	34	9	1926	3
45	2	1935	0	1935	34	9	1969	3
46	2	1978	0	1978	34	9	2012	3
47	2	2021	0	2021	34	9	2055	3
48	2	2064	0	2064	34	9	2098	3
49	2	2107	0	2107	34	9	2141	3
50	2	2150	0	2150	34	9	2184	3

MC3 =

1	3	77	0	77	34	77	111	4
2	3	120	0	120	34	9	154	4
3	3	163	0	163	34	9	197	4
4	3	206	0	206	34	9	240	4
5	3	249	0	249	34	9	283	4
6	3	292	0	292	34	9	326	4
7	3	335	0	335	34	9	369	4
8	3	378	0	378	34	9	412	4
9	3	421	0	421	34	9	455	4
10	3	464	0	464	34	9	498	4
11	3	507	0	507	34	9	541	4
12	3	550	0	550	34	9	584	4
13	3	593	0	593	34	9	627	4
14	3	636	0	636	34	9	670	4
15	3	679	0	679	34	9	713	4
16	3	722	0	722	34	9	756	4
17	3	765	0	765	34	9	799	4
18	3	808	0	808	34	9	842	4
19	3	851	0	851	34	9	885	4
20	3	894	0	894	34	9	928	4
21	3	937	0	937	34	9	971	4
22	3	980	0	980	34	9	1014	4
23	3	1023	0	1023	34	9	1057	4
24	3	1066	0	1066	34	9	1100	4
25	3	1109	0	1109	34	9	1143	4
26	3	1152	0	1152	34	9	1186	4
27	3	1195	0	1195	34	9	1229	4
28	3	1238	0	1238	34	9	1272	4
29	3	1281	0	1281	34	9	1315	4
30	3	1324	0	1324	34	9	1358	4
31	3	1367	0	1367	34	9	1401	4
32	3	1410	0	1410	34	9	1444	4
33	3	1453	0	1453	34	9	1487	4
34	3	1496	0	1496	34	9	1530	4
35	3	1539	0	1539	34	9	1573	4
36	3	1582	0	1582	34	9	1616	4
37	3	1625	0	1625	34	9	1659	4
38	3	1668	0	1668	34	9	1702	4
39	3	1711	0	1711	34	9	1745	4
40	3	1754	0	1754	34	9	1788	4
41	3	1797	0	1797	34	9	1831	4
42	3	1840	0	1840	34	9	1874	4
43	3	1883	0	1883	34	9	1917	4
44	3	1926	0	1926	34	9	1960	4
45	3	1969	0	1969	34	9	2003	4
46	3	2012	0	2012	34	9	2046	4
47	3	2055	0	2055	34	9	2089	4
48	3	2098	0	2098	34	9	2132	4
49	3	2141	0	2141	34	9	2175	4
50	3	2184	0	2184	34	9	2218	4

MC4 =

1	4	111	0	111	47	111	158	0
2	4	154	4	158	47	0	205	0
3	4	197	8	205	47	0	252	0
4	4	240	12	252	47	0	299	0
5	4	283	16	299	47	0	346	0
6	4	326	20	346	47	0	393	0
7	4	369	24	393	47	0	440	0
8	4	412	28	440	47	0	487	0
9	4	455	32	487	47	0	534	0
10	4	498	36	534	47	0	581	0
11	4	541	40	581	47	0	628	0
12	4	584	44	628	47	0	675	0
13	4	627	48	675	47	0	722	0
14	4	670	52	722	47	0	769	0
15	4	713	56	769	47	0	816	0
16	4	756	60	816	47	0	863	0
17	4	799	64	863	47	0	910	0
18	4	842	68	910	47	0	957	0
19	4	885	72	957	47	0	1004	0
20	4	928	76	1004	47	0	1051	0
21	4	971	80	1051	47	0	1098	0
22	4	1014	84	1098	47	0	1145	0
23	4	1057	88	1145	47	0	1192	0
24	4	1100	92	1192	47	0	1239	0
25	4	1143	96	1239	47	0	1286	0
26	4	1186	100	1286	47	0	1333	0
27	4	1229	104	1333	47	0	1380	0
28	4	1272	108	1380	47	0	1427	0
29	4	1315	112	1427	47	0	1474	0
30	4	1358	116	1474	47	0	1521	0
31	4	1401	120	1521	47	0	1568	0
32	4	1444	124	1568	47	0	1615	0
33	4	1487	128	1615	47	0	1662	0
34	4	1530	132	1662	47	0	1709	0
35	4	1573	136	1709	47	0	1756	0
36	4	1616	140	1756	47	0	1803	0
37	4	1659	144	1803	47	0	1850	0
38	4	1702	148	1850	47	0	1897	0
39	4	1745	152	1897	47	0	1944	0
40	4	1788	156	1944	47	0	1991	0
41	4	1831	160	1991	47	0	2038	0
42	4	1874	164	2038	47	0	2085	0
43	4	1917	168	2085	47	0	2132	0
44	4	1960	172	2132	47	0	2179	0
45	4	2003	176	2179	47	0	2226	0
46	4	2046	180	2226	47	0	2273	0
47	4	2089	184	2273	47	0	2320	0
48	4	2132	188	2320	47	0	2367	0
49	4	2175	192	2367	47	0	2414	0
50	4	2218	196	2414	47	0	2461	0

Honda CRV 05

Cmax = 3636

MC1 =

1	1	0	0	0	43	0	43	2
2	1	0	43	43	43	0	86	2
3	1	0	86	86	43	0	129	2
4	1	0	129	129	43	0	172	2
5	1	0	172	172	43	0	215	2
6	1	0	215	215	43	0	258	2
7	1	0	258	258	43	0	301	2
8	1	0	301	301	43	0	344	2
9	1	0	344	344	43	0	387	2
10	1	0	387	387	43	0	430	2
11	1	0	430	430	43	0	473	2
12	1	0	473	473	43	0	516	2
13	1	0	516	516	43	0	559	2
14	1	0	559	559	43	0	602	2
15	1	0	602	602	43	0	645	2
16	1	0	645	645	43	0	688	2
17	1	0	688	688	43	0	731	2
18	1	0	731	731	43	0	774	2
19	1	0	774	774	43	0	817	2
20	1	0	817	817	43	0	860	2
21	1	0	860	860	43	0	903	2
22	1	0	903	903	43	0	946	2
23	1	0	946	946	43	0	989	2
24	1	0	989	989	43	0	1032	2
25	1	0	1032	1032	43	0	1075	2
26	1	0	1075	1075	43	0	1118	2
27	1	0	1118	1118	43	0	1161	2
28	1	0	1161	1161	43	0	1204	2
29	1	0	1204	1204	43	0	1247	2
30	1	0	1247	1247	43	0	1290	2
31	1	0	1290	1290	43	0	1333	2
32	1	0	1333	1333	43	0	1376	2
33	1	0	1376	1376	43	0	1419	2
34	1	0	1419	1419	43	0	1462	2
35	1	0	1462	1462	43	0	1505	2
36	1	0	1505	1505	43	0	1548	2
37	1	0	1548	1548	43	0	1591	2
38	1	0	1591	1591	43	0	1634	2
39	1	0	1634	1634	43	0	1677	2
40	1	0	1677	1677	43	0	1720	2
41	1	0	1720	1720	43	0	1763	2
42	1	0	1763	1763	43	0	1806	2
43	1	0	1806	1806	43	0	1849	2
44	1	0	1849	1849	43	0	1892	2
45	1	0	1892	1892	43	0	1935	2
46	1	0	1935	1935	43	0	1978	2
47	1	0	1978	1978	43	0	2021	2
48	1	0	2021	2021	43	0	2064	2
49	1	0	2064	2064	43	0	2107	2
50	1	0	2107	2107	43	0	2150	2
51	1	0	2150	2150	43	0	2193	2
52	1	0	2193	2193	43	0	2236	2
53	1	0	2236	2236	43	0	2279	2
54	1	0	2279	2279	43	0	2322	2
55	1	0	2322	2322	43	0	2365	2
56	1	0	2365	2365	43	0	2408	2
57	1	0	2408	2408	43	0	2451	2
58	1	0	2451	2451	43	0	2494	2
59	1	0	2494	2494	43	0	2537	2
60	1	0	2537	2537	43	0	2580	2
61	1	0	2580	2580	43	0	2623	2
62	1	0	2623	2623	43	0	2666	2
63	1	0	2666	2666	43	0	2709	2
64	1	0	2709	2709	43	0	2752	2
65	1	0	2752	2752	43	0	2795	2
66	1	0	2795	2795	43	0	2838	2
67	1	0	2838	2838	43	0	2881	2
68	1	0	2881	2881	43	0	2924	2
69	1	0	2924	2924	43	0	2967	2
70	1	0	2967	2967	43	0	3010	2
71	1	0	3010	3010	43	0	3053	2
72	1	0	3053	3053	43	0	3096	2
73	1	0	3096	3096	43	0	3139	2
74	1	0	3139	3139	43	0	3182	2
75	1	0	3182	3182	43	0	3225	2

MC2

1	2	43	0	43	34	43	77	3
2	2	86	0	86	34	9	120	3
3	2	129	0	129	34	9	163	3
4	2	172	0	172	34	9	206	3
5	2	215	0	215	34	9	249	3
6	2	258	0	258	34	9	292	3
7	2	301	0	301	34	9	335	3
8	2	344	0	344	34	9	378	3
9	2	387	0	387	34	9	421	3
10	2	430	0	430	34	9	464	3
11	2	473	0	473	34	9	507	3
12	2	516	0	516	34	9	550	3
13	2	559	0	559	34	9	593	3
14	2	602	0	602	34	9	636	3
15	2	645	0	645	34	9	679	3
16	2	688	0	688	34	9	722	3
17	2	731	0	731	34	9	765	3
18	2	774	0	774	34	9	808	3
19	2	817	0	817	34	9	851	3
20	2	860	0	860	34	9	894	3
21	2	903	0	903	34	9	937	3
22	2	946	0	946	34	9	980	3
23	2	989	0	989	34	9	1023	3
24	2	1032	0	1032	34	9	1066	3
25	2	1075	0	1075	34	9	1109	3
26	2	1118	0	1118	34	9	1152	3
27	2	1161	0	1161	34	9	1195	3
28	2	1204	0	1204	34	9	1238	3
29	2	1247	0	1247	34	9	1281	3
30	2	1290	0	1290	34	9	1324	3
31	2	1333	0	1333	34	9	1367	3
32	2	1376	0	1376	34	9	1410	3
33	2	1419	0	1419	34	9	1453	3
34	2	1462	0	1462	34	9	1496	3
35	2	1505	0	1505	34	9	1539	3
36	2	1548	0	1548	34	9	1582	3
37	2	1591	0	1591	34	9	1625	3
38	2	1634	0	1634	34	9	1668	3
39	2	1677	0	1677	34	9	1711	3
40	2	1720	0	1720	34	9	1754	3
41	2	1763	0	1763	34	9	1797	3
42	2	1806	0	1806	34	9	1840	3
43	2	1849	0	1849	34	9	1883	3
44	2	1892	0	1892	34	9	1926	3
45	2	1935	0	1935	34	9	1969	3
46	2	1978	0	1978	34	9	2012	3
47	2	2021	0	2021	34	9	2055	3
48	2	2064	0	2064	34	9	2098	3
49	2	2107	0	2107	34	9	2141	3
50	2	2150	0	2150	34	9	2184	3
51	2	2193	0	2193	34	9	2227	3
52	2	2236	0	2236	34	9	2270	3
53	2	2279	0	2279	34	9	2313	3
54	2	2322	0	2322	34	9	2356	3
55	2	2365	0	2365	34	9	2399	3
56	2	2408	0	2408	34	9	2442	3
57	2	2451	0	2451	34	9	2485	3
58	2	2494	0	2494	34	9	2528	3
59	2	2537	0	2537	34	9	2571	3
60	2	2580	0	2580	34	9	2614	3
61	2	2623	0	2623	34	9	2657	3
62	2	2666	0	2666	34	9	2700	3
63	2	2709	0	2709	34	9	2743	3
64	2	2752	0	2752	34	9	2786	3
65	2	2795	0	2795	34	9	2829	3
66	2	2838	0	2838	34	9	2872	3
67	2	2881	0	2881	34	9	2915	3
68	2	2924	0	2924	34	9	2958	3
69	2	2967	0	2967	34	9	3001	3
70	2	3010	0	3010	34	9	3044	3
71	2	3053	0	3053	34	9	3087	3
72	2	3096	0	3096	34	9	3130	3
73	2	3139	0	3139	34	9	3173	3
74	2	3182	0	3182	34	9	3216	3
75	2	3225	0	3225	34	9	3259	3

MC3 =

1	3	77	0	77	34	77	111	4
2	3	120	0	120	34	9	154	4
3	3	163	0	163	34	9	197	4
4	3	206	0	206	34	9	240	4
5	3	249	0	249	34	9	283	4
6	3	292	0	292	34	9	326	4
7	3	335	0	335	34	9	369	4
8	3	378	0	378	34	9	412	4
9	3	421	0	421	34	9	455	4
10	3	464	0	464	34	9	498	4
11	3	507	0	507	34	9	541	4
12	3	550	0	550	34	9	584	4
13	3	593	0	593	34	9	627	4
14	3	636	0	636	34	9	670	4
15	3	679	0	679	34	9	713	4
16	3	722	0	722	34	9	756	4
17	3	765	0	765	34	9	799	4
18	3	808	0	808	34	9	842	4
19	3	851	0	851	34	9	885	4
20	3	894	0	894	34	9	928	4
21	3	937	0	937	34	9	971	4
22	3	980	0	980	34	9	1014	4
23	3	1023	0	1023	34	9	1057	4
24	3	1066	0	1066	34	9	1100	4
25	3	1109	0	1109	34	9	1143	4
26	3	1152	0	1152	34	9	1186	4
27	3	1195	0	1195	34	9	1229	4
28	3	1238	0	1238	34	9	1272	4
29	3	1281	0	1281	34	9	1315	4
30	3	1324	0	1324	34	9	1358	4
31	3	1367	0	1367	34	9	1401	4
32	3	1410	0	1410	34	9	1444	4
33	3	1453	0	1453	34	9	1487	4
34	3	1496	0	1496	34	9	1530	4
35	3	1539	0	1539	34	9	1573	4
36	3	1582	0	1582	34	9	1616	4
37	3	1625	0	1625	34	9	1659	4
38	3	1668	0	1668	34	9	1702	4
39	3	1711	0	1711	34	9	1745	4
40	3	1754	0	1754	34	9	1788	4
41	3	1797	0	1797	34	9	1831	4
42	3	1840	0	1840	34	9	1874	4
43	3	1883	0	1883	34	9	1917	4
44	3	1926	0	1926	34	9	1960	4
45	3	1969	0	1969	34	9	2003	4
46	3	2012	0	2012	34	9	2046	4
47	3	2055	0	2055	34	9	2089	4
48	3	2098	0	2098	34	9	2132	4

49	3	2141	0	2141	34	9	2175	4
50	3	2184	0	2184	34	9	2218	4
51	3	2227	0	2227	34	9	2261	4
52	3	2270	0	2270	34	9	2304	4
53	3	2313	0	2313	34	9	2347	4
54	3	2356	0	2356	34	9	2390	4
55	3	2399	0	2399	34	9	2433	4
56	3	2442	0	2442	34	9	2476	4
57	3	2485	0	2485	34	9	2519	4
58	3	2528	0	2528	34	9	2562	4
59	3	2571	0	2571	34	9	2605	4
60	3	2614	0	2614	34	9	2648	4
61	3	2657	0	2657	34	9	2691	4
62	3	2700	0	2700	34	9	2734	4
63	3	2743	0	2743	34	9	2777	4
64	3	2786	0	2786	34	9	2820	4
65	3	2829	0	2829	34	9	2863	4
66	3	2872	0	2872	34	9	2906	4
67	3	2915	0	2915	34	9	2949	4
68	3	2958	0	2958	34	9	2992	4
69	3	3001	0	3001	34	9	3035	4
70	3	3044	0	3044	34	9	3078	4
71	3	3087	0	3087	34	9	3121	4
72	3	3130	0	3130	34	9	3164	4
73	3	3173	0	3173	34	9	3207	4
74	3	3216	0	3216	34	9	3250	4
75	3	3259	0	3259	34	9	3293	4

MC4 =

1	4	111	0	111	47	111	158	0
2	4	154	4	158	47	0	205	0
3	4	197	8	205	47	0	252	0
4	4	240	12	252	47	0	299	0
5	4	283	16	299	47	0	346	0
6	4	326	20	346	47	0	393	0
7	4	369	24	393	47	0	440	0
8	4	412	28	440	47	0	487	0
9	4	455	32	487	47	0	534	0
10	4	498	36	534	47	0	581	0
11	4	541	40	581	47	0	628	0
12	4	584	44	628	47	0	675	0
13	4	627	48	675	47	0	722	0
14	4	670	52	722	47	0	769	0
15	4	713	56	769	47	0	816	0
16	4	756	60	816	47	0	863	0
17	4	799	64	863	47	0	910	0
18	4	842	68	910	47	0	957	0
19	4	885	72	957	47	0	1004	0
20	4	928	76	1004	47	0	1051	0
21	4	971	80	1051	47	0	1098	0
22	4	1014	84	1098	47	0	1145	0
23	4	1057	88	1145	47	0	1192	0
24	4	1100	92	1192	47	0	1239	0
25	4	1143	96	1239	47	0	1286	0
26	4	1186	100	1286	47	0	1333	0
27	4	1229	104	1333	47	0	1380	0
28	4	1272	108	1380	47	0	1427	0
29	4	1315	112	1427	47	0	1474	0
30	4	1358	116	1474	47	0	1521	0
31	4	1401	120	1521	47	0	1568	0
32	4	1444	124	1568	47	0	1615	0
33	4	1487	128	1615	47	0	1662	0
34	4	1530	132	1662	47	0	1709	0
35	4	1573	136	1709	47	0	1756	0
36	4	1616	140	1756	47	0	1803	0
37	4	1659	144	1803	47	0	1850	0
38	4	1702	148	1850	47	0	1897	0
39	4	1745	152	1897	47	0	1944	0
40	4	1788	156	1944	47	0	1991	0
41	4	1831	160	1991	47	0	2038	0
42	4	1874	164	2038	47	0	2085	0
43	4	1917	168	2085	47	0	2132	0
44	4	1960	172	2132	47	0	2179	0
45	4	2003	176	2179	47	0	2226	0
46	4	2046	180	2226	47	0	2273	0
47	4	2089	184	2273	47	0	2320	0
48	4	2132	188	2320	47	0	2367	0
49	4	2175	192	2367	47	0	2414	0
50	4	2218	196	2414	47	0	2461	0
51	4	2261	200	2461	47	0	2508	0
52	4	2304	204	2508	47	0	2555	0
53	4	2347	208	2555	47	0	2602	0
54	4	2390	212	2602	47	0	2649	0
55	4	2433	216	2649	47	0	2696	0
56	4	2476	220	2696	47	0	2743	0
57	4	2519	224	2743	47	0	2790	0
58	4	2562	228	2790	47	0	2837	0
59	4	2605	232	2837	47	0	2884	0
60	4	2648	236	2884	47	0	2931	0
61	4	2691	240	2931	47	0	2978	0
62	4	2734	244	2978	47	0	3025	0
63	4	2777	248	3025	47	0	3072	0
64	4	2820	252	3072	47	0	3119	0
65	4	2863	256	3119	47	0	3166	0
66	4	2906	260	3166	47	0	3213	0
67	4	2949	264	3213	47	0	3260	0
68	4	2992	268	3260	47	0	3307	0
69	4	3035	272	3307	47	0	3354	0
70	4	3078	276	3354	47	0	3401	0
71	4	3121	280	3401	47	0	3448	0
72	4	3164	284	3448	47	0	3495	0
73	4	3207	288	3495	47	0	3542	0
74	4	3250	292	3542	47	0	3589	0
75	4	3293	296	3589	47	0	3636	0

Honda CRV 06

Cmax = 4811

MC1 =

1	1	0	0	0	43	0	43	2
2	1	0	43	43	43	0	86	2
3	1	0	86	86	43	0	129	2
4	1	0	129	129	43	0	172	2
5	1	0	172	172	43	0	215	2
6	1	0	215	215	43	0	258	2
7	1	0	258	258	43	0	301	2
8	1	0	301	301	43	0	344	2
9	1	0	344	344	43	0	387	2
10	1	0	387	387	43	0	430	2
11	1	0	430	430	43	0	473	2
12	1	0	473	473	43	0	516	2
13	1	0	516	516	43	0	559	2
14	1	0	559	559	43	0	602	2
15	1	0	602	602	43	0	645	2
16	1	0	645	645	43	0	688	2
17	1	0	688	688	43	0	731	2
18	1	0	731	731	43	0	774	2
19	1	0	774	774	43	0	817	2
20	1	0	817	817	43	0	860	2
21	1	0	860	860	43	0	903	2
22	1	0	903	903	43	0	946	2
23	1	0	946	946	43	0	989	2
24	1	0	989	989	43	0	1032	2
25	1	0	1032	1032	43	0	1075	2
26	1	0	1075	1075	43	0	1118	2
27	1	0	1118	1118	43	0	1161	2
28	1	0	1161	1161	43	0	1204	2
29	1	0	1204	1204	43	0	1247	2
30	1	0	1247	1247	43	0	1290	2
31	1	0	1290	1290	43	0	1333	2
32	1	0	1333	1333	43	0	1376	2
33	1	0	1376	1376	43	0	1419	2
34	1	0	1419	1419	43	0	1462	2
35	1	0	1462	1462	43	0	1505	2
36	1	0	1505	1505	43	0	1548	2
37	1	0	1548	1548	43	0	1591	2
38	1	0	1591	1591	43	0	1634	2
39	1	0	1634	1634	43	0	1677	2
40	1	0	1677	1677	43	0	1720	2
41	1	0	1720	1720	43	0	1763	2
42	1	0	1763	1763	43	0	1806	2
43	1	0	1806	1806	43	0	1849	2
44	1	0	1849	1849	43	0	1892	2
45	1	0	1892	1892	43	0	1935	2
46	1	0	1935	1935	43	0	1978	2
47	1	0	1978	1978	43	0	2021	2
48	1	0	2021	2021	43	0	2064	2
49	1	0	2064	2064	43	0	2107	2
50	1	0	2107	2107	43	0	2150	2
51	1	0	2150	2150	43	0	2193	2
52	1	0	2193	2193	43	0	2236	2
53	1	0	2236	2236	43	0	2279	2
54	1	0	2279	2279	43	0	2322	2
55	1	0	2322	2322	43	0	2365	2
56	1	0	2365	2365	43	0	2408	2
57	1	0	2408	2408	43	0	2451	2
58	1	0	2451	2451	43	0	2494	2
59	1	0	2494	2494	43	0	2537	2
60	1	0	2537	2537	43	0	2580	2
61	1	0	2580	2580	43	0	2623	2
62	1	0	2623	2623	43	0	2666	2
63	1	0	2666	2666	43	0	2709	2
64	1	0	2709	2709	43	0	2752	2
65	1	0	2752	2752	43	0	2795	2
66	1	0	2795	2795	43	0	2838	2
67	1	0	2838	2838	43	0	2881	2
68	1	0	2881	2881	43	0	2924	2
69	1	0	2924	2924	43	0	2967	2
70	1	0	2967	2967	43	0	3010	2
71	1	0	3010	3010	43	0	3053	2
72	1	0	3053	3053	43	0	3096	2
73	1	0	3096	3096	43	0	3139	2
74	1	0	3139	3139	43	0	3182	2
75	1	0	3182	3182	43	0	3225	2
76	1	0	3225	3225	43	0	3268	2
77	1	0	3268	3268	43	0	3311	2
78	1	0	3311	3311	43	0	3354	2
79	1	0	3354	3354	43	0	3397	2
80	1	0	3397	3397	43	0	3440	2
81	1	0	3440	3440	43	0	3483	2
82	1	0	3483	3483	43	0	3526	2
83	1	0	3526	3526	43	0	3569	2
84	1	0	3569	3569	43	0	3612	2
85	1	0	3612	3612	43	0	3655	2
86	1	0	3655	3655	43	0	3698	2
87	1	0	3698	3698	43	0	3741	2
88	1	0	3741	3741	43	0	3784	2
89	1	0	3784	3784	43	0	3827	2
90	1	0	3827	3827	43	0	3870	2
91	1	0	3870	3870	43	0	3913	2
92	1	0	3913	3913	43	0	3956	2
93	1	0	3956	3956	43	0	3999	2
94	1	0	3999	3999	43	0	4042	2
95	1	0	4042	4042	43	0	4085	2
96	1	0	4085	4085	43	0	4128	2

97	1	0	4128	4128	43	0	4171	2
98	1	0	4171	4171	43	0	4214	2
99	1	0	4214	4214	43	0	4257	2
100	1	0	4257	4257	43	0	4300	2

MC2 =

1	2	43	0	43	34	43	77	3
2	2	86	0	86	34	9	120	3
3	2	129	0	129	34	9	163	3
4	2	172	0	172	34	9	206	3
5	2	215	0	215	34	9	249	3
6	2	258	0	258	34	9	292	3
7	2	301	0	301	34	9	335	3
8	2	344	0	344	34	9	378	3
9	2	387	0	387	34	9	421	3
10	2	430	0	430	34	9	464	3
11	2	473	0	473	34	9	507	3
12	2	516	0	516	34	9	550	3
13	2	559	0	559	34	9	593	3
14	2	602	0	602	34	9	636	3
15	2	645	0	645	34	9	679	3
16	2	688	0	688	34	9	722	3
17	2	731	0	731	34	9	765	3
18	2	774	0	774	34	9	808	3
19	2	817	0	817	34	9	851	3
20	2	860	0	860	34	9	894	3
21	2	903	0	903	34	9	937	3
22	2	946	0	946	34	9	980	3
23	2	989	0	989	34	9	1023	3
24	2	1032	0	1032	34	9	1066	3
25	2	1075	0	1075	34	9	1109	3
26	2	1118	0	1118	34	9	1152	3
27	2	1161	0	1161	34	9	1195	3
28	2	1204	0	1204	34	9	1238	3
29	2	1247	0	1247	34	9	1281	3
30	2	1290	0	1290	34	9	1324	3
31	2	1333	0	1333	34	9	1367	3
32	2	1376	0	1376	34	9	1410	3
33	2	1419	0	1419	34	9	1453	3
34	2	1462	0	1462	34	9	1496	3
35	2	1505	0	1505	34	9	1539	3
36	2	1548	0	1548	34	9	1582	3
37	2	1591	0	1591	34	9	1625	3
38	2	1634	0	1634	34	9	1668	3
39	2	1677	0	1677	34	9	1711	3
40	2	1720	0	1720	34	9	1754	3
41	2	1763	0	1763	34	9	1797	3
42	2	1806	0	1806	34	9	1840	3
43	2	1849	0	1849	34	9	1883	3
44	2	1892	0	1892	34	9	1926	3
45	2	1935	0	1935	34	9	1969	3
46	2	1978	0	1978	34	9	2012	3
47	2	2021	0	2021	34	9	2055	3
48	2	2064	0	2064	34	9	2098	3
49	2	2107	0	2107	34	9	2141	3
50	2	2150	0	2150	34	9	2184	3
51	2	2193	0	2193	34	9	2227	3
52	2	2236	0	2236	34	9	2270	3
53	2	2279	0	2279	34	9	2313	3
54	2	2322	0	2322	34	9	2356	3
55	2	2365	0	2365	34	9	2399	3
56	2	2408	0	2408	34	9	2442	3
57	2	2451	0	2451	34	9	2485	3
58	2	2494	0	2494	34	9	2528	3
59	2	2537	0	2537	34	9	2571	3
60	2	2580	0	2580	34	9	2614	3
61	2	2623	0	2623	34	9	2657	3
62	2	2666	0	2666	34	9	2700	3
63	2	2709	0	2709	34	9	2743	3
64	2	2752	0	2752	34	9	2786	3
65	2	2795	0	2795	34	9	2829	3
66	2	2838	0	2838	34	9	2872	3
67	2	2881	0	2881	34	9	2915	3
68	2	2924	0	2924	34	9	2958	3
69	2	2967	0	2967	34	9	3001	3
70	2	3010	0	3010	34	9	3044	3
71	2	3053	0	3053	34	9	3087	3
72	2	3096	0	3096	34	9	3130	3
73	2	3139	0	3139	34	9	3173	3
74	2	3182	0	3182	34	9	3216	3
75	2	3225	0	3225	34	9	3259	3
76	2	3268	0	3268	34	9	3302	3
77	2	3311	0	3311	34	9	3345	3
78	2	3354	0	3354	34	9	3388	3
79	2	3397	0	3397	34	9	3431	3
80	2	3440	0	3440	34	9	3474	3
81	2	3483	0	3483	34	9	3517	3
82	2	3526	0	3526	34	9	3560	3
83	2	3569	0	3569	34	9	3603	3
84	2	3612	0	3612	34	9	3646	3
85	2	3655	0	3655	34	9	3689	3
86	2	3698	0	3698	34	9	3732	3
87	2	3741	0	3741	34	9	3775	3
88	2	3784	0	3784	34	9	3818	3
89	2	3827	0	3827	34	9	3861	3
90	2	3870	0	3870	34	9	3904	3
91	2	3913	0	3913	34	9	3947	3
92	2	3956	0	3956	34	9	3990	3
93	2	3999	0	3999	34	9	4033	3
94	2	4042	0	4042	34	9	4076	3
95	2	4085	0	4085	34	9	4119	3
96	2	4128	0	4128	34	9	4162	3

97	2	4171	0	4171	34	9	4205	3
98	2	4214	0	4214	34	9	4248	3
99	2	4257	0	4257	34	9	4291	3
100	2	4300	0	4300	34	9	4334	3

MC3 =

1	3	77	0	77	34	77	111	4
2	3	120	0	120	34	9	154	4
3	3	163	0	163	34	9	197	4
4	3	206	0	206	34	9	240	4
5	3	249	0	249	34	9	283	4
6	3	292	0	292	34	9	326	4
7	3	335	0	335	34	9	369	4
8	3	378	0	378	34	9	412	4
9	3	421	0	421	34	9	455	4
10	3	464	0	464	34	9	498	4
11	3	507	0	507	34	9	541	4
12	3	550	0	550	34	9	584	4
13	3	593	0	593	34	9	627	4
14	3	636	0	636	34	9	670	4
15	3	679	0	679	34	9	713	4
16	3	722	0	722	34	9	756	4
17	3	765	0	765	34	9	799	4
18	3	808	0	808	34	9	842	4
19	3	851	0	851	34	9	885	4
20	3	894	0	894	34	9	928	4
21	3	937	0	937	34	9	971	4
22	3	980	0	980	34	9	1014	4
23	3	1023	0	1023	34	9	1057	4
24	3	1066	0	1066	34	9	1100	4
25	3	1109	0	1109	34	9	1143	4
26	3	1152	0	1152	34	9	1186	4
27	3	1195	0	1195	34	9	1229	4
28	3	1238	0	1238	34	9	1272	4
29	3	1281	0	1281	34	9	1315	4
30	3	1324	0	1324	34	9	1358	4
31	3	1367	0	1367	34	9	1401	4
32	3	1410	0	1410	34	9	1444	4
33	3	1453	0	1453	34	9	1487	4
34	3	1496	0	1496	34	9	1530	4
35	3	1539	0	1539	34	9	1573	4
36	3	1582	0	1582	34	9	1616	4
37	3	1625	0	1625	34	9	1659	4
38	3	1668	0	1668	34	9	1702	4
39	3	1711	0	1711	34	9	1745	4
40	3	1754	0	1754	34	9	1788	4
41	3	1797	0	1797	34	9	1831	4
42	3	1840	0	1840	34	9	1874	4
43	3	1883	0	1883	34	9	1917	4
44	3	1926	0	1926	34	9	1960	4
45	3	1969	0	1969	34	9	2003	4
46	3	2012	0	2012	34	9	2046	4
47	3	2055	0	2055	34	9	2089	4
48	3	2098	0	2098	34	9	2132	4
49	3	2141	0	2141	34	9	2175	4
50	3	2184	0	2184	34	9	2218	4
51	3	2227	0	2227	34	9	2261	4
52	3	2270	0	2270	34	9	2304	4
53	3	2313	0	2313	34	9	2347	4
54	3	2356	0	2356	34	9	2390	4
55	3	2399	0	2399	34	9	2433	4
56	3	2442	0	2442	34	9	2476	4
57	3	2485	0	2485	34	9	2519	4
58	3	2528	0	2528	34	9	2562	4
59	3	2571	0	2571	34	9	2605	4
60	3	2614	0	2614	34	9	2648	4
61	3	2657	0	2657	34	9	2691	4
62	3	2700	0	2700	34	9	2734	4
63	3	2743	0	2743	34	9	2777	4
64	3	2786	0	2786	34	9	2820	4
65	3	2829	0	2829	34	9	2863	4
66	3	2872	0	2872	34	9	2906	4
67	3	2915	0	2915	34	9	2949	4
68	3	2958	0	2958	34	9	2992	4
69	3	3001	0	3001	34	9	3035	4
70	3	3044	0	3044	34	9	3078	4
71	3	3087	0	3087	34	9	3121	4
72	3	3130	0	3130	34	9	3164	4
73	3	3173	0	3173	34	9	3207	4
74	3	3216	0	3216	34	9	3250	4
75	3	3259	0	3259	34	9	3293	4
76	3	3302	0	3302	34	9	3336	4
77	3	3345	0	3345	34	9	3379	4
78	3	3388	0	3388	34	9	3422	4
79	3	3431	0	3431	34	9	3465	4
80	3	3474	0	3474	34	9	3508	4
81	3	3517	0	3517	34	9	3551	4
82	3	3560	0	3560	34	9	3594	4
83	3	3603	0	3603	34	9	3637	4
84	3	3646	0	3646	34	9	3680	4
85	3	3689	0	3689	34	9	3723	4
86	3	3732	0	3732	34	9	3766	4
87	3	3775	0	3775	34	9	3809	4
88	3	3818	0	3818	34	9	3852	4
89	3	3861	0	3861	34	9	3895	4
90	3	3904	0	3904	34	9	3938	4
91	3	3947	0	3947	34	9	3981	4
92	3	3990	0	3990	34	9	4024	4
93	3	4033	0	4033	34	9	4067	4
94	3	4076	0	4076	34	9	4110	4
95	3	4119	0	4119	34	9	4153	4
96	3	4162	0	4162	34	9	4196	4

97	3	4205	0	4205	34	9	4239	4
98	3	4248	0	4248	34	9	4282	4
99	3	4291	0	4291	34	9	4325	4
100	3	4334	0	4334	34	9	4368	4

MC4 =

1	4	111	0	111	47	111	158	0
2	4	154	4	158	47	0	205	0
3	4	197	8	205	47	0	252	0
4	4	240	12	252	47	0	299	0
5	4	283	16	299	47	0	346	0
6	4	326	20	346	47	0	393	0
7	4	369	24	393	47	0	440	0
8	4	412	28	440	47	0	487	0
9	4	455	32	487	47	0	534	0
10	4	498	36	534	47	0	581	0
11	4	541	40	581	47	0	628	0
12	4	584	44	628	47	0	675	0
13	4	627	48	675	47	0	722	0
14	4	670	52	722	47	0	769	0
15	4	713	56	769	47	0	816	0
16	4	756	60	816	47	0	863	0
17	4	799	64	863	47	0	910	0
18	4	842	68	910	47	0	957	0
19	4	885	72	957	47	0	1004	0
20	4	928	76	1004	47	0	1051	0
21	4	971	80	1051	47	0	1098	0
22	4	1014	84	1098	47	0	1145	0
23	4	1057	88	1145	47	0	1192	0
24	4	1100	92	1192	47	0	1239	0
25	4	1143	96	1239	47	0	1286	0
26	4	1186	100	1286	47	0	1333	0
27	4	1229	104	1333	47	0	1380	0
28	4	1272	108	1380	47	0	1427	0
29	4	1315	112	1427	47	0	1474	0
30	4	1358	116	1474	47	0	1521	0
31	4	1401	120	1521	47	0	1568	0
32	4	1444	124	1568	47	0	1615	0
33	4	1487	128	1615	47	0	1662	0
34	4	1530	132	1662	47	0	1709	0
35	4	1573	136	1709	47	0	1756	0
36	4	1616	140	1756	47	0	1803	0
37	4	1659	144	1803	47	0	1850	0
38	4	1702	148	1850	47	0	1897	0
39	4	1745	152	1897	47	0	1944	0
40	4	1788	156	1944	47	0	1991	0
41	4	1831	160	1991	47	0	2038	0
42	4	1874	164	2038	47	0	2085	0
43	4	1917	168	2085	47	0	2132	0
44	4	1960	172	2132	47	0	2179	0
45	4	2003	176	2179	47	0	2226	0
46	4	2046	180	2226	47	0	2273	0
47	4	2089	184	2273	47	0	2320	0
48	4	2132	188	2320	47	0	2367	0
49	4	2175	192	2367	47	0	2414	0
50	4	2218	196	2414	47	0	2461	0
51	4	2261	200	2461	47	0	2508	0
52	4	2304	204	2508	47	0	2555	0
53	4	2347	208	2555	47	0	2602	0
54	4	2390	212	2602	47	0	2649	0
55	4	2433	216	2649	47	0	2696	0
56	4	2476	220	2696	47	0	2743	0
57	4	2519	224	2743	47	0	2790	0
58	4	2562	228	2790	47	0	2837	0
59	4	2605	232	2837	47	0	2884	0
60	4	2648	236	2884	47	0	2931	0
61	4	2691	240	2931	47	0	2978	0
62	4	2734	244	2978	47	0	3025	0
63	4	2777	248	3025	47	0	3072	0
64	4	2820	252	3072	47	0	3119	0
65	4	2863	256	3119	47	0	3166	0
66	4	2906	260	3166	47	0	3213	0
67	4	2949	264	3213	47	0	3260	0
68	4	2992	268	3260	47	0	3307	0
69	4	3035	272	3307	47	0	3354	0
70	4	3078	276	3354	47	0	3401	0
71	4	3121	280	3401	47	0	3448	0
72	4	3164	284	3448	47	0	3495	0
73	4	3207	288	3495	47	0	3542	0
74	4	3250	292	3542	47	0	3589	0
75	4	3293	296	3589	47	0	3636	0
76	4	3336	300	3636	47	0	3683	0
77	4	3379	304	3683	47	0	3730	0
78	4	3422	308	3730	47	0	3777	0
79	4	3465	312	3777	47	0	3824	0
80	4	3508	316	3824	47	0	3871	0
81	4	3551	320	3871	47	0	3918	0
82	4	3594	324	3918	47	0	3965	0
83	4	3637	328	3965	47	0	4012	0
84	4	3680	332	4012	47	0	4059	0
85	4	3723	336	4059	47	0	4106	0
86	4	3766	340	4106	47	0	4153	0
87	4	3809	344	4153	47	0	4200	0
88	4	3852	348	4200	47	0	4247	0
89	4	3895	352	4247	47	0	4294	0
90	4	3938	356	4294	47	0	4341	0
91	4	3981	360	4341	47	0	4388	0
92	4	4024	364	4388	47	0	4435	0
93	4	4067	368	4435	47	0	4482	0
94	4	4110	372	4482	47	0	4529	0
95	4	4153	376	4529	47	0	4576	0
96	4	4196	380	4576	47	0	4623	0
97	4	4239	384	4623	47	0	4670	0
98	4	4282	388	4670	47	0	4717	0
99	4	4325	392	4717	47	0	4764	0
100	4	4368	396	4764	47	0	4811	0

HHR (AGVR) Rover 400-1

Cmax = 364

MC1 =

1	1	0	0	0	34	0	34	2
2	1	0	34	34	34	0	68	2
3	1	0	68	68	34	0	102	2
4	1	0	102	102	34	0	136	2
5	1	0	136	136	34	0	170	2

MC2 =

1	2	34	0	34	36	34	70	3
2	2	68	2	70	36	0	106	3
3	2	102	4	106	36	0	142	3
4	2	136	6	142	36	0	178	3
5	2	170	8	178	36	0	214	3

MC3 =

1	3	70	0	70	37	70	107	4
2	3	106	1	107	37	0	144	4
3	3	142	2	144	37	0	181	4
4	3	178	3	181	37	0	218	4
5	3	214	4	218	37	0	255	4

MC4 =

1	4	107	0	107	43	107	150	5
2	4	144	6	150	43	0	193	5
3	4	181	12	193	43	0	236	5
4	4	218	18	236	43	0	279	5
5	4	255	24	279	43	0	322	5

MC5 =

1	5	150	0	150	42	150	192	0
2	5	193	0	193	42	1	235	0
3	5	236	0	236	42	1	278	0
4	5	279	0	279	42	1	321	0
5	5	322	0	322	42	1	364	0

HHR (AGVR) Rover 400-2

Cmax = 579

MC1 =

1	1	0	0	0	34	0	34	2
2	1	0	34	34	34	0	68	2
3	1	0	68	68	34	0	102	2
4	1	0	102	102	34	0	136	2
5	1	0	136	136	34	0	170	2
6	1	0	170	170	34	0	204	2
7	1	0	204	204	34	0	238	2
8	1	0	238	238	34	0	272	2
9	1	0	272	272	34	0	306	2
10	1	0	306	306	34	0	340	2

MC2 =

1	2	34	0	34	36	34	70	3
2	2	68	2	70	36	0	106	3
3	2	102	4	106	36	0	142	3
4	2	136	6	142	36	0	178	3
5	2	170	8	178	36	0	214	3
6	2	204	10	214	36	0	250	3
7	2	238	12	250	36	0	286	3
8	2	272	14	286	36	0	322	3
9	2	306	16	322	36	0	358	3
10	2	340	18	358	36	0	394	3

MC3 =

1	3	70	0	70	37	70	107	4
2	3	106	1	107	37	0	144	4
3	3	142	2	144	37	0	181	4
4	3	178	3	181	37	0	218	4
5	3	214	4	218	37	0	255	4
6	3	250	5	255	37	0	292	4
7	3	286	6	292	37	0	329	4
8	3	322	7	329	37	0	366	4
9	3	358	8	366	37	0	403	4
10	3	394	9	403	37	0	440	4

MC4 =

1	4	107	0	107	43	107	150	5
2	4	144	6	150	43	0	193	5
3	4	181	12	193	43	0	236	5
4	4	218	18	236	43	0	279	5
5	4	255	24	279	43	0	322	5
6	4	292	30	322	43	0	365	5
7	4	329	36	365	43	0	408	5
8	4	366	42	408	43	0	451	5
9	4	403	48	451	43	0	494	5
10	4	440	54	494	43	0	537	5

MC5 =

1	5	150	0	150	42	150	192	0
2	5	193	0	193	42	1	235	0
3	5	236	0	236	42	1	278	0
4	5	279	0	279	42	1	321	0
5	5	322	0	322	42	1	364	0
6	5	365	0	365	42	1	407	0
7	5	408	0	408	42	1	450	0
8	5	451	0	451	42	1	493	0
9	5	494	0	494	42	1	536	0
10	5	537	0	537	42	1	579	0

HHR (AGVR) Rover 400-3

Cmax = 1224

MC1 =

1	1	0	0	0	34	0	34	2
2	1	0	34	34	34	0	68	2
3	1	0	68	68	34	0	102	2
4	1	0	102	102	34	0	136	2
5	1	0	136	136	34	0	170	2

6	1	0	170	170	34	0	204	2
7	1	0	204	204	34	0	238	2
8	1	0	238	238	34	0	272	2
9	1	0	272	272	34	0	306	2
10	1	0	306	306	34	0	340	2
11	1	0	340	340	34	0	374	2
12	1	0	374	374	34	0	408	2
13	1	0	408	408	34	0	442	2
14	1	0	442	442	34	0	476	2
15	1	0	476	476	34	0	510	2
16	1	0	510	510	34	0	544	2
17	1	0	544	544	34	0	578	2
18	1	0	578	578	34	0	612	2
19	1	0	612	612	34	0	646	2
20	1	0	646	646	34	0	680	2
21	1	0	680	680	34	0	714	2
22	1	0	714	714	34	0	748	2
23	1	0	748	748	34	0	782	2
24	1	0	782	782	34	0	816	2
25	1	0	816	816	34	0	850	2

MC2 =

1	2	34	0	34	36	34	70	3
2	2	68	2	70	36	0	106	3
3	2	102	4	106	36	0	142	3
4	2	136	6	142	36	0	178	3
5	2	170	8	178	36	0	214	3
6	2	204	10	214	36	0	250	3
7	2	238	12	250	36	0	286	3
8	2	272	14	286	36	0	322	3
9	2	306	16	322	36	0	358	3
10	2	340	18	358	36	0	394	3
11	2	374	20	394	36	0	430	3
12	2	408	22	430	36	0	466	3
13	2	442	24	466	36	0	502	3
14	2	476	26	502	36	0	538	3
15	2	510	28	538	36	0	574	3
16	2	544	30	574	36	0	610	3
17	2	578	32	610	36	0	646	3
18	2	612	34	646	36	0	682	3
19	2	646	36	682	36	0	718	3
20	2	680	38	718	36	0	754	3
21	2	714	40	754	36	0	790	3
22	2	748	42	790	36	0	826	3
23	2	782	44	826	36	0	862	3
24	2	816	46	862	36	0	898	3
25	2	850	48	898	36	0	934	3

MC3 =

1	3	70	0	70	37	70	107	4
2	3	106	1	107	37	0	144	4
3	3	142	2	144	37	0	181	4
4	3	178	3	181	37	0	218	4
5	3	214	4	218	37	0	255	4
6	3	250	5	255	37	0	292	4
7	3	286	6	292	37	0	329	4
8	3	322	7	329	37	0	366	4
9	3	358	8	366	37	0	403	4
10	3	394	9	403	37	0	440	4
11	3	430	10	440	37	0	477	4
12	3	466	11	477	37	0	514	4
13	3	502	12	514	37	0	551	4
14	3	538	13	551	37	0	588	4
15	3	574	14	588	37	0	625	4
16	3	610	15	625	37	0	662	4
17	3	646	16	662	37	0	699	4
18	3	682	17	699	37	0	736	4
19	3	718	18	736	37	0	773	4
20	3	754	19	773	37	0	810	4
21	3	790	20	810	37	0	847	4
22	3	826	21	847	37	0	884	4
23	3	862	22	884	37	0	921	4
24	3	898	23	921	37	0	958	4
25	3	934	24	958	37	0	995	4

MC4 =

1	4	107	0	107	43	107	150	5
2	4	144	6	150	43	0	193	5
3	4	181	12	193	43	0	236	5
4	4	218	18	236	43	0	279	5
5	4	255	24	279	43	0	322	5
6	4	292	30	322	43	0	365	5
7	4	329	36	365	43	0	408	5
8	4	366	42	408	43	0	451	5
9	4	403	48	451	43	0	494	5
10	4	440	54	494	43	0	537	5
11	4	477	60	537	43	0	580	5
12	4	514	66	580	43	0	623	5
13	4	551	72	623	43	0	666	5
14	4	588	78	666	43	0	709	5
15	4	625	84	709	43	0	752	5
16	4	662	90	752	43	0	795	5
17	4	699	96	795	43	0	838	5
18	4	736	102	838	43	0	881	5
19	4	773	108	881	43	0	924	5
20	4	810	114	924	43	0	967	5
21	4	847	120	967	43	0	1010	5
22	4	884	126	1010	43	0	1053	5
23	4	921	132	1053	43	0	1096	5
24	4	958	138	1096	43	0	1139	5
25	4	995	144	1139	43	0	1182	5

MC5 =

1	5	150	0	150	42	150	192	0
2	5	193	0	193	42	1	235	0
3	5	236	0	236	42	1	278	0
4	5	279	0	279	42	1	321	0
5	5	322	0	322	42	1	364	0
6	5	365	0	365	42	1	407	0
7	5	408	0	408	42	1	450	0
8	5	451	0	451	42	1	493	0
9	5	494	0	494	42	1	536	0
10	5	537	0	537	42	1	579	0
11	5	580	0	580	42	1	622	0
12	5	623	0	623	42	1	665	0
13	5	666	0	666	42	1	708	0
14	5	709	0	709	42	1	751	0
15	5	752	0	752	42	1	794	0
16	5	795	0	795	42	1	837	0
17	5	838	0	838	42	1	880	0
18	5	881	0	881	42	1	923	0
19	5	924	0	924	42	1	966	0
20	5	967	0	967	42	1	1009	0
21	5	1010	0	1010	42	1	1052	0
22	5	1053	0	1053	42	1	1095	0
23	5	1096	0	1096	42	1	1138	0
24	5	1139	0	1139	42	1	1181	0
25	5	1182	0	1182	42	1	1224	0

HHR (AGVR) Rover 400-4

Cmax = 2299

MC1 =

1	1	0	0	0	34	0	34	2
2	1	0	34	34	34	0	68	2
3	1	0	68	68	34	0	102	2
4	1	0	102	102	34	0	136	2
5	1	0	136	136	34	0	170	2
6	1	0	170	170	34	0	204	2
7	1	0	204	204	34	0	238	2
8	1	0	238	238	34	0	272	2
9	1	0	272	272	34	0	306	2
10	1	0	306	306	34	0	340	2
11	1	0	340	340	34	0	374	2
12	1	0	374	374	34	0	408	2
13	1	0	408	408	34	0	442	2
14	1	0	442	442	34	0	476	2
15	1	0	476	476	34	0	510	2
16	1	0	510	510	34	0	544	2
17	1	0	544	544	34	0	578	2
18	1	0	578	578	34	0	612	2
19	1	0	612	612	34	0	646	2
20	1	0	646	646	34	0	680	2
21	1	0	680	680	34	0	714	2
22	1	0	714	714	34	0	748	2
23	1	0	748	748	34	0	782	2
24	1	0	782	782	34	0	816	2
25	1	0	816	816	34	0	850	2
26	1	0	850	850	34	0	884	2
27	1	0	884	884	34	0	918	2
28	1	0	918	918	34	0	952	2
29	1	0	952	952	34	0	986	2
30	1	0	986	986	34	0	1020	2
31	1	0	1020	1020	34	0	1054	2
32	1	0	1054	1054	34	0	1088	2
33	1	0	1088	1088	34	0	1122	2
34	1	0	1122	1122	34	0	1156	2
35	1	0	1156	1156	34	0	1190	2
36	1	0	1190	1190	34	0	1224	2
37	1	0	1224	1224	34	0	1258	2
38	1	0	1258	1258	34	0	1292	2
39	1	0	1292	1292	34	0	1326	2
40	1	0	1326	1326	34	0	1360	2
41	1	0	1360	1360	34	0	1394	2
42	1	0	1394	1394	34	0	1428	2
43	1	0	1428	1428	34	0	1462	2
44	1	0	1462	1462	34	0	1496	2
45	1	0	1496	1496	34	0	1530	2
46	1	0	1530	1530	34	0	1564	2
47	1	0	1564	1564	34	0	1598	2
48	1	0	1598	1598	34	0	1632	2
49	1	0	1632	1632	34	0	1666	2
50	1	0	1666	1666	34	0	1700	2

MC2 =

1	2	34	0	34	36	34	70	3
2	2	68	2	70	36	0	106	3
3	2	102	4	106	36	0	142	3
4	2	136	6	142	36	0	178	3
5	2	170	8	178	36	0	214	3
6	2	204	10	214	36	0	250	3
7	2	238	12	250	36	0	286	3
8	2	272	14	286	36	0	322	3
9	2	306	16	322	36	0	358	3
10	2	340	18	358	36	0	394	3
11	2	374	20	394	36	0	430	3
12	2	408	22	430	36	0	466	3
13	2	442	24	466	36	0	502	3
14	2	476	26	502	36	0	538	3
15	2	510	28	538	36	0	574	3
16	2	544	30	574	36	0	610	3
17	2	578	32	610	36	0	646	3
18	2	612	34	646	36	0	682	3
19	2	646	36	682	36	0	718	3
20	2	680	38	718	36	0	754	3
21	2	714	40	754	36	0	790	3
22	2	748	42	790	36	0	826	3
23	2	782	44	826	36	0	862	3
24	2	816	46	862	36	0	898	3

25	2	850	48	898	36	0	934	3
26	2	884	50	934	36	0	970	3
27	2	918	52	970	36	0	1006	3
28	2	952	54	1006	36	0	1042	3
29	2	986	56	1042	36	0	1078	3
30	2	1020	58	1078	36	0	1114	3
31	2	1054	60	1114	36	0	1150	3
32	2	1088	62	1150	36	0	1186	3
33	2	1122	64	1186	36	0	1222	3
34	2	1156	66	1222	36	0	1258	3
35	2	1190	68	1258	36	0	1294	3
36	2	1224	70	1294	36	0	1330	3
37	2	1258	72	1330	36	0	1366	3
38	2	1292	74	1366	36	0	1402	3
39	2	1326	76	1402	36	0	1438	3
40	2	1360	78	1438	36	0	1474	3
41	2	1394	80	1474	36	0	1510	3
42	2	1428	82	1510	36	0	1546	3
43	2	1462	84	1546	36	0	1582	3
44	2	1496	86	1582	36	0	1618	3
45	2	1530	88	1618	36	0	1654	3
46	2	1564	90	1654	36	0	1690	3
47	2	1598	92	1690	36	0	1726	3
48	2	1632	94	1726	36	0	1762	3
49	2	1666	96	1762	36	0	1798	3
50	2	1700	98	1798	36	0	1834	3

MC3 =

1	3	70	0	70	37	70	107	4
2	3	106	1	107	37	0	144	4
3	3	142	2	144	37	0	181	4
4	3	178	3	181	37	0	218	4
5	3	214	4	218	37	0	255	4
6	3	250	5	255	37	0	292	4
7	3	286	6	292	37	0	329	4
8	3	322	7	329	37	0	366	4
9	3	358	8	366	37	0	403	4
10	3	394	9	403	37	0	440	4
11	3	430	10	440	37	0	477	4
12	3	466	11	477	37	0	514	4
13	3	502	12	514	37	0	551	4
14	3	538	13	551	37	0	588	4
15	3	574	14	588	37	0	625	4
16	3	610	15	625	37	0	662	4
17	3	646	16	662	37	0	699	4
18	3	682	17	699	37	0	736	4
19	3	718	18	736	37	0	773	4
20	3	754	19	773	37	0	810	4
21	3	790	20	810	37	0	847	4
22	3	826	21	847	37	0	884	4
23	3	862	22	884	37	0	921	4
24	3	898	23	921	37	0	958	4
25	3	934	24	958	37	0	995	4
26	3	970	25	995	37	0	1032	4
27	3	1006	26	1032	37	0	1069	4
28	3	1042	27	1069	37	0	1106	4
29	3	1078	28	1106	37	0	1143	4
30	3	1114	29	1143	37	0	1180	4
31	3	1150	30	1180	37	0	1217	4
32	3	1186	31	1217	37	0	1254	4
33	3	1222	32	1254	37	0	1291	4
34	3	1258	33	1291	37	0	1328	4
35	3	1294	34	1328	37	0	1365	4
36	3	1330	35	1365	37	0	1402	4
37	3	1366	36	1402	37	0	1439	4
38	3	1402	37	1439	37	0	1476	4
39	3	1438	38	1476	37	0	1513	4
40	3	1474	39	1513	37	0	1550	4
41	3	1510	40	1550	37	0	1587	4
42	3	1546	41	1587	37	0	1624	4
43	3	1582	42	1624	37	0	1661	4
44	3	1618	43	1661	37	0	1698	4
45	3	1654	44	1698	37	0	1735	4
46	3	1690	45	1735	37	0	1772	4
47	3	1726	46	1772	37	0	1809	4
48	3	1762	47	1809	37	0	1846	4
49	3	1798	48	1846	37	0	1883	4
50	3	1834	49	1883	37	0	1920	4

MC4 =

1	4	107	0	107	43	107	150	5
2	4	144	6	150	43	0	193	5
3	4	181	12	193	43	0	236	5
4	4	218	18	236	43	0	279	5
5	4	255	24	279	43	0	322	5
6	4	292	30	322	43	0	365	5
7	4	329	36	365	43	0	408	5
8	4	366	42	408	43	0	451	5
9	4	403	48	451	43	0	494	5
10	4	440	54	494	43	0	537	5
11	4	477	60	537	43	0	580	5
12	4	514	66	580	43	0	623	5
13	4	551	72	623	43	0	666	5
14	4	588	78	666	43	0	709	5
15	4	625	84	709	43	0	752	5
16	4	662	90	752	43	0	795	5
17	4	699	96	795	43	0	838	5
18	4	736	102	838	43	0	881	5
19	4	773	108	881	43	0	924	5
20	4	810	114	924	43	0	967	5
21	4	847	120	967	43	0	1010	5
22	4	884	126	1010	43	0	1053	5
23	4	921	132	1053	43	0	1096	5
24	4	958	138	1096	43	0	1139	5
25	4	995	144	1139	43	0	1182	5
26	4	1032	150	1182	43	0	1225	5
27	4	1069	156	1225	43	0	1268	5
28	4	1106	162	1268	43	0	1311	5
29	4	1143	168	1311	43	0	1354	5
30	4	1180	174	1354	43	0	1397	5
31	4	1217	180	1397	43	0	1440	5
32	4	1254	186	1440	43	0	1483	5
33	4	1291	192	1483	43	0	1526	5
34	4	1328	198	1526	43	0	1569	5
35	4	1365	204	1569	43	0	1612	5
36	4	1402	210	1612	43	0	1655	5
37	4	1439	216	1655	43	0	1698	5
38	4	1476	222	1698	43	0	1741	5
39	4	1513	228	1741	43	0	1784	5
40	4	1550	234	1784	43	0	1827	5
41	4	1587	240	1827	43	0	1870	5
42	4	1624	246	1870	43	0	1913	5
43	4	1661	252	1913	43	0	1956	5
44	4	1698	258	1956	43	0	1999	5
45	4	1735	264	1999	43	0	2042	5
46	4	1772	270	2042	43	0	2085	5
47	4	1809	276	2085	43	0	2128	5
48	4	1846	282	2128	43	0	2171	5
49	4	1883	288	2171	43	0	2214	5
50	4	1920	294	2214	43	0	2257	5

MC5 =

1	5	150	0	150	42	150	192	0
2	5	193	0	193	42	1	235	0
3	5	236	0	236	42	1	278	0
4	5	279	0	279	42	1	321	0
5	5	322	0	322	42	1	364	0
6	5	365	0	365	42	1	407	0
7	5	408	0	408	42	1	450	0
8	5	451	0	451	42	1	493	0
9	5	494	0	494	42	1	536	0
10	5	537	0	537	42	1	579	0
11	5	580	0	580	42	1	622	0
12	5	623	0	623	42	1	665	0
13	5	666	0	666	42	1	708	0
14	5	709	0	709	42	1	751	0
15	5	752	0	752	42	1	794	0
16	5	795	0	795	42	1	837	0
17	5	838	0	838	42	1	880	0
18	5	881	0	881	42	1	923	0
19	5	924	0	924	42	1	966	0
20	5	967	0	967	42	1	1009	0
21	5	1010	0	1010	42	1	1052	0
22	5	1053	0	1053	42	1	1095	0
23	5	1096	0	1096	42	1	1138	0
24	5	1139	0	1139	42	1	1181	0
25	5	1182	0	1182	42	1	1224	0
26	5	1225	0	1225	42	1	1267	0
27	5	1268	0	1268	42	1	1310	0
28	5	1311	0	1311	42	1	1353	0
29	5	1354	0	1354	42	1	1396	0
30	5	1397	0	1397	42	1	1439	0
31	5	1440	0	1440	42	1	1482	0
32	5	1483	0	1483	42	1	1525	0
33	5	1526	0	1526	42	1	1568	0
34	5	1569	0	1569	42	1	1611	0
35	5	1612	0	1612	42	1	1654	0
36	5	1655	0	1655	42	1	1697	0
37	5	1698	0	1698	42	1	1740	0
38	5	1741	0	1741	42	1	1783	0
39	5	1784	0	1784	42	1	1826	0
40	5	1827	0	1827	42	1	1869	0
41	5	1870	0	1870	42	1	1912	0
42	5	1913	0	1913	42	1	1955	0
43	5	1956	0	1956	42	1	1998	0
44	5	1999	0	1999	42	1	2041	0
45	5	2042	0	2042	42	1	2084	0
46	5	2085	0	2085	42	1	2127	0
47	5	2128	0	2128	42	1	2170	0
48	5	2171	0	2171	42	1	2213	0
49	5	2214	0	2214	42	1	2256	0
50	5	2257	0	2257	42	1	2299	0

HHR (AGVR) Rover 400-5

Cmax = 3374

MC1 =

1	1	0	0	0	34	0	34	2
2	1	0	34	34	34	0	68	2
3	1	0	68	68	34	0	102	2
4	1	0	102	102	34	0	136	2
5	1	0	136	136	34	0	170	2
6	1	0	170	170	34	0	204	2
7	1	0	204	204	34	0	238	2
8	1	0	238	238	34	0	272	2
9	1	0	272	272	34	0	306	2
10	1	0	306	306	34	0	340	2
11	1	0	340	340	34	0	374	2
12	1	0	374	374	34	0	408	2
13	1	0	408	408	34	0	442	2
14	1	0	442	442	34	0	476	2
15	1	0	476	476	34	0	510	2
16	1	0	510	510	34	0	544	2
17	1	0	544	544	34	0	578	2
18	1	0	578	578	34	0	612	2
19	1	0	612	612	34	0	646	2
20	1	0	646	646	34	0	680	2

21	1	0	680	680	34	0	714	2
22	1	0	714	714	34	0	748	2
23	1	0	748	748	34	0	782	2
24	1	0	782	782	34	0	816	2
25	1	0	816	816	34	0	850	2
26	1	0	850	850	34	0	884	2
27	1	0	884	884	34	0	918	2
28	1	0	918	918	34	0	952	2
29	1	0	952	952	34	0	986	2
30	1	0	986	986	34	0	1020	2
31	1	0	1020	1020	34	0	1054	2
32	1	0	1054	1054	34	0	1088	2
33	1	0	1088	1088	34	0	1122	2
34	1	0	1122	1122	34	0	1156	2
35	1	0	1156	1156	34	0	1190	2
36	1	0	1190	1190	34	0	1224	2
37	1	0	1224	1224	34	0	1258	2
38	1	0	1258	1258	34	0	1292	2
39	1	0	1292	1292	34	0	1326	2
40	1	0	1326	1326	34	0	1360	2
41	1	0	1360	1360	34	0	1394	2
42	1	0	1394	1394	34	0	1428	2
43	1	0	1428	1428	34	0	1462	2
44	1	0	1462	1462	34	0	1496	2
45	1	0	1496	1496	34	0	1530	2
46	1	0	1530	1530	34	0	1564	2
47	1	0	1564	1564	34	0	1598	2
48	1	0	1598	1598	34	0	1632	2
49	1	0	1632	1632	34	0	1666	2
50	1	0	1666	1666	34	0	1700	2
51	1	0	1700	1700	34	0	1734	2
52	1	0	1734	1734	34	0	1768	2
53	1	0	1768	1768	34	0	1802	2
54	1	0	1802	1802	34	0	1836	2
55	1	0	1836	1836	34	0	1870	2
56	1	0	1870	1870	34	0	1904	2
57	1	0	1904	1904	34	0	1938	2
58	1	0	1938	1938	34	0	1972	2
59	1	0	1972	1972	34	0	2006	2
60	1	0	2006	2006	34	0	2040	2
61	1	0	2040	2040	34	0	2074	2
62	1	0	2074	2074	34	0	2108	2
63	1	0	2108	2108	34	0	2142	2
64	1	0	2142	2142	34	0	2176	2
65	1	0	2176	2176	34	0	2210	2
66	1	0	2210	2210	34	0	2244	2
67	1	0	2244	2244	34	0	2278	2
68	1	0	2278	2278	34	0	2312	2
69	1	0	2312	2312	34	0	2346	2
70	1	0	2346	2346	34	0	2380	2
71	1	0	2380	2380	34	0	2414	2
72	1	0	2414	2414	34	0	2448	2
73	1	0	2448	2448	34	0	2482	2
74	1	0	2482	2482	34	0	2516	2
75	1	0	2516	2516	34	0	2550	2

MC2 =

1	2	34	0	34	36	34	70	3
2	2	68	2	70	36	0	106	3
3	2	102	4	106	36	0	142	3
4	2	136	6	142	36	0	178	3
5	2	170	8	178	36	0	214	3
6	2	204	10	214	36	0	250	3
7	2	238	12	250	36	0	286	3
8	2	272	14	286	36	0	322	3
9	2	306	16	322	36	0	358	3
10	2	340	18	358	36	0	394	3
11	2	374	20	394	36	0	430	3
12	2	408	22	430	36	0	466	3
13	2	442	24	466	36	0	502	3
14	2	476	26	502	36	0	538	3
15	2	510	28	538	36	0	574	3
16	2	544	30	574	36	0	610	3
17	2	578	32	610	36	0	646	3
18	2	612	34	646	36	0	682	3
19	2	646	36	682	36	0	718	3
20	2	680	38	718	36	0	754	3
21	2	714	40	754	36	0	790	3
22	2	748	42	790	36	0	826	3
23	2	782	44	826	36	0	862	3
24	2	816	46	862	36	0	898	3
25	2	850	48	898	36	0	934	3
26	2	884	50	934	36	0	970	3
27	2	918	52	970	36	0	1006	3
28	2	952	54	1006	36	0	1042	3
29	2	986	56	1042	36	0	1078	3
30	2	1020	58	1078	36	0	1114	3
31	2	1054	60	1114	36	0	1150	3
32	2	1088	62	1150	36	0	1186	3
33	2	1122	64	1186	36	0	1222	3
34	2	1156	66	1222	36	0	1258	3
35	2	1190	68	1258	36	0	1294	3
36	2	1224	70	1294	36	0	1330	3
37	2	1258	72	1330	36	0	1366	3
38	2	1292	74	1366	36	0	1402	3
39	2	1326	76	1402	36	0	1438	3
40	2	1360	78	1438	36	0	1474	3
41	2	1394	80	1474	36	0	1510	3
42	2	1428	82	1510	36	0	1546	3
43	2	1462	84	1546	36	0	1582	3
44	2	1496	86	1582	36	0	1618	3
45	2	1530	88	1618	36	0	1654	3
46	2	1564	90	1654	36	0	1690	3
47	2	1598	92	1690	36	0	1726	3
48	2	1632	94	1726	36	0	1762	3
49	2	1666	96	1762	36	0	1798	3
50	2	1700	98	1798	36	0	1834	3
51	2	1734	100	1834	36	0	1870	3
52	2	1768	102	1870	36	0	1906	3
53	2	1802	104	1906	36	0	1942	3
54	2	1836	106	1942	36	0	1978	3
55	2	1870	108	1978	36	0	2014	3
56	2	1904	110	2014	36	0	2050	3
57	2	1938	112	2050	36	0	2086	3
58	2	1972	114	2086	36	0	2122	3
59	2	2006	116	2122	36	0	2158	3
60	2	2040	118	2158	36	0	2194	3
61	2	2074	120	2194	36	0	2230	3
62	2	2108	122	2230	36	0	2266	3
63	2	2142	124	2266	36	0	2302	3
64	2	2176	126	2302	36	0	2338	3
65	2	2210	128	2338	36	0	2374	3
66	2	2244	130	2374	36	0	2410	3
67	2	2278	132	2410	36	0	2446	3
68	2	2312	134	2446	36	0	2482	3
69	2	2346	136	2482	36	0	2518	3
70	2	2380	138	2518	36	0	2554	3
71	2	2414	140	2554	36	0	2590	3
72	2	2448	142	2590	36	0	2626	3
73	2	2482	144	2626	36	0	2662	3
74	2	2516	146	2662	36	0	2698	3
75	2	2550	148	2698	36	0	2734	3

MC3 =

1	3	70	0	70	37	70	107	4
2	3	106	1	107	37	0	144	4
3	3	142	2	144	37	0	181	4
4	3	178	3	181	37	0	218	4
5	3	214	4	218	37	0	255	4
6	3	250	5	255	37	0	292	4
7	3	286	6	292	37	0	329	4
8	3	322	7	329	37	0	366	4
9	3	358	8	366	37	0	403	4
10	3	394	9	403	37	0	440	4
11	3	430	10	440	37	0	477	4
12	3	466	11	477	37	0	514	4
13	3	502	12	514	37	0	551	4
14	3	538	13	551	37	0	588	4
15	3	574	14	588	37	0	625	4
16	3	610	15	625	37	0	662	4
17	3	646	16	662	37	0	699	4
18	3	682	17	699	37	0	736	4
19	3	718	18	736	37	0	773	4
20	3	754	19	773	37	0	810	4
21	3	790	20	810	37	0	847	4
22	3	826	21	847	37	0	884	4
23	3	862	22	884	37	0	921	4
24	3	898	23	921	37	0	958	4
25	3	934	24	958	37	0	995	4
26	3	970	25	995	37	0	1032	4
27	3	1006	26	1032	37	0	1069	4
28	3	1042	27	1069	37	0	1106	4
29	3	1078	28	1106	37	0	1143	4
30	3	1114	29	1143	37	0	1180	4
31	3	1150	30	1180	37	0	1217	4
32	3	1186	31	1217	37	0	1254	4
33	3	1222	32	1254	37	0	1291	4
34	3	1258	33	1291	37	0	1328	4
35	3	1294	34	1328	37	0	1365	4
36	3	1330	35	1365	37	0	1402	4
37	3	1366	36	1402	37	0	1439	4
38	3	1402	37	1439	37	0	1476	4
39	3	1438	38	1476	37	0	1513	4
40	3	1474	39	1513	37	0	1550	4
41	3	1510	40	1550	37	0	1587	4
42	3	1546	41	1587	37	0	1624	4
43	3	1582	42	1624	37	0	1661	4
44	3	1618	43	1661	37	0	1698	4
45	3	1654	44	1698	37	0	1735	4
46	3	1690	45	1735	37	0	1772	4
47	3	1726	46	1772	37	0	1809	4
48	3	1762	47	1809	37	0	1846	4
49	3	1798	48	1846	37	0	1883	4
50	3	1834	49	1883	37	0	1920	4
51	3	1870	50	1920	37	0	1957	4
52	3	1906	51	1957	37	0	1994	4
53	3	1942	52	1994	37	0	2031	4
54	3	1978	53	2031	37	0	2068	4
55	3	2014	54	2068	37	0	2105	4
56	3	2050	55	2105	37	0	2142	4
57	3	2086	56	2142	37	0	2179	4
58	3	2122	57	2179	37	0	2216	4
59	3	2158	58	2216	37	0	2253	4
60	3	2194	59	2253	37	0	2290	4
61	3	2230	60	2290	37	0	2327	4
62	3	2266	61	2327	37	0	2364	4
63	3	2302	62	2364	37	0	2401	4
64	3	2338	63	2401	37	0	2438	4
65	3	2374	64	2438	37	0	2475	4
66	3	2410	65	2475	37	0	2512	4
67	3	2446	66	2512	37	0	2549	4
68	3	2482	67	2549	37	0	2586	4
69	3	2518	68	2586	37	0	2623	4
70	3	2554	69	2623	37	0	2660	4

71	3	2590	70	2660	37	0	2697	4
72	3	2626	71	2697	37	0	2734	4
73	3	2662	72	2734	37	0	2771	4
74	3	2698	73	2771	37	0	2808	4
75	3	2734	74	2808	37	0	2845	4

MC4 =

1	4	107	0	107	43	107	150	5
2	4	144	6	150	43	0	193	5
3	4	181	12	193	43	0	236	5
4	4	218	18	236	43	0	279	5
5	4	255	24	279	43	0	322	5
6	4	292	30	322	43	0	365	5
7	4	329	36	365	43	0	408	5
8	4	366	42	408	43	0	451	5
9	4	403	48	451	43	0	494	5
10	4	440	54	494	43	0	537	5
11	4	477	60	537	43	0	580	5
12	4	514	66	580	43	0	623	5
13	4	551	72	623	43	0	666	5
14	4	588	78	666	43	0	709	5
15	4	625	84	709	43	0	752	5
16	4	662	90	752	43	0	795	5
17	4	699	96	795	43	0	838	5
18	4	736	102	838	43	0	881	5
19	4	773	108	881	43	0	924	5
20	4	810	114	924	43	0	967	5
21	4	847	120	967	43	0	1010	5
22	4	884	126	1010	43	0	1053	5
23	4	921	132	1053	43	0	1096	5
24	4	958	138	1096	43	0	1139	5
25	4	995	144	1139	43	0	1182	5
26	4	1032	150	1182	43	0	1225	5
27	4	1069	156	1225	43	0	1268	5
28	4	1106	162	1268	43	0	1311	5
29	4	1143	168	1311	43	0	1354	5
30	4	1180	174	1354	43	0	1397	5
31	4	1217	180	1397	43	0	1440	5
32	4	1254	186	1440	43	0	1483	5
33	4	1291	192	1483	43	0	1526	5
34	4	1328	198	1526	43	0	1569	5
35	4	1365	204	1569	43	0	1612	5
36	4	1402	210	1612	43	0	1655	5
37	4	1439	216	1655	43	0	1698	5
38	4	1476	222	1698	43	0	1741	5
39	4	1513	228	1741	43	0	1784	5
40	4	1550	234	1784	43	0	1827	5
41	4	1587	240	1827	43	0	1870	5
42	4	1624	246	1870	43	0	1913	5
43	4	1661	252	1913	43	0	1956	5
44	4	1698	258	1956	43	0	1999	5
45	4	1735	264	1999	43	0	2042	5
46	4	1772	270	2042	43	0	2085	5
47	4	1809	276	2085	43	0	2128	5
48	4	1846	282	2128	43	0	2171	5
49	4	1883	288	2171	43	0	2214	5
50	4	1920	294	2214	43	0	2257	5
51	4	1957	300	2257	43	0	2300	5
52	4	1994	306	2300	43	0	2343	5
53	4	2031	312	2343	43	0	2386	5
54	4	2068	318	2386	43	0	2429	5
55	4	2105	324	2429	43	0	2472	5
56	4	2142	330	2472	43	0	2515	5
57	4	2179	336	2515	43	0	2558	5
58	4	2216	342	2558	43	0	2601	5
59	4	2253	348	2601	43	0	2644	5
60	4	2290	354	2644	43	0	2687	5
61	4	2327	360	2687	43	0	2730	5
62	4	2364	366	2730	43	0	2773	5
63	4	2401	372	2773	43	0	2816	5
64	4	2438	378	2816	43	0	2859	5
65	4	2475	384	2859	43	0	2902	5
66	4	2512	390	2902	43	0	2945	5
67	4	2549	396	2945	43	0	2988	5
68	4	2586	402	2988	43	0	3031	5
69	4	2623	408	3031	43	0	3074	5
70	4	2660	414	3074	43	0	3117	5
71	4	2697	420	3117	43	0	3160	5
72	4	2734	426	3160	43	0	3203	5
73	4	2771	432	3203	43	0	3246	5
74	4	2808	438	3246	43	0	3289	5
75	4	2845	444	3289	43	0	3332	5

MC5 =

1	5	150	0	150	42	150	192	0
2	5	193	0	193	42	1	235	0
3	5	236	0	236	42	1	278	0
4	5	279	0	279	42	1	321	0
5	5	322	0	322	42	1	364	0
6	5	365	0	365	42	1	407	0
7	5	408	0	408	42	1	450	0
8	5	451	0	451	42	1	493	0
9	5	494	0	494	42	1	536	0
10	5	537	0	537	42	1	579	0
11	5	580	0	580	42	1	622	0
12	5	623	0	623	42	1	665	0
13	5	666	0	666	42	1	708	0
14	5	709	0	709	42	1	751	0
15	5	752	0	752	42	1	794	0
16	5	795	0	795	42	1	837	0
17	5	838	0	838	42	1	880	0
18	5	881	0	881	42	1	923	0
19	5	924	0	924	42	1	966	0
20	5	967	0	967	42	1	1009	0
21	5	1010	0	1010	42	1	1052	0
22	5	1053	0	1053	42	1	1095	0
23	5	1096	0	1096	42	1	1138	0
24	5	1139	0	1139	42	1	1181	0
25	5	1182	0	1182	42	1	1224	0
26	5	1225	0	1225	42	1	1267	0
27	5	1268	0	1268	42	1	1310	0
28	5	1311	0	1311	42	1	1353	0
29	5	1354	0	1354	42	1	1396	0
30	5	1397	0	1397	42	1	1439	0
31	5	1440	0	1440	42	1	1482	0
32	5	1483	0	1483	42	1	1525	0
33	5	1526	0	1526	42	1	1568	0
34	5	1569	0	1569	42	1	1611	0
35	5	1612	0	1612	42	1	1654	0
36	5	1655	0	1655	42	1	1697	0
37	5	1698	0	1698	42	1	1740	0
38	5	1741	0	1741	42	1	1783	0
39	5	1784	0	1784	42	1	1826	0
40	5	1827	0	1827	42	1	1869	0
41	5	1870	0	1870	42	1	1912	0
42	5	1913	0	1913	42	1	1955	0
43	5	1956	0	1956	42	1	1998	0
44	5	1999	0	1999	42	1	2041	0
45	5	2042	0	2042	42	1	2084	0
46	5	2085	0	2085	42	1	2127	0
47	5	2128	0	2128	42	1	2170	0
48	5	2171	0	2171	42	1	2213	0
49	5	2214	0	2214	42	1	2256	0
50	5	2257	0	2257	42	1	2299	0
51	5	2300	0	2300	42	1	2342	0
52	5	2343	0	2343	42	1	2385	0
53	5	2386	0	2386	42	1	2428	0
54	5	2429	0	2429	42	1	2471	0
55	5	2472	0	2472	42	1	2514	0
56	5	2515	0	2515	42	1	2557	0
57	5	2558	0	2558	42	1	2600	0
58	5	2601	0	2601	42	1	2643	0
59	5	2644	0	2644	42	1	2686	0
60	5	2687	0	2687	42	1	2729	0
61	5	2730	0	2730	42	1	2772	0
62	5	2773	0	2773	42	1	2815	0
63	5	2816	0	2816	42	1	2858	0
64	5	2859	0	2859	42	1	2901	0
65	5	2902	0	2902	42	1	2944	0
66	5	2945	0	2945	42	1	2987	0
67	5	2988	0	2988	42	1	3030	0
68	5	3031	0	3031	42	1	3073	0
69	5	3074	0	3074	42	1	3116	0
70	5	3117	0	3117	42	1	3159	0
71	5	3160	0	3160	42	1	3202	0
72	5	3203	0	3203	42	1	3245	0
73	5	3246	0	3246	42	1	3288	0
74	5	3289	0	3289	42	1	3331	0
75	5	3332	0	3332	42	1	3374	0

HHR (AGVR) Rover 400-6

Cmax = 4449

MC1 =

1	1	0	0	0	34	0	34	2
2	1	0	34	34	34	0	68	2
3	1	0	68	68	34	0	102	2
4	1	0	102	102	34	0	136	2
5	1	0	136	136	34	0	170	2
6	1	0	170	170	34	0	204	2
7	1	0	204	204	34	0	238	2
8	1	0	238	238	34	0	272	2
9	1	0	272	272	34	0	306	2
10	1	0	306	306	34	0	340	2
11	1	0	340	340	34	0	374	2
12	1	0	374	374	34	0	408	2
13	1	0	408	408	34	0	442	2
14	1	0	442	442	34	0	476	2
15	1	0	476	476	34	0	510	2
16	1	0	510	510	34	0	544	2
17	1	0	544	544	34	0	578	2
18	1	0	578	578	34	0	612	2
19	1	0	612	612	34	0	646	2
20	1	0	646	646	34	0	680	2
21	1	0	680	680	34	0	714	2
22	1	0	714	714	34	0	748	2
23	1	0	748	748	34	0	782	2
24	1	0	782	782	34	0	816	2
25	1	0	816	816	34	0	850	2
26	1	0	850	850	34	0	884	2
27	1	0	884	884	34	0	918	2
28	1	0	918	918	34	0	952	2
29	1	0	952	952	34	0	986	2
30	1	0	986	986	34	0	1020	2
31	1	0	1020	1020	34	0	1054	2
32	1	0	1054	1054	34	0	1088	2
33	1	0	1088	1088	34	0	1122	2
34	1	0	1122	1122	34	0	1156	2
35	1	0	1156	1156	34	0	1190	2
36	1	0	1190	1190	34	0	1224	2
37	1	0	1224	1224	34	0	1258	2
38	1	0	1258	1258	34	0	1292	2
39	1	0	1292	1292	34	0	1326	2
40	1	0	1326	1326	34	0	1360	2
41	1	0	1360	1360	34	0	1394	2
42	1	0	1394	1394	34	0	1428	2

43	1	0	1428	1428	34	0	1462	2
44	1	0	1462	1462	34	0	1496	2
45	1	0	1496	1496	34	0	1530	2
46	1	0	1530	1530	34	0	1564	2
47	1	0	1564	1564	34	0	1598	2
48	1	0	1598	1598	34	0	1632	2
49	1	0	1632	1632	34	0	1666	2
50	1	0	1666	1666	34	0	1700	2
51	1	0	1700	1700	34	0	1734	2
52	1	0	1734	1734	34	0	1768	2
53	1	0	1768	1768	34	0	1802	2
54	1	0	1802	1802	34	0	1836	2
55	1	0	1836	1836	34	0	1870	2
56	1	0	1870	1870	34	0	1904	2
57	1	0	1904	1904	34	0	1938	2
58	1	0	1938	1938	34	0	1972	2
59	1	0	1972	1972	34	0	2006	2
60	1	0	2006	2006	34	0	2040	2
61	1	0	2040	2040	34	0	2074	2
62	1	0	2074	2074	34	0	2108	2
63	1	0	2108	2108	34	0	2142	2
64	1	0	2142	2142	34	0	2176	2
65	1	0	2176	2176	34	0	2210	2
66	1	0	2210	2210	34	0	2244	2
67	1	0	2244	2244	34	0	2278	2
68	1	0	2278	2278	34	0	2312	2
69	1	0	2312	2312	34	0	2346	2
70	1	0	2346	2346	34	0	2380	2
71	1	0	2380	2380	34	0	2414	2
72	1	0	2414	2414	34	0	2448	2
73	1	0	2448	2448	34	0	2482	2
74	1	0	2482	2482	34	0	2516	2
75	1	0	2516	2516	34	0	2550	2
76	1	0	2550	2550	34	0	2584	2
77	1	0	2584	2584	34	0	2618	2
78	1	0	2618	2618	34	0	2652	2
79	1	0	2652	2652	34	0	2686	2
80	1	0	2686	2686	34	0	2720	2
81	1	0	2720	2720	34	0	2754	2
82	1	0	2754	2754	34	0	2788	2
83	1	0	2788	2788	34	0	2822	2
84	1	0	2822	2822	34	0	2856	2
85	1	0	2856	2856	34	0	2890	2
86	1	0	2890	2890	34	0	2924	2
87	1	0	2924	2924	34	0	2958	2
88	1	0	2958	2958	34	0	2992	2
89	1	0	2992	2992	34	0	3026	2
90	1	0	3026	3026	34	0	3060	2
91	1	0	3060	3060	34	0	3094	2
92	1	0	3094	3094	34	0	3128	2
93	1	0	3128	3128	34	0	3162	2
94	1	0	3162	3162	34	0	3196	2
95	1	0	3196	3196	34	0	3230	2
96	1	0	3230	3230	34	0	3264	2
97	1	0	3264	3264	34	0	3298	2
98	1	0	3298	3298	34	0	3332	2
99	1	0	3332	3332	34	0	3366	2
100	1	0	3366	3366	34	0	3400	2

MC2 =

1	2	34	0	34	36	34	70	3
2	2	68	2	70	36	0	106	3
3	2	102	4	106	36	0	142	3
4	2	136	6	142	36	0	178	3
5	2	170	8	178	36	0	214	3
6	2	204	10	214	36	0	250	3
7	2	238	12	250	36	0	286	3
8	2	272	14	286	36	0	322	3
9	2	306	16	322	36	0	358	3
10	2	340	18	358	36	0	394	3
11	2	374	20	394	36	0	430	3
12	2	408	22	430	36	0	466	3
13	2	442	24	466	36	0	502	3
14	2	476	26	502	36	0	538	3
15	2	510	28	538	36	0	574	3
16	2	544	30	574	36	0	610	3
17	2	578	32	610	36	0	646	3
18	2	612	34	646	36	0	682	3
19	2	646	36	682	36	0	718	3
20	2	680	38	718	36	0	754	3
21	2	714	40	754	36	0	790	3
22	2	748	42	790	36	0	826	3
23	2	782	44	826	36	0	862	3
24	2	816	46	862	36	0	898	3
25	2	850	48	898	36	0	934	3
26	2	884	50	934	36	0	970	3
27	2	918	52	970	36	0	1006	3
28	2	952	54	1006	36	0	1042	3
29	2	986	56	1042	36	0	1078	3
30	2	1020	58	1078	36	0	1114	3
31	2	1054	60	1114	36	0	1150	3
32	2	1088	62	1150	36	0	1186	3
33	2	1122	64	1186	36	0	1222	3
34	2	1156	66	1222	36	0	1258	3
35	2	1190	68	1258	36	0	1294	3
36	2	1224	70	1294	36	0	1330	3
37	2	1258	72	1330	36	0	1366	3
38	2	1292	74	1366	36	0	1402	3
39	2	1326	76	1402	36	0	1438	3
40	2	1360	78	1438	36	0	1474	3
41	2	1394	80	1474	36	0	1510	3
42	2	1428	82	1510	36	0	1546	3

43	2	1462	84	1546	36	0	1582	3
44	2	1496	86	1582	36	0	1618	3
45	2	1530	88	1618	36	0	1654	3
46	2	1564	90	1654	36	0	1690	3
47	2	1598	92	1690	36	0	1726	3
48	2	1632	94	1726	36	0	1762	3
49	2	1666	96	1762	36	0	1798	3
50	2	1700	98	1798	36	0	1834	3
51	2	1734	100	1834	36	0	1870	3
52	2	1768	102	1870	36	0	1906	3
53	2	1802	104	1906	36	0	1942	3
54	2	1836	106	1942	36	0	1978	3
55	2	1870	108	1978	36	0	2014	3
56	2	1904	110	2014	36	0	2050	3
57	2	1938	112	2050	36	0	2086	3
58	2	1972	114	2086	36	0	2122	3
59	2	2006	116	2122	36	0	2158	3
60	2	2040	118	2158	36	0	2194	3
61	2	2074	120	2194	36	0	2230	3
62	2	2108	122	2230	36	0	2266	3
63	2	2142	124	2266	36	0	2302	3
64	2	2176	126	2302	36	0	2338	3
65	2	2210	128	2338	36	0	2374	3
66	2	2244	130	2374	36	0	2410	3
67	2	2278	132	2410	36	0	2446	3
68	2	2312	134	2446	36	0	2482	3
69	2	2346	136	2482	36	0	2518	3
70	2	2380	138	2518	36	0	2554	3
71	2	2414	140	2554	36	0	2590	3
72	2	2448	142	2590	36	0	2626	3
73	2	2482	144	2626	36	0	2662	3
74	2	2516	146	2662	36	0	2698	3
75	2	2550	148	2698	36	0	2734	3
76	2	2584	150	2734	36	0	2770	3
77	2	2618	152	2770	36	0	2806	3
78	2	2652	154	2806	36	0	2842	3
79	2	2686	156	2842	36	0	2878	3
80	2	2720	158	2878	36	0	2914	3
81	2	2754	160	2914	36	0	2950	3
82	2	2788	162	2950	36	0	2986	3
83	2	2822	164	2986	36	0	3022	3
84	2	2856	166	3022	36	0	3058	3
85	2	2890	168	3058	36	0	3094	3
86	2	2924	170	3094	36	0	3130	3
87	2	2958	172	3130	36	0	3166	3
88	2	2992	174	3166	36	0	3202	3
89	2	3026	176	3202	36	0	3238	3
90	2	3060	178	3238	36	0	3274	3
91	2	3094	180	3274	36	0	3310	3
92	2	3128	182	3310	36	0	3346	3
93	2	3162	184	3346	36	0	3382	3
94	2	3196	186	3382	36	0	3418	3
95	2	3230	188	3418	36	0	3454	3
96	2	3264	190	3454	36	0	3490	3
97	2	3298	192	3490	36	0	3526	3
98	2	3332	194	3526	36	0	3562	3
99	2	3366	196	3562	36	0	3598	3
100	2	3400	198	3598	36	0	3634	3

MC3 =

1	3	70	0	70	37	70	107	4
2	3	106	1	107	37	0	144	4
3	3	142	2	144	37	0	181	4
4	3	178	3	181	37	0	218	4
5	3	214	4	218	37	0	255	4
6	3	250	5	255	37	0	292	4
7	3	286	6	292	37	0	329	4
8	3	322	7	329	37	0	366	4
9	3	358	8	366	37	0	403	4
10	3	394	9	403	37	0	440	4
11	3	430	10	440	37	0	477	4
12	3	466	11	477	37	0	514	4
13	3	502	12	514	37	0	551	4
14	3	538	13	551	37	0	588	4
15	3	574	14	588	37	0	625	4
16	3	610	15	625	37	0	662	4
17	3	646	16	662	37	0	699	4
18	3	682	17	699	37	0	736	4
19	3	718	18	736	37	0	773	4
20	3	754	19	773	37	0	810	4
21	3	790	20	810	37	0	847	4
22	3	826	21	847	37	0	884	4
23	3	862	22	884	37	0	921	4
24	3	898	23	921	37	0	958	4
25	3	934	24	958	37	0	995	4
26	3	970	25	995	37	0	1032	4
27	3	1006	26	1032	37	0	1069	4
28	3	1042	27	1069	37	0	1106	4
29	3	1078	28	1106	37	0	1143	4
30	3	1114	29	1143	37	0	1180	4
31	3	1150	30	1180	37	0	1217	4
32	3	1186	31	1217	37	0	1254	4
33	3	1222	32	1254	37	0	1291	4
34	3	1258	33	1291	37	0	1328	4
35	3	1294	34	1328	37	0	1365	4
36	3	1330	35	1365	37	0	1402	4
37	3	1366	36	1402	37	0	1439	4
38	3	1402	37	1439	37	0	1476	4
39	3	1438	38	1476	37	0	1513	4
40	3	1474	39	1513	37	0	1550	4
41	3	1510	40	1550	37	0	1587	4
42	3	1546	41	1587	37	0	1624	4

43	3	1582	42	1624	37	0	1661	4
44	3	1618	43	1661	37	0	1698	4
45	3	1654	44	1698	37	0	1735	4
46	3	1690	45	1735	37	0	1772	4
47	3	1726	46	1772	37	0	1809	4
48	3	1762	47	1809	37	0	1846	4
49	3	1798	48	1846	37	0	1883	4
50	3	1834	49	1883	37	0	1920	4
51	3	1870	50	1920	37	0	1957	4
52	3	1906	51	1957	37	0	1994	4
53	3	1942	52	1994	37	0	2031	4
54	3	1978	53	2031	37	0	2068	4
55	3	2014	54	2068	37	0	2105	4
56	3	2050	55	2105	37	0	2142	4
57	3	2086	56	2142	37	0	2179	4
58	3	2122	57	2179	37	0	2216	4
59	3	2158	58	2216	37	0	2253	4
60	3	2194	59	2253	37	0	2290	4
61	3	2230	60	2290	37	0	2327	4
62	3	2266	61	2327	37	0	2364	4
63	3	2302	62	2364	37	0	2401	4
64	3	2338	63	2401	37	0	2438	4
65	3	2374	64	2438	37	0	2475	4
66	3	2410	65	2475	37	0	2512	4
67	3	2446	66	2512	37	0	2549	4
68	3	2482	67	2549	37	0	2586	4
69	3	2518	68	2586	37	0	2623	4
70	3	2554	69	2623	37	0	2660	4
71	3	2590	70	2660	37	0	2697	4
72	3	2626	71	2697	37	0	2734	4
73	3	2662	72	2734	37	0	2771	4
74	3	2698	73	2771	37	0	2808	4
75	3	2734	74	2808	37	0	2845	4
76	3	2770	75	2845	37	0	2882	4
77	3	2806	76	2882	37	0	2919	4
78	3	2842	77	2919	37	0	2956	4
79	3	2878	78	2956	37	0	2993	4
80	3	2914	79	2993	37	0	3030	4
81	3	2950	80	3030	37	0	3067	4
82	3	2986	81	3067	37	0	3104	4
83	3	3022	82	3104	37	0	3141	4
84	3	3058	83	3141	37	0	3178	4
85	3	3094	84	3178	37	0	3215	4
86	3	3130	85	3215	37	0	3252	4
87	3	3166	86	3252	37	0	3289	4
88	3	3202	87	3289	37	0	3326	4
89	3	3238	88	3326	37	0	3363	4
90	3	3274	89	3363	37	0	3400	4
91	3	3310	90	3400	37	0	3437	4
92	3	3346	91	3437	37	0	3474	4
93	3	3382	92	3474	37	0	3511	4
94	3	3418	93	3511	37	0	3548	4
95	3	3454	94	3548	37	0	3585	4
96	3	3490	95	3585	37	0	3622	4
97	3	3526	96	3622	37	0	3659	4
98	3	3562	97	3659	37	0	3696	4
99	3	3598	98	3696	37	0	3733	4
100	3	3634	99	3733	37	0	3770	4

MC4 =

1	4	107	0	107	43	107	150	5
2	4	144	6	150	43	0	193	5
3	4	181	12	193	43	0	236	5
4	4	218	18	236	43	0	279	5
5	4	255	24	279	43	0	322	5
6	4	292	30	322	43	0	365	5
7	4	329	36	365	43	0	408	5
8	4	366	42	408	43	0	451	5
9	4	403	48	451	43	0	494	5
10	4	440	54	494	43	0	537	5
11	4	477	60	537	43	0	580	5
12	4	514	66	580	43	0	623	5
13	4	551	72	623	43	0	666	5
14	4	588	78	666	43	0	709	5
15	4	625	84	709	43	0	752	5
16	4	662	90	752	43	0	795	5
17	4	699	96	795	43	0	838	5
18	4	736	102	838	43	0	881	5
19	4	773	108	881	43	0	924	5
20	4	810	114	924	43	0	967	5
21	4	847	120	967	43	0	1010	5
22	4	884	126	1010	43	0	1053	5
23	4	921	132	1053	43	0	1096	5
24	4	958	138	1096	43	0	1139	5
25	4	995	144	1139	43	0	1182	5
26	4	1032	150	1182	43	0	1225	5
27	4	1069	156	1225	43	0	1268	5
28	4	1106	162	1268	43	0	1311	5
29	4	1143	168	1311	43	0	1354	5
30	4	1180	174	1354	43	0	1397	5
31	4	1217	180	1397	43	0	1440	5
32	4	1254	186	1440	43	0	1483	5
33	4	1291	192	1483	43	0	1526	5
34	4	1328	198	1526	43	0	1569	5
35	4	1365	204	1569	43	0	1612	5
36	4	1402	210	1612	43	0	1655	5
37	4	1439	216	1655	43	0	1698	5
38	4	1476	222	1698	43	0	1741	5
39	4	1513	228	1741	43	0	1784	5
40	4	1550	234	1784	43	0	1827	5
41	4	1587	240	1827	43	0	1870	5
42	4	1624	246	1870	43	0	1913	5

43	4	1661	252	1913	43	0	1956	5
44	4	1698	258	1956	43	0	1999	5
45	4	1735	264	1999	43	0	2042	5
46	4	1772	270	2042	43	0	2085	5
47	4	1809	276	2085	43	0	2128	5
48	4	1846	282	2128	43	0	2171	5
49	4	1883	288	2171	43	0	2214	5
50	4	1920	294	2214	43	0	2257	5
51	4	1957	300	2257	43	0	2300	5
52	4	1994	306	2300	43	0	2343	5
53	4	2031	312	2343	43	0	2386	5
54	4	2068	318	2386	43	0	2429	5
55	4	2105	324	2429	43	0	2472	5
56	4	2142	330	2472	43	0	2515	5
57	4	2179	336	2515	43	0	2558	5
58	4	2216	342	2558	43	0	2601	5
59	4	2253	348	2601	43	0	2644	5
60	4	2290	354	2644	43	0	2687	5
61	4	2327	360	2687	43	0	2730	5
62	4	2364	366	2730	43	0	2773	5
63	4	2401	372	2773	43	0	2816	5
64	4	2438	378	2816	43	0	2859	5
65	4	2475	384	2859	43	0	2902	5
66	4	2512	390	2902	43	0	2945	5
67	4	2549	396	2945	43	0	2988	5
68	4	2586	402	2988	43	0	3031	5
69	4	2623	408	3031	43	0	3074	5
70	4	2660	414	3074	43	0	3117	5
71	4	2697	420	3117	43	0	3160	5
72	4	2734	426	3160	43	0	3203	5
73	4	2771	432	3203	43	0	3246	5
74	4	2808	438	3246	43	0	3289	5
75	4	2845	444	3289	43	0	3332	5
76	4	2882	450	3332	43	0	3375	5
77	4	2919	456	3375	43	0	3418	5
78	4	2956	462	3418	43	0	3461	5
79	4	2993	468	3461	43	0	3504	5
80	4	3030	474	3504	43	0	3547	5
81	4	3067	480	3547	43	0	3590	5
82	4	3104	486	3590	43	0	3633	5
83	4	3141	492	3633	43	0	3676	5
84	4	3178	498	3676	43	0	3719	5
85	4	3215	504	3719	43	0	3762	5
86	4	3252	510	3762	43	0	3805	5
87	4	3289	516	3805	43	0	3848	5
88	4	3326	522	3848	43	0	3891	5
89	4	3363	528	3891	43	0	3934	5
90	4	3400	534	3934	43	0	3977	5
91	4	3437	540	3977	43	0	4020	5
92	4	3474	546	4020	43	0	4063	5
93	4	3511	552	4063	43	0	4106	5
94	4	3548	558	4106	43	0	4149	5
95	4	3585	564	4149	43	0	4192	5
96	4	3622	570	4192	43	0	4235	5
97	4	3659	576	4235	43	0	4278	5
98	4	3696	582	4278	43	0	4321	5
99	4	3733	588	4321	43	0	4364	5
100	4	3770	594	4364	43	0	4407	5

MC5 =

1	5	150	0	150	42	150	192	0
2	5	193	0	193	42	1	235	0
3	5	236	0	236	42	1	278	0
4	5	279	0	279	42	1	321	0
5	5	322	0	322	42	1	364	0
6	5	365	0	365	42	1	407	0
7	5	408	0	408	42	1	450	0
8	5	451	0	451	42	1	493	0
9	5	494	0	494	42	1	536	0
10	5	537	0	537	42	1	579	0
11	5	580	0	580	42	1	622	0
12	5	623	0	623	42	1	665	0
13	5	666	0	666	42	1	708	0
14	5	709	0	709	42	1	751	0
15	5	752	0	752	42	1	794	0
16	5	795	0	795	42	1	837	0
17	5	838	0	838	42	1	880	0
18	5	881	0	881	42	1	923	0
19	5	924	0	924	42	1	966	0
20	5	967	0	967	42	1	1009	0
21	5	1010	0	1010	42	1	1052	0
22	5	1053	0	1053	42	1	1095	0
23	5	1096	0	1096	42	1	1138	0
24	5	1139	0	1139	42	1	1181	0
25	5	1182	0	1182	42	1	1224	0
26	5	1225	0	1225	42	1	1267	0
27	5	1268	0	1268	42	1	1310	0
28	5	1311	0	1311	42	1	1353	0
29	5	1354	0	1354	42	1	1396	0
30	5	1397	0	1397	42	1	1439	0
31	5	1440	0	1440	42	1	1482	0
32	5	1483	0	1483	42	1	1525	0
33	5	1526	0	1526	42	1	1568	0
34	5	1569	0	1569	42	1	1611	0
35	5	1612	0	1612	42	1	1654	0
36	5	1655	0	1655	42	1	1697	0
37	5	1698	0	1698	42	1	1740	0
38	5	1741	0	1741	42	1	1783	0
39	5	1784	0	1784	42	1	1826	0
40	5	1827	0	1827	42	1	1869	0
41	5	1870	0	1870	42	1	1912	0
42	5	1913	0	1913	42	1	1955	0

43	5	1956	0	1956	42	1	1998	0
44	5	1999	0	1999	42	1	2041	0
45	5	2042	0	2042	42	1	2084	0
46	5	2085	0	2085	42	1	2127	0
47	5	2128	0	2128	42	1	2170	0
48	5	2171	0	2171	42	1	2213	0
49	5	2214	0	2214	42	1	2256	0
50	5	2257	0	2257	42	1	2299	0
51	5	2300	0	2300	42	1	2342	0
52	5	2343	0	2343	42	1	2385	0
53	5	2386	0	2386	42	1	2428	0
54	5	2429	0	2429	42	1	2471	0
55	5	2472	0	2472	42	1	2514	0
56	5	2515	0	2515	42	1	2557	0
57	5	2558	0	2558	42	1	2600	0
58	5	2601	0	2601	42	1	2643	0
59	5	2644	0	2644	42	1	2686	0
60	5	2687	0	2687	42	1	2729	0
61	5	2730	0	2730	42	1	2772	0
62	5	2773	0	2773	42	1	2815	0
63	5	2816	0	2816	42	1	2858	0
64	5	2859	0	2859	42	1	2901	0
65	5	2902	0	2902	42	1	2944	0
66	5	2945	0	2945	42	1	2987	0
67	5	2988	0	2988	42	1	3030	0
68	5	3031	0	3031	42	1	3073	0
69	5	3074	0	3074	42	1	3116	0
70	5	3117	0	3117	42	1	3159	0
71	5	3160	0	3160	42	1	3202	0
72	5	3203	0	3203	42	1	3245	0
73	5	3246	0	3246	42	1	3288	0
74	5	3289	0	3289	42	1	3331	0
75	5	3332	0	3332	42	1	3374	0
76	5	3375	0	3375	42	1	3417	0
77	5	3418	0	3418	42	1	3460	0
78	5	3461	0	3461	42	1	3503	0
79	5	3504	0	3504	42	1	3546	0
80	5	3547	0	3547	42	1	3589	0
81	5	3590	0	3590	42	1	3632	0
82	5	3633	0	3633	42	1	3675	0
83	5	3676	0	3676	42	1	3718	0
84	5	3719	0	3719	42	1	3761	0
85	5	3762	0	3762	42	1	3804	0
86	5	3805	0	3805	42	1	3847	0
87	5	3848	0	3848	42	1	3890	0
88	5	3891	0	3891	42	1	3933	0
89	5	3934	0	3934	42	1	3976	0
90	5	3977	0	3977	42	1	4019	0
91	5	4020	0	4020	42	1	4062	0
92	5	4063	0	4063	42	1	4105	0
93	5	4106	0	4106	42	1	4148	0
94	5	4149	0	4149	42	1	4191	0
95	5	4192	0	4192	42	1	4234	0
96	5	4235	0	4235	42	1	4277	0
97	5	4278	0	4278	42	1	4320	0
98	5	4321	0	4321	42	1	4363	0
99	5	4364	0	4364	42	1	4406	0
100	5	4407	0	4407	42	1	4449	0

L319 heated-1

Cmax = 408

MC1 =

1	1	0	0	0	68	0	68	2
2	1	0	68	68	68	0	136	2
3	1	0	136	136	68	0	204	2
4	1	0	204	204	68	0	272	2
5	1	0	272	272	68	0	340	2

MC2 =

1	2	68	0	68	68	68	136	0
2	2	136	0	136	68	0	204	0
3	2	204	0	204	68	0	272	0
4	2	272	0	272	68	0	340	0
5	2	340	0	340	68	0	408	0

L319 heated-2

Cmax = 748

MC1 =

1	1	0	0	0	68	0	68	2
2	1	0	68	68	68	0	136	2
3	1	0	136	136	68	0	204	2
4	1	0	204	204	68	0	272	2
5	1	0	272	272	68	0	340	2
6	1	0	340	340	68	0	408	2
7	1	0	408	408	68	0	476	2
8	1	0	476	476	68	0	544	2
9	1	0	544	544	68	0	612	2
10	1	0	612	612	68	0	680	2

MC2 =

1	2	68	0	68	68	68	136	0
2	2	136	0	136	68	0	204	0
3	2	204	0	204	68	0	272	0
4	2	272	0	272	68	0	340	0
5	2	340	0	340	68	0	408	0
6	2	408	0	408	68	0	476	0
7	2	476	0	476	68	0	544	0
8	2	544	0	544	68	0	612	0
9	2	612	0	612	68	0	680	0
10	2	680	0	680	68	0	748	0

L319 heated-3

Cmax = 1768

MC1 =

1	1	0	0	0	68	0	68	2
2	1	0	68	68	68	0	136	2
3	1	0	136	136	68	0	204	2
4	1	0	204	204	68	0	272	2
5	1	0	272	272	68	0	340	2
6	1	0	340	340	68	0	408	2
7	1	0	408	408	68	0	476	2
8	1	0	476	476	68	0	544	2
9	1	0	544	544	68	0	612	2
10	1	0	612	612	68	0	680	2
11	1	0	680	680	68	0	748	2
12	1	0	748	748	68	0	816	2
13	1	0	816	816	68	0	884	2
14	1	0	884	884	68	0	952	2
15	1	0	952	952	68	0	1020	2
16	1	0	1020	1020	68	0	1088	2
17	1	0	1088	1088	68	0	1156	2
18	1	0	1156	1156	68	0	1224	2
19	1	0	1224	1224	68	0	1292	2
20	1	0	1292	1292	68	0	1360	2
21	1	0	1360	1360	68	0	1428	2
22	1	0	1428	1428	68	0	1496	2
23	1	0	1496	1496	68	0	1564	2
24	1	0	1564	1564	68	0	1632	2
25	1	0	1632	1632	68	0	1700	2

MC2 =

1	2	68	0	68	68	68	136	0
2	2	136	0	136	68	0	204	0
3	2	204	0	204	68	0	272	0
4	2	272	0	272	68	0	340	0
5	2	340	0	340	68	0	408	0
6	2	408	0	408	68	0	476	0
7	2	476	0	476	68	0	544	0
8	2	544	0	544	68	0	612	0
9	2	612	0	612	68	0	680	0
10	2	680	0	680	68	0	748	0
11	2	748	0	748	68	0	816	0
12	2	816	0	816	68	0	884	0
13	2	884	0	884	68	0	952	0
14	2	952	0	952	68	0	1020	0
15	2	1020	0	1020	68	0	1088	0
16	2	1088	0	1088	68	0	1156	0
17	2	1156	0	1156	68	0	1224	0
18	2	1224	0	1224	68	0	1292	0
19	2	1292	0	1292	68	0	1360	0
20	2	1360	0	1360	68	0	1428	0
21	2	1428	0	1428	68	0	1496	0
22	2	1496	0	1496	68	0	1564	0
23	2	1564	0	1564	68	0	1632	0
24	2	1632	0	1632	68	0	1700	0
25	2	1700	0	1700	68	0	1768	0

L319 heated-4

Cmax = 3468

MC1 =

1	1	0	0	0	68	0	68	2
2	1	0	68	68	68	0	136	2
3	1	0	136	136	68	0	204	2
4	1	0	204	204	68	0	272	2
5	1	0	272	272	68	0	340	2
6	1	0	340	340	68	0	408	2
7	1	0	408	408	68	0	476	2
8	1	0	476	476	68	0	544	2
9	1	0	544	544	68	0	612	2
10	1	0	612	612	68	0	680	2
11	1	0	680	680	68	0	748	2
12	1	0	748	748	68	0	816	2
13	1	0	816	816	68	0	884	2
14	1	0	884	884	68	0	952	2
15	1	0	952	952	68	0	1020	2
16	1	0	1020	1020	68	0	1088	2
17	1	0	1088	1088	68	0	1156	2
18	1	0	1156	1156	68	0	1224	2
19	1	0	1224	1224	68	0	1292	2
20	1	0	1292	1292	68	0	1360	2
21	1	0	1360	1360	68	0	1428	2
22	1	0	1428	1428	68	0	1496	2
23	1	0	1496	1496	68	0	1564	2
24	1	0	1564	1564	68	0	1632	2
25	1	0	1632	1632	68	0	1700	2
26	1	0	1700	1700	68	0	1768	2
27	1	0	1768	1768	68	0	1836	2
28	1	0	1836	1836	68	0	1904	2
29	1	0	1904	1904	68	0	1972	2
30	1	0	1972	1972	68	0	2040	2
31	1	0	2040	2040	68	0	2108	2
32	1	0	2108	2108	68	0	2176	2
33	1	0	2176	2176	68	0	2244	2
34	1	0	2244	2244	68	0	2312	2
35	1	0	2312	2312	68	0	2380	2
36	1	0	2380	2380	68	0	2448	2
37	1	0	2448	2448	68	0	2516	2
38	1	0	2516	2516	68	0	2584	2
39	1	0	2584	2584	68	0	2652	2
40	1	0	2652	2652	68	0	2720	2
41	1	0	2720	2720	68	0	2788	2
42	1	0	2788	2788	68	0	2856	2
43	1	0	2856	2856	68	0	2924	2
44	1	0	2924	2924	68	0	2992	2
45	1	0	2992	2992	68	0	3060	2
46	1	0	3060	3060	68	0	3128	2
47	1	0	3128	3128	68	0	3196	2
48	1	0	3196	3196	68	0	3264	2
49	1	0	3264	3264	68	0	3332	2

50	1	0	3332	3332	68	0	3400	2

MC2 =

1	2	68	0	68	68	68	136	0
2	2	136	0	136	68	0	204	0
3	2	204	0	204	68	0	272	0
4	2	272	0	272	68	0	340	0
5	2	340	0	340	68	0	408	0
6	2	408	0	408	68	0	476	0
7	2	476	0	476	68	0	544	0
8	2	544	0	544	68	0	612	0
9	2	612	0	612	68	0	680	0
10	2	680	0	680	68	0	748	0
11	2	748	0	748	68	0	816	0
12	2	816	0	816	68	0	884	0
13	2	884	0	884	68	0	952	0
14	2	952	0	952	68	0	1020	0
15	2	1020	0	1020	68	0	1088	0
16	2	1088	0	1088	68	0	1156	0
17	2	1156	0	1156	68	0	1224	0
18	2	1224	0	1224	68	0	1292	0
19	2	1292	0	1292	68	0	1360	0
20	2	1360	0	1360	68	0	1428	0
21	2	1428	0	1428	68	0	1496	0
22	2	1496	0	1496	68	0	1564	0
23	2	1564	0	1564	68	0	1632	0
24	2	1632	0	1632	68	0	1700	0
25	2	1700	0	1700	68	0	1768	0
26	2	1768	0	1768	68	0	1836	0
27	2	1836	0	1836	68	0	1904	0
28	2	1904	0	1904	68	0	1972	0
29	2	1972	0	1972	68	0	2040	0
30	2	2040	0	2040	68	0	2108	0
31	2	2108	0	2108	68	0	2176	0
32	2	2176	0	2176	68	0	2244	0
33	2	2244	0	2244	68	0	2312	0
34	2	2312	0	2312	68	0	2380	0
35	2	2380	0	2380	68	0	2448	0
36	2	2448	0	2448	68	0	2516	0
37	2	2516	0	2516	68	0	2584	0
38	2	2584	0	2584	68	0	2652	0
39	2	2652	0	2652	68	0	2720	0
40	2	2720	0	2720	68	0	2788	0
41	2	2788	0	2788	68	0	2856	0
42	2	2856	0	2856	68	0	2924	0
43	2	2924	0	2924	68	0	2992	0
44	2	2992	0	2992	68	0	3060	0
45	2	3060	0	3060	68	0	3128	0
46	2	3128	0	3128	68	0	3196	0
47	2	3196	0	3196	68	0	3264	0
48	2	3264	0	3264	68	0	3332	0
49	2	3332	0	3332	68	0	3400	0
50	2	3400	0	3400	68	0	3468	0

L319 heated-5

Cmax = 3468

MC1 =

1	1	0	0	0	68	0	68	2
2	1	0	68	68	68	0	136	2
3	1	0	136	136	68	0	204	2
4	1	0	204	204	68	0	272	2
5	1	0	272	272	68	0	340	2
6	1	0	340	340	68	0	408	2
7	1	0	408	408	68	0	476	2
8	1	0	476	476	68	0	544	2
9	1	0	544	544	68	0	612	2
10	1	0	612	612	68	0	680	2
11	1	0	680	680	68	0	748	2
12	1	0	748	748	68	0	816	2
13	1	0	816	816	68	0	884	2
14	1	0	884	884	68	0	952	2
15	1	0	952	952	68	0	1020	2
16	1	0	1020	1020	68	0	1088	2
17	1	0	1088	1088	68	0	1156	2
18	1	0	1156	1156	68	0	1224	2
19	1	0	1224	1224	68	0	1292	2
20	1	0	1292	1292	68	0	1360	2
21	1	0	1360	1360	68	0	1428	2
22	1	0	1428	1428	68	0	1496	2
23	1	0	1496	1496	68	0	1564	2
24	1	0	1564	1564	68	0	1632	2
25	1	0	1632	1632	68	0	1700	2
26	1	0	1700	1700	68	0	1768	2
27	1	0	1768	1768	68	0	1836	2
28	1	0	1836	1836	68	0	1904	2
29	1	0	1904	1904	68	0	1972	2
30	1	0	1972	1972	68	0	2040	2
31	1	0	2040	2040	68	0	2108	2
32	1	0	2108	2108	68	0	2176	2
33	1	0	2176	2176	68	0	2244	2
34	1	0	2244	2244	68	0	2312	2
35	1	0	2312	2312	68	0	2380	2
36	1	0	2380	2380	68	0	2448	2
37	1	0	2448	2448	68	0	2516	2
38	1	0	2516	2516	68	0	2584	2
39	1	0	2584	2584	68	0	2652	2
40	1	0	2652	2652	68	0	2720	2
41	1	0	2720	2720	68	0	2788	2
42	1	0	2788	2788	68	0	2856	2
43	1	0	2856	2856	68	0	2924	2
44	1	0	2924	2924	68	0	2992	2
45	1	0	2992	2992	68	0	3060	2
46	1	0	3060	3060	68	0	3128	2
47	1	0	3128	3128	68	0	3196	2
48	1	0	3196	3196	68	0	3264	2
49	1	0	3264	3264	68	0	3332	2
50	1	0	3332	3332	68	0	3400	2

MC2 =

1	2	68	0	68	68	68	136	0
2	2	136	0	136	68	0	204	0
3	2	204	0	204	68	0	272	0
4	2	272	0	272	68	0	340	0
5	2	340	0	340	68	0	408	0
6	2	408	0	408	68	0	476	0
7	2	476	0	476	68	0	544	0
8	2	544	0	544	68	0	612	0
9	2	612	0	612	68	0	680	0
10	2	680	0	680	68	0	748	0
11	2	748	0	748	68	0	816	0
12	2	816	0	816	68	0	884	0
13	2	884	0	884	68	0	952	0
14	2	952	0	952	68	0	1020	0
15	2	1020	0	1020	68	0	1088	0
16	2	1088	0	1088	68	0	1156	0
17	2	1156	0	1156	68	0	1224	0
18	2	1224	0	1224	68	0	1292	0
19	2	1292	0	1292	68	0	1360	0
20	2	1360	0	1360	68	0	1428	0
21	2	1428	0	1428	68	0	1496	0
22	2	1496	0	1496	68	0	1564	0
23	2	1564	0	1564	68	0	1632	0
24	2	1632	0	1632	68	0	1700	0
25	2	1700	0	1700	68	0	1768	0
26	2	1768	0	1768	68	0	1836	0
27	2	1836	0	1836	68	0	1904	0
28	2	1904	0	1904	68	0	1972	0
29	2	1972	0	1972	68	0	2040	0
30	2	2040	0	2040	68	0	2108	0
31	2	2108	0	2108	68	0	2176	0
32	2	2176	0	2176	68	0	2244	0
33	2	2244	0	2244	68	0	2312	0
34	2	2312	0	2312	68	0	2380	0
35	2	2380	0	2380	68	0	2448	0
36	2	2448	0	2448	68	0	2516	0
37	2	2516	0	2516	68	0	2584	0
38	2	2584	0	2584	68	0	2652	0
39	2	2652	0	2652	68	0	2720	0
40	2	2720	0	2720	68	0	2788	0
41	2	2788	0	2788	68	0	2856	0
42	2	2856	0	2856	68	0	2924	0
43	2	2924	0	2924	68	0	2992	0
44	2	2992	0	2992	68	0	3060	0
45	2	3060	0	3060	68	0	3128	0
46	2	3128	0	3128	68	0	3196	0
47	2	3196	0	3196	68	0	3264	0
48	2	3264	0	3264	68	0	3332	0
49	2	3332	0	3332	68	0	3400	0
50	2	3400	0	3400	68	0	3468	0

L319 heated-6

Cmax = 6800

MC1 =

1	1	0	0	0	68	0	68	2
2	1	0	68	68	68	0	136	2
3	1	0	136	136	68	0	204	2
4	1	0	204	204	68	0	272	2
5	1	0	272	272	68	0	340	2
6	1	0	340	340	68	0	408	2
7	1	0	408	408	68	0	476	2
8	1	0	476	476	68	0	544	2
9	1	0	544	544	68	0	612	2
10	1	0	612	612	68	0	680	2
11	1	0	680	680	68	0	748	2
12	1	0	748	748	68	0	816	2
13	1	0	816	816	68	0	884	2
14	1	0	884	884	68	0	952	2
15	1	0	952	952	68	0	1020	2
16	1	0	1020	1020	68	0	1088	2
17	1	0	1088	1088	68	0	1156	2
18	1	0	1156	1156	68	0	1224	2
19	1	0	1224	1224	68	0	1292	2
20	1	0	1292	1292	68	0	1360	2
21	1	0	1360	1360	68	0	1428	2
22	1	0	1428	1428	68	0	1496	2
23	1	0	1496	1496	68	0	1564	2
24	1	0	1564	1564	68	0	1632	2
25	1	0	1632	1632	68	0	1700	2
26	1	0	1700	1700	68	0	1768	2
27	1	0	1768	1768	68	0	1836	2
28	1	0	1836	1836	68	0	1904	2
29	1	0	1904	1904	68	0	1972	2
30	1	0	1972	1972	68	0	2040	2
31	1	0	2040	2040	68	0	2108	2
32	1	0	2108	2108	68	0	2176	2
33	1	0	2176	2176	68	0	2244	2
34	1	0	2244	2244	68	0	2312	2
35	1	0	2312	2312	68	0	2380	2
36	1	0	2380	2380	68	0	2448	2
37	1	0	2448	2448	68	0	2516	2
38	1	0	2516	2516	68	0	2584	2
39	1	0	2584	2584	68	0	2652	2
40	1	0	2652	2652	68	0	2720	2
41	1	0	2720	2720	68	0	2788	2
42	1	0	2788	2788	68	0	2856	2
43	1	0	2856	2856	68	0	2924	2

44	1	0	2924	2924	68	0	2992	2
45	1	0	2992	2992	68	0	3060	2
46	1	0	3060	3060	68	0	3128	2
47	1	0	3128	3128	68	0	3196	2
48	1	0	3196	3196	68	0	3264	2
49	1	0	3264	3264	68	0	3332	2
50	1	0	3332	3332	68	0	3400	2
51	1	0	3400	3400	68	0	3468	2
52	1	0	3468	3468	68	0	3536	2
53	1	0	3536	3536	68	0	3604	2
54	1	0	3604	3604	68	0	3672	2
55	1	0	3672	3672	68	0	3740	2
56	1	0	3740	3740	68	0	3808	2
57	1	0	3808	3808	68	0	3876	2
58	1	0	3876	3876	68	0	3944	2
59	1	0	3944	3944	68	0	4012	2
60	1	0	4012	4012	68	0	4080	2
61	1	0	4080	4080	68	0	4148	2
62	1	0	4148	4148	68	0	4216	2
63	1	0	4216	4216	68	0	4284	2
64	1	0	4284	4284	68	0	4352	2
65	1	0	4352	4352	68	0	4420	2
66	1	0	4420	4420	68	0	4488	2
67	1	0	4488	4488	68	0	4556	2
68	1	0	4556	4556	68	0	4624	2
69	1	0	4624	4624	68	0	4692	2
70	1	0	4692	4692	68	0	4760	2
71	1	0	4760	4760	68	0	4828	2
72	1	0	4828	4828	68	0	4896	2
73	1	0	4896	4896	68	0	4964	2
74	1	0	4964	4964	68	0	5032	2
75	1	0	5032	5032	68	0	5100	2
76	1	0	5100	5100	68	0	5168	2
77	1	0	5168	5168	68	0	5236	2
78	1	0	5236	5236	68	0	5304	2
79	1	0	5304	5304	68	0	5372	2
80	1	0	5372	5372	68	0	5440	2
81	1	0	5440	5440	68	0	5508	2
82	1	0	5508	5508	68	0	5576	2
83	1	0	5576	5576	68	0	5644	2
84	1	0	5644	5644	68	0	5712	2
85	1	0	5712	5712	68	0	5780	2
86	1	0	5780	5780	68	0	5848	2
87	1	0	5848	5848	68	0	5916	2
88	1	0	5916	5916	68	0	5984	2
89	1	0	5984	5984	68	0	6052	2
90	1	0	6052	6052	68	0	6120	2
91	1	0	6120	6120	68	0	6188	2
92	1	0	6188	6188	68	0	6256	2
93	1	0	6256	6256	68	0	6324	2
94	1	0	6324	6324	68	0	6392	2
95	1	0	6392	6392	68	0	6460	2
96	1	0	6460	6460	68	0	6528	2
97	1	0	6528	6528	68	0	6596	2
98	1	0	6596	6596	68	0	6664	2
99	1	0	6664	6664	68	0	6732	2
100	1	0	6732	6732	68	0	6800	2

MC2 =

1	2	68	0	68	68	68	136	0
2	2	136	0	136	68	0	204	0
3	2	204	0	204	68	0	272	0
4	2	272	0	272	68	0	340	0
5	2	340	0	340	68	0	408	0
6	2	408	0	408	68	0	476	0
7	2	476	0	476	68	0	544	0
8	2	544	0	544	68	0	612	0
9	2	612	0	612	68	0	680	0
10	2	680	0	680	68	0	748	0
11	2	748	0	748	68	0	816	0
12	2	816	0	816	68	0	884	0
13	2	884	0	884	68	0	952	0
14	2	952	0	952	68	0	1020	0
15	2	1020	0	1020	68	0	1088	0
16	2	1088	0	1088	68	0	1156	0
17	2	1156	0	1156	68	0	1224	0
18	2	1224	0	1224	68	0	1292	0
19	2	1292	0	1292	68	0	1360	0
20	2	1360	0	1360	68	0	1428	0
21	2	1428	0	1428	68	0	1496	0
22	2	1496	0	1496	68	0	1564	0
23	2	1564	0	1564	68	0	1632	0
24	2	1632	0	1632	68	0	1700	0
25	2	1700	0	1700	68	0	1768	0
26	2	1768	0	1768	68	0	1836	0
27	2	1836	0	1836	68	0	1904	0
28	2	1904	0	1904	68	0	1972	0
29	2	1972	0	1972	68	0	2040	0
30	2	2040	0	2040	68	0	2108	0
31	2	2108	0	2108	68	0	2176	0
32	2	2176	0	2176	68	0	2244	0
33	2	2244	0	2244	68	0	2312	0
34	2	2312	0	2312	68	0	2380	0
35	2	2380	0	2380	68	0	2448	0
36	2	2448	0	2448	68	0	2516	0
37	2	2516	0	2516	68	0	2584	0
38	2	2584	0	2584	68	0	2652	0
39	2	2652	0	2652	68	0	2720	0
40	2	2720	0	2720	68	0	2788	0
41	2	2788	0	2788	68	0	2856	0
42	2	2856	0	2856	68	0	2924	0
43	2	2924	0	2924	68	0	2992	0
44	2	2992	0	2992	68	0	3060	0
45	2	3060	0	3060	68	0	3128	0
46	2	3128	0	3128	68	0	3196	0
47	2	3196	0	3196	68	0	3264	0
48	2	3264	0	3264	68	0	3332	0
49	2	3332	0	3332	68	0	3400	0
50	2	3400	0	3400	68	0	3468	0
51	2	3468	0	3468	68	0	3536	0
52	2	3536	0	3536	68	0	3604	0
53	2	3604	0	3604	68	0	3672	0
54	2	3672	0	3672	68	0	3740	0
55	2	3740	0	3740	68	0	3808	0
56	2	3808	0	3808	68	0	3876	0
57	2	3876	0	3876	68	0	3944	0
58	2	3944	0	3944	68	0	4012	0
59	2	4012	0	4012	68	0	4080	0
60	2	4080	0	4080	68	0	4148	0
61	2	4148	0	4148	68	0	4216	0
62	2	4216	0	4216	68	0	4284	0
63	2	4284	0	4284	68	0	4352	0
64	2	4352	0	4352	68	0	4420	0
65	2	4420	0	4420	68	0	4488	0
66	2	4488	0	4488	68	0	4556	0
67	2	4556	0	4556	68	0	4624	0
68	2	4624	0	4624	68	0	4692	0
69	2	4692	0	4692	68	0	4760	0
70	2	4760	0	4760	68	0	4828	0
71	2	4828	0	4828	68	0	4896	0
72	2	4896	0	4896	68	0	4964	0
73	2	4964	0	4964	68	0	5032	0
74	2	5032	0	5032	68	0	5100	0
75	2	5100	0	5100	68	0	5168	0
76	2	5168	0	5168	68	0	5236	0
77	2	5236	0	5236	68	0	5304	0
78	2	5304	0	5304	68	0	5372	0
79	2	5372	0	5372	68	0	5440	0
80	2	5440	0	5440	68	0	5508	0
81	2	5508	0	5508	68	0	5576	0
82	2	5576	0	5576	68	0	5644	0
83	2	5644	0	5644	68	0	5712	0
84	2	5712	0	5712	68	0	5780	0
85	2	5780	0	5780	68	0	5848	0
86	2	5848	0	5848	68	0	5916	0
87	2	5916	0	5916	68	0	5984	0
88	2	5984	0	5984	68	0	6052	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0
0	0	0	0	0	0	0	6000	0

L319 non heated-1

Cmax = 324

MC1 =

1	1	0	0	0	54	0	54	2
2	1	0	54	54	54	0	108	2
3	1	0	108	108	54	0	162	2
4	1	0	162	162	54	0	216	2
5	1	0	216	216	54	0	270	2

MC2

1	2	54	0	54	54	54	108	0
2	2	108	0	108	54	0	162	0
3	2	162	0	162	54	0	216	0
4	2	216	0	216	54	0	270	0
5	2	270	0	270	54	0	324	0

L319 non heated-2

Cmax = 594

MC1 =

1	1	0	0	0	54	0	54	2
2	1	0	54	54	54	0	108	2
3	1	0	108	108	54	0	162	2
4	1	0	162	162	54	0	216	2
5	1	0	216	216	54	0	270	2
6	1	0	270	270	54	0	324	2
7	1	0	324	324	54	0	378	2
8	1	0	378	378	54	0	432	2
9	1	0	432	432	54	0	486	2
10	1	0	486	486	54	0	540	2

MC2 =

1	2	54	0	54	54	54	108	0
2	2	108	0	108	54	0	162	0
3	2	162	0	162	54	0	216	0
4	2	216	0	216	54	0	270	0
5	2	270	0	270	54	0	324	0
6	2	324	0	324	54	0	378	0
7	2	378	0	378	54	0	432	0
8	2	432	0	432	54	0	486	0
9	2	486	0	486	54	0	540	0
10	2	540	0	540	54	0	594	0

L319 non heated-3

Cmax = 1404

MC1 =

1	1	0	0	0	54	0	54	2
2	1	0	54	54	54	0	108	2

3	1	0	108	108	54	0	162	2
4	1	0	162	162	54	0	216	2
5	1	0	216	216	54	0	270	2
6	1	0	270	270	54	0	324	2
7	1	0	324	324	54	0	378	2
8	1	0	378	378	54	0	432	2
9	1	0	432	432	54	0	486	2
10	1	0	486	486	54	0	540	2
11	1	0	540	540	54	0	594	2
12	1	0	594	594	54	0	648	2
13	1	0	648	648	54	0	702	2
14	1	0	702	702	54	0	756	2
15	1	0	756	756	54	0	810	2
16	1	0	810	810	54	0	864	2
17	1	0	864	864	54	0	918	2
18	1	0	918	918	54	0	972	2
19	1	0	972	972	54	0	1026	2
20	1	0	1026	1026	54	0	1080	2
21	1	0	1080	1080	54	0	1134	2
22	1	0	1134	1134	54	0	1188	2
23	1	0	1188	1188	54	0	1242	2
24	1	0	1242	1242	54	0	1296	2
25	1	0	1296	1296	54	0	1350	2

MC2 =

1	2	54	0	54	54	54	108	0
2	2	108	0	108	54	0	162	0
3	2	162	0	162	54	0	216	0
4	2	216	0	216	54	0	270	0
5	2	270	0	270	54	0	324	0
6	2	324	0	324	54	0	378	0
7	2	378	0	378	54	0	432	0
8	2	432	0	432	54	0	486	0
9	2	486	0	486	54	0	540	0
10	2	540	0	540	54	0	594	0
11	2	594	0	594	54	0	648	0
12	2	648	0	648	54	0	702	0
13	2	702	0	702	54	0	756	0
14	2	756	0	756	54	0	810	0
15	2	810	0	810	54	0	864	0
16	2	864	0	864	54	0	918	0
17	2	918	0	918	54	0	972	0
18	2	972	0	972	54	0	1026	0
19	2	1026	0	1026	54	0	1080	0
20	2	1080	0	1080	54	0	1134	0
21	2	1134	0	1134	54	0	1188	0
22	2	1188	0	1188	54	0	1242	0
23	2	1242	0	1242	54	0	1296	0
24	2	1296	0	1296	54	0	1350	0
25	2	1350	0	1350	54	0	1404	0

L319 non heated-4

Cmax = 2754

MC1 =

1	1	0	0	0	54	0	54	2
2	1	0	54	54	54	0	108	2
3	1	0	108	108	54	0	162	2
4	1	0	162	162	54	0	216	2
5	1	0	216	216	54	0	270	2
6	1	0	270	270	54	0	324	2
7	1	0	324	324	54	0	378	2
8	1	0	378	378	54	0	432	2
9	1	0	432	432	54	0	486	2
10	1	0	486	486	54	0	540	2
11	1	0	540	540	54	0	594	2
12	1	0	594	594	54	0	648	2
13	1	0	648	648	54	0	702	2
14	1	0	702	702	54	0	756	2
15	1	0	756	756	54	0	810	2
16	1	0	810	810	54	0	864	2
17	1	0	864	864	54	0	918	2
18	1	0	918	918	54	0	972	2
19	1	0	972	972	54	0	1026	2
20	1	0	1026	1026	54	0	1080	2
21	1	0	1080	1080	54	0	1134	2
22	1	0	1134	1134	54	0	1188	2
23	1	0	1188	1188	54	0	1242	2
24	1	0	1242	1242	54	0	1296	2
25	1	0	1296	1296	54	0	1350	2
26	1	0	1350	1350	54	0	1404	2
27	1	0	1404	1404	54	0	1458	2
28	1	0	1458	1458	54	0	1512	2
29	1	0	1512	1512	54	0	1566	2
30	1	0	1566	1566	54	0	1620	2
31	1	0	1620	1620	54	0	1674	2
32	1	0	1674	1674	54	0	1728	2
33	1	0	1728	1728	54	0	1782	2
34	1	0	1782	1782	54	0	1836	2
35	1	0	1836	1836	54	0	1890	2
36	1	0	1890	1890	54	0	1944	2
37	1	0	1944	1944	54	0	1998	2
38	1	0	1998	1998	54	0	2052	2
39	1	0	2052	2052	54	0	2106	2
40	1	0	2106	2106	54	0	2160	2
41	1	0	2160	2160	54	0	2214	2
42	1	0	2214	2214	54	0	2268	2
43	1	0	2268	2268	54	0	2322	2
44	1	0	2322	2322	54	0	2376	2
45	1	0	2376	2376	54	0	2430	2
46	1	0	2430	2430	54	0	2484	2
47	1	0	2484	2484	54	0	2538	2
48	1	0	2538	2538	54	0	2592	2
49	1	0	2592	2592	54	0	2646	2
50	1	0	2646	2646	54	0	2700	2

MC2 =

1	2	54	0	54	54	54	108	0
2	2	108	0	108	54	0	162	0
3	2	162	0	162	54	0	216	0
4	2	216	0	216	54	0	270	0
5	2	270	0	270	54	0	324	0
6	2	324	0	324	54	0	378	0
7	2	378	0	378	54	0	432	0
8	2	432	0	432	54	0	486	0
9	2	486	0	486	54	0	540	0
10	2	540	0	540	54	0	594	0
11	2	594	0	594	54	0	648	0
12	2	648	0	648	54	0	702	0
13	2	702	0	702	54	0	756	0
14	2	756	0	756	54	0	810	0
15	2	810	0	810	54	0	864	0
16	2	864	0	864	54	0	918	0
17	2	918	0	918	54	0	972	0
18	2	972	0	972	54	0	1026	0
19	2	1026	0	1026	54	0	1080	0
20	2	1080	0	1080	54	0	1134	0
21	2	1134	0	1134	54	0	1188	0
22	2	1188	0	1188	54	0	1242	0
23	2	1242	0	1242	54	0	1296	0
24	2	1296	0	1296	54	0	1350	0
25	2	1350	0	1350	54	0	1404	0
26	2	1404	0	1404	54	0	1458	0
27	2	1458	0	1458	54	0	1512	0
28	2	1512	0	1512	54	0	1566	0
29	2	1566	0	1566	54	0	1620	0
30	2	1620	0	1620	54	0	1674	0
31	2	1674	0	1674	54	0	1728	0
32	2	1728	0	1728	54	0	1782	0
33	2	1782	0	1782	54	0	1836	0
34	2	1836	0	1836	54	0	1890	0
35	2	1890	0	1890	54	0	1944	0
36	2	1944	0	1944	54	0	1998	0
37	2	1998	0	1998	54	0	2052	0
38	2	2052	0	2052	54	0	2106	0
39	2	2106	0	2106	54	0	2160	0
40	2	2160	0	2160	54	0	2214	0
41	2	2214	0	2214	54	0	2268	0
42	2	2268	0	2268	54	0	2322	0
43	2	2322	0	2322	54	0	2376	0
44	2	2376	0	2376	54	0	2430	0
45	2	2430	0	2430	54	0	2484	0
46	2	2484	0	2484	54	0	2538	0
47	2	2538	0	2538	54	0	2592	0
48	2	2592	0	2592	54	0	2646	0
49	2	2646	0	2646	54	0	2700	0
50	2	2700	0	2700	54	0	2754	0

L319 non heated-5

Cmax = 4104

MC1 =

1	1	0	0	0	54	0	54	2
2	1	0	54	54	54	0	108	2
3	1	0	108	108	54	0	162	2
4	1	0	162	162	54	0	216	2
5	1	0	216	216	54	0	270	2
6	1	0	270	270	54	0	324	2
7	1	0	324	324	54	0	378	2
8	1	0	378	378	54	0	432	2
9	1	0	432	432	54	0	486	2
10	1	0	486	486	54	0	540	2
11	1	0	540	540	54	0	594	2
12	1	0	594	594	54	0	648	2
13	1	0	648	648	54	0	702	2
14	1	0	702	702	54	0	756	2
15	1	0	756	756	54	0	810	2
16	1	0	810	810	54	0	864	2
17	1	0	864	864	54	0	918	2
18	1	0	918	918	54	0	972	2
19	1	0	972	972	54	0	1026	2
20	1	0	1026	1026	54	0	1080	2
21	1	0	1080	1080	54	0	1134	2
22	1	0	1134	1134	54	0	1188	2
23	1	0	1188	1188	54	0	1242	2
24	1	0	1242	1242	54	0	1296	2
25	1	0	1296	1296	54	0	1350	2
26	1	0	1350	1350	54	0	1404	2
27	1	0	1404	1404	54	0	1458	2
28	1	0	1458	1458	54	0	1512	2
29	1	0	1512	1512	54	0	1566	2
30	1	0	1566	1566	54	0	1620	2
31	1	0	1620	1620	54	0	1674	2
32	1	0	1674	1674	54	0	1728	2
33	1	0	1728	1728	54	0	1782	2
34	1	0	1782	1782	54	0	1836	2
35	1	0	1836	1836	54	0	1890	2
36	1	0	1890	1890	54	0	1944	2
37	1	0	1944	1944	54	0	1998	2
38	1	0	1998	1998	54	0	2052	2
39	1	0	2052	2052	54	0	2106	2
40	1	0	2106	2106	54	0	2160	2
41	1	0	2160	2160	54	0	2214	2
42	1	0	2214	2214	54	0	2268	2
43	1	0	2268	2268	54	0	2322	2
44	1	0	2322	2322	54	0	2376	2
45	1	0	2376	2376	54	0	2430	2
46	1	0	2430	2430	54	0	2484	2

47	1	0	2484	2484	54	0	2538	2
48	1	0	2538	2538	54	0	2592	2
49	1	0	2592	2592	54	0	2646	2
50	1	0	2646	2646	54	0	2700	2
51	1	0	2700	2700	54	0	2754	2
52	1	0	2754	2754	54	0	2808	2
53	1	0	2808	2808	54	0	2862	2
54	1	0	2862	2862	54	0	2916	2
55	1	0	2916	2916	54	0	2970	2
56	1	0	2970	2970	54	0	3024	2
57	1	0	3024	3024	54	0	3078	2
58	1	0	3078	3078	54	0	3132	2
59	1	0	3132	3132	54	0	3186	2
60	1	0	3186	3186	54	0	3240	2
61	1	0	3240	3240	54	0	3294	2
62	1	0	3294	3294	54	0	3348	2
63	1	0	3348	3348	54	0	3402	2
64	1	0	3402	3402	54	0	3456	2
65	1	0	3456	3456	54	0	3510	2
66	1	0	3510	3510	54	0	3564	2
67	1	0	3564	3564	54	0	3618	2
68	1	0	3618	3618	54	0	3672	2
69	1	0	3672	3672	54	0	3726	2
70	1	0	3726	3726	54	0	3780	2
71	1	0	3780	3780	54	0	3834	2
72	1	0	3834	3834	54	0	3888	2
73	1	0	3888	3888	54	0	3942	2
74	1	0	3942	3942	54	0	3996	2
75	1	0	3996	3996	54	0	4050	2

MC2 =

1	2	54	0	54	54	54	108	0
2	2	108	0	108	54	0	162	0
3	2	162	0	162	54	0	216	0
4	2	216	0	216	54	0	270	0
5	2	270	0	270	54	0	324	0
6	2	324	0	324	54	0	378	0
7	2	378	0	378	54	0	432	0
8	2	432	0	432	54	0	486	0
9	2	486	0	486	54	0	540	0
10	2	540	0	540	54	0	594	0
11	2	594	0	594	54	0	648	0
12	2	648	0	648	54	0	702	0
13	2	702	0	702	54	0	756	0
14	2	756	0	756	54	0	810	0
15	2	810	0	810	54	0	864	0
16	2	864	0	864	54	0	918	0
17	2	918	0	918	54	0	972	0
18	2	972	0	972	54	0	1026	0
19	2	1026	0	1026	54	0	1080	0
20	2	1080	0	1080	54	0	1134	0
21	2	1134	0	1134	54	0	1188	0
22	2	1188	0	1188	54	0	1242	0
23	2	1242	0	1242	54	0	1296	0
24	2	1296	0	1296	54	0	1350	0
25	2	1350	0	1350	54	0	1404	0
26	2	1404	0	1404	54	0	1458	0
27	2	1458	0	1458	54	0	1512	0
28	2	1512	0	1512	54	0	1566	0
29	2	1566	0	1566	54	0	1620	0
30	2	1620	0	1620	54	0	1674	0
31	2	1674	0	1674	54	0	1728	0
32	2	1728	0	1728	54	0	1782	0
33	2	1782	0	1782	54	0	1836	0
34	2	1836	0	1836	54	0	1890	0
35	2	1890	0	1890	54	0	1944	0
36	2	1944	0	1944	54	0	1998	0
37	2	1998	0	1998	54	0	2052	0
38	2	2052	0	2052	54	0	2106	0
39	2	2106	0	2106	54	0	2160	0
40	2	2160	0	2160	54	0	2214	0
41	2	2214	0	2214	54	0	2268	0
42	2	2268	0	2268	54	0	2322	0
43	2	2322	0	2322	54	0	2376	0
44	2	2376	0	2376	54	0	2430	0
45	2	2430	0	2430	54	0	2484	0
46	2	2484	0	2484	54	0	2538	0
47	2	2538	0	2538	54	0	2592	0
48	2	2592	0	2592	54	0	2646	0
49	2	2646	0	2646	54	0	2700	0
50	2	2700	0	2700	54	0	2754	0
51	2	2754	0	2754	54	0	2808	0
52	2	2808	0	2808	54	0	2862	0
53	2	2862	0	2862	54	0	2916	0
54	2	2916	0	2916	54	0	2970	0
55	2	2970	0	2970	54	0	3024	0
56	2	3024	0	3024	54	0	3078	0
57	2	3078	0	3078	54	0	3132	0
58	2	3132	0	3132	54	0	3186	0
59	2	3186	0	3186	54	0	3240	0
60	2	3240	0	3240	54	0	3294	0
61	2	3294	0	3294	54	0	3348	0
62	2	3348	0	3348	54	0	3402	0
63	2	3402	0	3402	54	0	3456	0
64	2	3456	0	3456	54	0	3510	0
65	2	3510	0	3510	54	0	3564	0
66	2	3564	0	3564	54	0	3618	0
67	2	3618	0	3618	54	0	3672	0
68	2	3672	0	3672	54	0	3726	0
69	2	3726	0	3726	54	0	3780	0
70	2	3780	0	3780	54	0	3834	0
71	2	3834	0	3834	54	0	3888	0
72	2	3888	0	3888	54	0	3942	0
73	2	3942	0	3942	54	0	3996	0
74	2	3996	0	3996	54	0	4050	0
75	2	4050	0	4050	54	0	4104	0

L319 non heated-6

Cmax = 5454

MC1 =

1	1	0	0	0	54	0	54	2
2	1	0	54	54	54	0	108	2
3	1	0	108	108	54	0	162	2
4	1	0	162	162	54	0	216	2
5	1	0	216	216	54	0	270	2
6	1	0	270	270	54	0	324	2
7	1	0	324	324	54	0	378	2
8	1	0	378	378	54	0	432	2
9	1	0	432	432	54	0	486	2
10	1	0	486	486	54	0	540	2
11	1	0	540	540	54	0	594	2
12	1	0	594	594	54	0	648	2
13	1	0	648	648	54	0	702	2
14	1	0	702	702	54	0	756	2
15	1	0	756	756	54	0	810	2
16	1	0	810	810	54	0	864	2
17	1	0	864	864	54	0	918	2
18	1	0	918	918	54	0	972	2
19	1	0	972	972	54	0	1026	2
20	1	0	1026	1026	54	0	1080	2
21	1	0	1080	1080	54	0	1134	2
22	1	0	1134	1134	54	0	1188	2
23	1	0	1188	1188	54	0	1242	2
24	1	0	1242	1242	54	0	1296	2
25	1	0	1296	1296	54	0	1350	2
26	1	0	1350	1350	54	0	1404	2
27	1	0	1404	1404	54	0	1458	2
28	1	0	1458	1458	54	0	1512	2
29	1	0	1512	1512	54	0	1566	2
30	1	0	1566	1566	54	0	1620	2
31	1	0	1620	1620	54	0	1674	2
32	1	0	1674	1674	54	0	1728	2
33	1	0	1728	1728	54	0	1782	2
34	1	0	1782	1782	54	0	1836	2
35	1	0	1836	1836	54	0	1890	2
36	1	0	1890	1890	54	0	1944	2
37	1	0	1944	1944	54	0	1998	2
38	1	0	1998	1998	54	0	2052	2
39	1	0	2052	2052	54	0	2106	2
40	1	0	2106	2106	54	0	2160	2
41	1	0	2160	2160	54	0	2214	2
42	1	0	2214	2214	54	0	2268	2
43	1	0	2268	2268	54	0	2322	2
44	1	0	2322	2322	54	0	2376	2
45	1	0	2376	2376	54	0	2430	2
46	1	0	2430	2430	54	0	2484	2
47	1	0	2484	2484	54	0	2538	2
48	1	0	2538	2538	54	0	2592	2
49	1	0	2592	2592	54	0	2646	2
50	1	0	2646	2646	54	0	2700	2
51	1	0	2700	2700	54	0	2754	2
52	1	0	2754	2754	54	0	2808	2
53	1	0	2808	2808	54	0	2862	2
54	1	0	2862	2862	54	0	2916	2
55	1	0	2916	2916	54	0	2970	2
56	1	0	2970	2970	54	0	3024	2
57	1	0	3024	3024	54	0	3078	2
58	1	0	3078	3078	54	0	3132	2
59	1	0	3132	3132	54	0	3186	2
60	1	0	3186	3186	54	0	3240	2
61	1	0	3240	3240	54	0	3294	2
62	1	0	3294	3294	54	0	3348	2
63	1	0	3348	3348	54	0	3402	2
64	1	0	3402	3402	54	0	3456	2
65	1	0	3456	3456	54	0	3510	2
66	1	0	3510	3510	54	0	3564	2
67	1	0	3564	3564	54	0	3618	2
68	1	0	3618	3618	54	0	3672	2
69	1	0	3672	3672	54	0	3726	2
70	1	0	3726	3726	54	0	3780	2
71	1	0	3780	3780	54	0	3834	2
72	1	0	3834	3834	54	0	3888	2
73	1	0	3888	3888	54	0	3942	2
74	1	0	3942	3942	54	0	3996	2
75	1	0	3996	3996	54	0	4050	2
76	1	0	4050	4050	54	0	4104	2
77	1	0	4104	4104	54	0	4158	2
78	1	0	4158	4158	54	0	4212	2
79	1	0	4212	4212	54	0	4266	2
80	1	0	4266	4266	54	0	4320	2
81	1	0	4320	4320	54	0	4374	2
82	1	0	4374	4374	54	0	4428	2
83	1	0	4428	4428	54	0	4482	2
84	1	0	4482	4482	54	0	4536	2
85	1	0	4536	4536	54	0	4590	2
86	1	0	4590	4590	54	0	4644	2
87	1	0	4644	4644	54	0	4698	2
88	1	0	4698	4698	54	0	4752	2
89	1	0	4752	4752	54	0	4806	2
90	1	0	4806	4806	54	0	4860	2
91	1	0	4860	4860	54	0	4914	2
92	1	0	4914	4914	54	0	4968	2
93	1	0	4968	4968	54	0	5022	2
94	1	0	5022	5022	54	0	5076	2

95 1 0 5076 5076 54 0 5130 2
96 1 0 5130 5130 54 0 5184 2
97 1 0 5184 5184 54 0 5238 2
98 1 0 5238 5238 54 0 5292 2
99 1 0 5292 5292 54 0 5346 2
100 1 0 5346 5346 54 0 5400 2
MC2 =
1 2 54 0 54 54 54 108 0
2 2 108 0 108 54 0 162 0
3 2 162 0 162 54 0 216 0
4 2 216 0 216 54 0 270 0
5 2 270 0 270 54 0 324 0
6 2 324 0 324 54 0 378 0
7 2 378 0 378 54 0 432 0
8 2 432 0 432 54 0 486 0
9 2 486 0 486 54 0 540 0
10 2 540 0 540 54 0 594 0
11 2 594 0 594 54 0 648 0
12 2 648 0 648 54 0 702 0
13 2 702 0 702 54 0 756 0
14 2 756 0 756 54 0 810 0
15 2 810 0 810 54 0 864 0
16 2 864 0 864 54 0 918 0
17 2 918 0 918 54 0 972 0
18 2 972 0 972 54 0 1026 0
19 2 1026 0 1026 54 0 1080 0
20 2 1080 0 1080 54 0 1134 0
21 2 1134 0 1134 54 0 1188 0
22 2 1188 0 1188 54 0 1242 0
23 2 1242 0 1242 54 0 1296 0
24 2 1296 0 1296 54 0 1350 0
25 2 1350 0 1350 54 0 1404 0
26 2 1404 0 1404 54 0 1458 0
27 2 1458 0 1458 54 0 1512 0
28 2 1512 0 1512 54 0 1566 0
29 2 1566 0 1566 54 0 1620 0
30 2 1620 0 1620 54 0 1674 0
31 2 1674 0 1674 54 0 1728 0
32 2 1728 0 1728 54 0 1782 0
33 2 1782 0 1782 54 0 1836 0
34 2 1836 0 1836 54 0 1890 0
35 2 1890 0 1890 54 0 1944 0
36 2 1944 0 1944 54 0 1998 0
37 2 1998 0 1998 54 0 2052 0
38 2 2052 0 2052 54 0 2106 0
39 2 2106 0 2106 54 0 2160 0
40 2 2160 0 2160 54 0 2214 0
41 2 2214 0 2214 54 0 2268 0
42 2 2268 0 2268 54 0 2322 0
43 2 2322 0 2322 54 0 2376 0
44 2 2376 0 2376 54 0 2430 0
45 2 2430 0 2430 54 0 2484 0
46 2 2484 0 2484 54 0 2538 0
47 2 2538 0 2538 54 0 2592 0
48 2 2592 0 2592 54 0 2646 0
49 2 2646 0 2646 54 0 2700 0
50 2 2700 0 2700 54 0 2754 0
51 2 2754 0 2754 54 0 2808 0
52 2 2808 0 2808 54 0 2862 0
53 2 2862 0 2862 54 0 2916 0
54 2 2916 0 2916 54 0 2970 0
55 2 2970 0 2970 54 0 3024 0
56 2 3024 0 3024 54 0 3078 0
57 2 3078 0 3078 54 0 3132 0
58 2 3132 0 3132 54 0 3186 0
59 2 3186 0 3186 54 0 3240 0
60 2 3240 0 3240 54 0 3294 0
61 2 3294 0 3294 54 0 3348 0
62 2 3348 0 3348 54 0 3402 0
63 2 3402 0 3402 54 0 3456 0
64 2 3456 0 3456 54 0 3510 0
65 2 3510 0 3510 54 0 3564 0
66 2 3564 0 3564 54 0 3618 0
67 2 3618 0 3618 54 0 3672 0
68 2 3672 0 3672 54 0 3726 0
69 2 3726 0 3726 54 0 3780 0
70 2 3780 0 3780 54 0 3834 0
71 2 3834 0 3834 54 0 3888 0
72 2 3888 0 3888 54 0 3942 0
73 2 3942 0 3942 54 0 3996 0
74 2 3996 0 3996 54 0 4050 0
75 2 4050 0 4050 54 0 4104 0
76 2 4104 0 4104 54 0 4158 0
77 2 4158 0 4158 54 0 4212 0
78 2 4212 0 4212 54 0 4266 0
79 2 4266 0 4266 54 0 4320 0
80 2 4320 0 4320 54 0 4374 0
81 2 4374 0 4374 54 0 4428 0
82 2 4428 0 4428 54 0 4482 0
83 2 4482 0 4482 54 0 4536 0
84 2 4536 0 4536 54 0 4590 0
85 2 4590 0 4590 54 0 4644 0
86 2 4644 0 4644 54 0 4698 0
87 2 4698 0 4698 54 0 4752 0
88 2 4752 0 4752 54 0 4806 0
89 2 4806 0 4806 54 0 4860 0
90 2 4860 0 4860 54 0 4914 0
91 2 4914 0 4914 54 0 4968 0
92 2 4968 0 4968 54 0 5022 0
93 2 5022 0 5022 54 0 5076 0
94 2 5076 0 5076 54 0 5130 0
95 2 5130 0 5130 54 0 5184 0
96 2 5184 0 5184 54 0 5238 0
97 2 5238 0 5238 54 0 5292 0
98 2 5292 0 5292 54 0 5346 0
99 2 5346 0 5346 54 0 5400 0
100 2 5400 0 5400 54 0 5454 0
CB 40-1
Cmax = 321
MC1 =
1 1 0 0 0 47 0 47 2
2 1 0 47 47 47 0 94 2
3 1 0 94 94 47 0 141 2
4 1 0 141 141 47 0 188 2
5 1 0 188 188 47 0 235 2
MC2 =
1 2 47 0 47 44 47 91 3
2 2 94 0 94 44 3 138 3
3 2 141 0 141 44 3 185 3
4 2 188 0 188 44 3 232 3
5 2 235 0 235 44 3 279 3
MC3 =
1 3 91 0 91 42 91 133 0
2 3 138 0 138 42 5 180 0
3 3 185 0 185 42 5 227 0
4 3 232 0 232 42 5 274 0
5 3 279 0 279 42 5 321 0
CB 40-2
Cmax = 556
MC1 =
1 1 0 0 0 47 0 47 2
2 1 0 47 47 47 0 94 2
3 1 0 94 94 47 0 141 2
4 1 0 141 141 47 0 188 2
5 1 0 188 188 47 0 235 2
6 1 0 235 235 47 0 282 2
7 1 0 282 282 47 0 329 2
8 1 0 329 329 47 0 376 2
9 1 0 376 376 47 0 423 2
10 1 0 423 423 47 0 470 2
MC2 =
1 2 47 0 47 44 47 91 3
2 2 94 0 94 44 3 138 3
3 2 141 0 141 44 3 185 3
4 2 188 0 188 44 3 232 3
5 2 235 0 235 44 3 279 3
6 2 282 0 282 44 3 326 3
7 2 329 0 329 44 3 373 3
8 2 376 0 376 44 3 420 3
9 2 423 0 423 44 3 467 3
10 2 470 0 470 44 3 514 3
MC3 =
1 3 91 0 91 42 91 133 0
2 3 138 0 138 42 5 180 0
3 3 185 0 185 42 5 227 0
4 3 232 0 232 42 5 274 0
5 3 279 0 279 42 5 321 0
6 3 326 0 326 42 5 368 0
7 3 373 0 373 42 5 415 0
8 3 420 0 420 42 5 462 0
9 3 467 0 467 42 5 509 0
10 3 514 0 514 42 5 556 0
CB 40-3
Cmax = 1261
MC1 =
1 1 0 0 0 47 0 47 2
2 1 0 47 47 47 0 94 2
3 1 0 94 94 47 0 141 2
4 1 0 141 141 47 0 188 2
5 1 0 188 188 47 0 235 2
6 1 0 235 235 47 0 282 2
7 1 0 282 282 47 0 329 2
8 1 0 329 329 47 0 376 2
9 1 0 376 376 47 0 423 2
10 1 0 423 423 47 0 470 2
11 1 0 470 470 47 0 517 2
12 1 0 517 517 47 0 564 2
13 1 0 564 564 47 0 611 2
14 1 0 611 611 47 0 658 2
15 1 0 658 658 47 0 705 2
16 1 0 705 705 47 0 752 2
17 1 0 752 752 47 0 799 2
18 1 0 799 799 47 0 846 2
19 1 0 846 846 47 0 893 2
20 1 0 893 893 47 0 940 2
21 1 0 940 940 47 0 987 2
22 1 0 987 987 47 0 1034 2
23 1 0 1034 1034 47 0 1081 2
24 1 0 1081 1081 47 0 1128 2
25 1 0 1128 1128 47 0 1175 2
MC2 =
1 2 47 0 47 44 47 91 3
2 2 94 0 94 44 3 138 3
3 2 141 0 141 44 3 185 3
4 2 188 0 188 44 3 232 3
5 2 235 0 235 44 3 279 3
6 2 282 0 282 44 3 326 3
7 2 329 0 329 44 3 373 3
8 2 376 0 376 44 3 420 3
9 2 423 0 423 44 3 467 3
10 2 470 0 470 44 3 514 3
11 2 517 0 517 44 3 561 3

12	2	564	0	564	44	3	608	3
13	2	611	0	611	44	3	655	3
14	2	658	0	658	44	3	702	3
15	2	705	0	705	44	3	749	3
16	2	752	0	752	44	3	796	3
17	2	799	0	799	44	3	843	3
18	2	846	0	846	44	3	890	3
19	2	893	0	893	44	3	937	3
20	2	940	0	940	44	3	984	3
21	2	987	0	987	44	3	1031	3
22	2	1034	0	1034	44	3	1078	3
23	2	1081	0	1081	44	3	1125	3
24	2	1128	0	1128	44	3	1172	3
25	2	1175	0	1175	44	3	1219	3

MC3 =

1	3	91	0	91	42	91	133	0
2	3	138	0	138	42	5	180	0
3	3	185	0	185	42	5	227	0
4	3	232	0	232	42	5	274	0
5	3	279	0	279	42	5	321	0
6	3	326	0	326	42	5	368	0
7	3	373	0	373	42	5	415	0
8	3	420	0	420	42	5	462	0
9	3	467	0	467	42	5	509	0
10	3	514	0	514	42	5	556	0
11	3	561	0	561	42	5	603	0
12	3	608	0	608	42	5	650	0
13	3	655	0	655	42	5	697	0
14	3	702	0	702	42	5	744	0
15	3	749	0	749	42	5	791	0
16	3	796	0	796	42	5	838	0
17	3	843	0	843	42	5	885	0
18	3	890	0	890	42	5	932	0
19	3	937	0	937	42	5	979	0
20	3	984	0	984	42	5	1026	0
21	3	1031	0	1031	42	5	1073	0
22	3	1078	0	1078	42	5	1120	0
23	3	1125	0	1125	42	5	1167	0
24	3	1172	0	1172	42	5	1214	0
25	3	1219	0	1219	42	5	1261	0

CB 40-4

Cmax = 2436

MC1 =

1	1	0	0	0	47	0	47	2
2	1	0	47	47	47	0	94	2
3	1	0	94	94	47	0	141	2
4	1	0	141	141	47	0	188	2
5	1	0	188	188	47	0	235	2
6	1	0	235	235	47	0	282	2
7	1	0	282	282	47	0	329	2
8	1	0	329	329	47	0	376	2
9	1	0	376	376	47	0	423	2
10	1	0	423	423	47	0	470	2
11	1	0	470	470	47	0	517	2
12	1	0	517	517	47	0	564	2
13	1	0	564	564	47	0	611	2
14	1	0	611	611	47	0	658	2
15	1	0	658	658	47	0	705	2
16	1	0	705	705	47	0	752	2
17	1	0	752	752	47	0	799	2
18	1	0	799	799	47	0	846	2
19	1	0	846	846	47	0	893	2
20	1	0	893	893	47	0	940	2
21	1	0	940	940	47	0	987	2
22	1	0	987	987	47	0	1034	2
23	1	0	1034	1034	47	0	1081	2
24	1	0	1081	1081	47	0	1128	2
25	1	0	1128	1128	47	0	1175	2
26	1	0	1175	1175	47	0	1222	2
27	1	0	1222	1222	47	0	1269	2
28	1	0	1269	1269	47	0	1316	2
29	1	0	1316	1316	47	0	1363	2
30	1	0	1363	1363	47	0	1410	2
31	1	0	1410	1410	47	0	1457	2
32	1	0	1457	1457	47	0	1504	2
33	1	0	1504	1504	47	0	1551	2
34	1	0	1551	1551	47	0	1598	2
35	1	0	1598	1598	47	0	1645	2
36	1	0	1645	1645	47	0	1692	2
37	1	0	1692	1692	47	0	1739	2
38	1	0	1739	1739	47	0	1786	2
39	1	0	1786	1786	47	0	1833	2
40	1	0	1833	1833	47	0	1880	2
41	1	0	1880	1880	47	0	1927	2
42	1	0	1927	1927	47	0	1974	2
43	1	0	1974	1974	47	0	2021	2
44	1	0	2021	2021	47	0	2068	2
45	1	0	2068	2068	47	0	2115	2
46	1	0	2115	2115	47	0	2162	2
47	1	0	2162	2162	47	0	2209	2
48	1	0	2209	2209	47	0	2256	2
49	1	0	2256	2256	47	0	2303	2
50	1	0	2303	2303	47	0	2350	2

MC2 =

1	2	47	0	47	44	47	91	3
2	2	94	0	94	44	3	138	3
3	2	141	0	141	44	3	185	3
4	2	188	0	188	44	3	232	3
5	2	235	0	235	44	3	279	3
6	2	282	0	282	44	3	326	3
7	2	329	0	329	44	3	373	3
8	2	376	0	376	44	3	420	3
9	2	423	0	423	44	3	467	3
10	2	470	0	470	44	3	514	3
11	2	517	0	517	44	3	561	3
12	2	564	0	564	44	3	608	3
13	2	611	0	611	44	3	655	3
14	2	658	0	658	44	3	702	3
15	2	705	0	705	44	3	749	3
16	2	752	0	752	44	3	796	3
17	2	799	0	799	44	3	843	3
18	2	846	0	846	44	3	890	3
19	2	893	0	893	44	3	937	3
20	2	940	0	940	44	3	984	3
21	2	987	0	987	44	3	1031	3
22	2	1034	0	1034	44	3	1078	3
23	2	1081	0	1081	44	3	1125	3
24	2	1128	0	1128	44	3	1172	3
25	2	1175	0	1175	44	3	1219	3
26	2	1222	0	1222	44	3	1266	3
27	2	1269	0	1269	44	3	1313	3
28	2	1316	0	1316	44	3	1360	3
29	2	1363	0	1363	44	3	1407	3
30	2	1410	0	1410	44	3	1454	3
31	2	1457	0	1457	44	3	1501	3
32	2	1504	0	1504	44	3	1548	3
33	2	1551	0	1551	44	3	1595	3
34	2	1598	0	1598	44	3	1642	3
35	2	1645	0	1645	44	3	1689	3
36	2	1692	0	1692	44	3	1736	3
37	2	1739	0	1739	44	3	1783	3
38	2	1786	0	1786	44	3	1830	3
39	2	1833	0	1833	44	3	1877	3
40	2	1880	0	1880	44	3	1924	3
41	2	1927	0	1927	44	3	1971	3
42	2	1974	0	1974	44	3	2018	3
43	2	2021	0	2021	44	3	2065	3
44	2	2068	0	2068	44	3	2112	3
45	2	2115	0	2115	44	3	2159	3
46	2	2162	0	2162	44	3	2206	3
47	2	2209	0	2209	44	3	2253	3
48	2	2256	0	2256	44	3	2300	3
49	2	2303	0	2303	44	3	2347	3
50	2	2350	0	2350	44	3	2394	3

MC3 =

1	3	91	0	91	42	91	133	0
2	3	138	0	138	42	5	180	0
3	3	185	0	185	42	5	227	0
4	3	232	0	232	42	5	274	0
5	3	279	0	279	42	5	321	0
6	3	326	0	326	42	5	368	0
7	3	373	0	373	42	5	415	0
8	3	420	0	420	42	5	462	0
9	3	467	0	467	42	5	509	0
10	3	514	0	514	42	5	556	0
11	3	561	0	561	42	5	603	0
12	3	608	0	608	42	5	650	0
13	3	655	0	655	42	5	697	0
14	3	702	0	702	42	5	744	0
15	3	749	0	749	42	5	791	0
16	3	796	0	796	42	5	838	0
17	3	843	0	843	42	5	885	0
18	3	890	0	890	42	5	932	0
19	3	937	0	937	42	5	979	0
20	3	984	0	984	42	5	1026	0
21	3	1031	0	1031	42	5	1073	0
22	3	1078	0	1078	42	5	1120	0
23	3	1125	0	1125	42	5	1167	0
24	3	1172	0	1172	42	5	1214	0
25	3	1219	0	1219	42	5	1261	0
26	3	1266	0	1266	42	5	1308	0
27	3	1313	0	1313	42	5	1355	0
28	3	1360	0	1360	42	5	1402	0
29	3	1407	0	1407	42	5	1449	0
30	3	1454	0	1454	42	5	1496	0
31	3	1501	0	1501	42	5	1543	0
32	3	1548	0	1548	42	5	1590	0
33	3	1595	0	1595	42	5	1637	0
34	3	1642	0	1642	42	5	1684	0
35	3	1689	0	1689	42	5	1731	0
36	3	1736	0	1736	42	5	1778	0
37	3	1783	0	1783	42	5	1825	0
38	3	1830	0	1830	42	5	1872	0
39	3	1877	0	1877	42	5	1919	0
40	3	1924	0	1924	42	5	1966	0
41	3	1971	0	1971	42	5	2013	0
42	3	2018	0	2018	42	5	2060	0
43	3	2065	0	2065	42	5	2107	0
44	3	2112	0	2112	42	5	2154	0
45	3	2159	0	2159	42	5	2201	0
46	3	2206	0	2206	42	5	2248	0
47	3	2253	0	2253	42	5	2295	0
48	3	2300	0	2300	42	5	2342	0
49	3	2347	0	2347	42	5	2389	0
50	3	2394	0	2394	42	5	2436	0

CB 40-5

Cmax = 3611

MC1 =

1	1	0	0	0	47	0	47	2
2	1	0	47	47	47	0	94	2
3	1	0	94	94	47	0	141	2
4	1	0	141	141	47	0	188	2

5	1	0	188	188	47	0	235	2
6	1	0	235	235	47	0	282	2
7	1	0	282	282	47	0	329	2
8	1	0	329	329	47	0	376	2
9	1	0	376	376	47	0	423	2
10	1	0	423	423	47	0	470	2
11	1	0	470	470	47	0	517	2
12	1	0	517	517	47	0	564	2
13	1	0	564	564	47	0	611	2
14	1	0	611	611	47	0	658	2
15	1	0	658	658	47	0	705	2
16	1	0	705	705	47	0	752	2
17	1	0	752	752	47	0	799	2
18	1	0	799	799	47	0	846	2
19	1	0	846	846	47	0	893	2
20	1	0	893	893	47	0	940	2
21	1	0	940	940	47	0	987	2
22	1	0	987	987	47	0	1034	2
23	1	0	1034	1034	47	0	1081	2
24	1	0	1081	1081	47	0	1128	2
25	1	0	1128	1128	47	0	1175	2
26	1	0	1175	1175	47	0	1222	2
27	1	0	1222	1222	47	0	1269	2
28	1	0	1269	1269	47	0	1316	2
29	1	0	1316	1316	47	0	1363	2
30	1	0	1363	1363	47	0	1410	2
31	1	0	1410	1410	47	0	1457	2
32	1	0	1457	1457	47	0	1504	2
33	1	0	1504	1504	47	0	1551	2
34	1	0	1551	1551	47	0	1598	2
35	1	0	1598	1598	47	0	1645	2
36	1	0	1645	1645	47	0	1692	2
37	1	0	1692	1692	47	0	1739	2
38	1	0	1739	1739	47	0	1786	2
39	1	0	1786	1786	47	0	1833	2
40	1	0	1833	1833	47	0	1880	2
41	1	0	1880	1880	47	0	1927	2
42	1	0	1927	1927	47	0	1974	2
43	1	0	1974	1974	47	0	2021	2
44	1	0	2021	2021	47	0	2068	2
45	1	0	2068	2068	47	0	2115	2
46	1	0	2115	2115	47	0	2162	2
47	1	0	2162	2162	47	0	2209	2
48	1	0	2209	2209	47	0	2256	2
49	1	0	2256	2256	47	0	2303	2
50	1	0	2303	2303	47	0	2350	2
51	1	0	2350	2350	47	0	2397	2
52	1	0	2397	2397	47	0	2444	2
53	1	0	2444	2444	47	0	2491	2
54	1	0	2491	2491	47	0	2538	2
55	1	0	2538	2538	47	0	2585	2
56	1	0	2585	2585	47	0	2632	2
57	1	0	2632	2632	47	0	2679	2
58	1	0	2679	2679	47	0	2726	2
59	1	0	2726	2726	47	0	2773	2
60	1	0	2773	2773	47	0	2820	2
61	1	0	2820	2820	47	0	2867	2
62	1	0	2867	2867	47	0	2914	2
63	1	0	2914	2914	47	0	2961	2
64	1	0	2961	2961	47	0	3008	2
65	1	0	3008	3008	47	0	3055	2
66	1	0	3055	3055	47	0	3102	2
67	1	0	3102	3102	47	0	3149	2
68	1	0	3149	3149	47	0	3196	2
69	1	0	3196	3196	47	0	3243	2
70	1	0	3243	3243	47	0	3290	2
71	1	0	3290	3290	47	0	3337	2
72	1	0	3337	3337	47	0	3384	2
73	1	0	3384	3384	47	0	3431	2
74	1	0	3431	3431	47	0	3478	2
75	1	0	3478	3478	47	0	3525	2

MC2 =

1	2	47	0	47	44	47	91	3
2	2	94	0	94	44	3	138	3
3	2	141	0	141	44	3	185	3
4	2	188	0	188	44	3	232	3
5	2	235	0	235	44	3	279	3
6	2	282	0	282	44	3	326	3
7	2	329	0	329	44	3	373	3
8	2	376	0	376	44	3	420	3
9	2	423	0	423	44	3	467	3
10	2	470	0	470	44	3	514	3
11	2	517	0	517	44	3	561	3
12	2	564	0	564	44	3	608	3
13	2	611	0	611	44	3	655	3
14	2	658	0	658	44	3	702	3
15	2	705	0	705	44	3	749	3
16	2	752	0	752	44	3	796	3
17	2	799	0	799	44	3	843	3
18	2	846	0	846	44	3	890	3
19	2	893	0	893	44	3	937	3
20	2	940	0	940	44	3	984	3
21	2	987	0	987	44	3	1031	3
22	2	1034	0	1034	44	3	1078	3
23	2	1081	0	1081	44	3	1125	3
24	2	1128	0	1128	44	3	1172	3
25	2	1175	0	1175	44	3	1219	3
26	2	1222	0	1222	44	3	1266	3
27	2	1269	0	1269	44	3	1313	3
28	2	1316	0	1316	44	3	1360	3
29	2	1363	0	1363	44	3	1407	3
30	2	1410	0	1410	44	3	1454	3
31	2	1457	0	1457	44	3	1501	3
32	2	1504	0	1504	44	3	1548	3
33	2	1551	0	1551	44	3	1595	3
34	2	1598	0	1598	44	3	1642	3
35	2	1645	0	1645	44	3	1689	3
36	2	1692	0	1692	44	3	1736	3
37	2	1739	0	1739	44	3	1783	3
38	2	1786	0	1786	44	3	1830	3
39	2	1833	0	1833	44	3	1877	3
40	2	1880	0	1880	44	3	1924	3
41	2	1927	0	1927	44	3	1971	3
42	2	1974	0	1974	44	3	2018	3
43	2	2021	0	2021	44	3	2065	3
44	2	2068	0	2068	44	3	2112	3
45	2	2115	0	2115	44	3	2159	3
46	2	2162	0	2162	44	3	2206	3
47	2	2209	0	2209	44	3	2253	3
48	2	2256	0	2256	44	3	2300	3
49	2	2303	0	2303	44	3	2347	3
50	2	2350	0	2350	44	3	2394	3
51	2	2397	0	2397	44	3	2441	3
52	2	2444	0	2444	44	3	2488	3
53	2	2491	0	2491	44	3	2535	3
54	2	2538	0	2538	44	3	2582	3
55	2	2585	0	2585	44	3	2629	3
56	2	2632	0	2632	44	3	2676	3
57	2	2679	0	2679	44	3	2723	3
58	2	2726	0	2726	44	3	2770	3
59	2	2773	0	2773	44	3	2817	3
60	2	2820	0	2820	44	3	2864	3
61	2	2867	0	2867	44	3	2911	3
62	2	2914	0	2914	44	3	2958	3
63	2	2961	0	2961	44	3	3005	3
64	2	3008	0	3008	44	3	3052	3
65	2	3055	0	3055	44	3	3099	3
66	2	3102	0	3102	44	3	3146	3
67	2	3149	0	3149	44	3	3193	3
68	2	3196	0	3196	44	3	3240	3
69	2	3243	0	3243	44	3	3287	3
70	2	3290	0	3290	44	3	3334	3
71	2	3337	0	3337	44	3	3381	3
72	2	3384	0	3384	44	3	3428	3
73	2	3431	0	3431	44	3	3475	3
74	2	3478	0	3478	44	3	3522	3
75	2	3525	0	3525	44	3	3569	3

MC3 =

1	3	91	0	91	42	91	133	0
2	3	138	0	138	42	5	180	0
3	3	185	0	185	42	5	227	0
4	3	232	0	232	42	5	274	0
5	3	279	0	279	42	5	321	0
6	3	326	0	326	42	5	368	0
7	3	373	0	373	42	5	415	0
8	3	420	0	420	42	5	462	0
9	3	467	0	467	42	5	509	0
10	3	514	0	514	42	5	556	0
11	3	561	0	561	42	5	603	0
12	3	608	0	608	42	5	650	0
13	3	655	0	655	42	5	697	0
14	3	702	0	702	42	5	744	0
15	3	749	0	749	42	5	791	0
16	3	796	0	796	42	5	838	0
17	3	843	0	843	42	5	885	0
18	3	890	0	890	42	5	932	0
19	3	937	0	937	42	5	979	0
20	3	984	0	984	42	5	1026	0
21	3	1031	0	1031	42	5	1073	0
22	3	1078	0	1078	42	5	1120	0
23	3	1125	0	1125	42	5	1167	0
24	3	1172	0	1172	42	5	1214	0
25	3	1219	0	1219	42	5	1261	0
26	3	1266	0	1266	42	5	1308	0
27	3	1313	0	1313	42	5	1355	0
28	3	1360	0	1360	42	5	1402	0
29	3	1407	0	1407	42	5	1449	0
30	3	1454	0	1454	42	5	1496	0
31	3	1501	0	1501	42	5	1543	0
32	3	1548	0	1548	42	5	1590	0
33	3	1595	0	1595	42	5	1637	0
34	3	1642	0	1642	42	5	1684	0
35	3	1689	0	1689	42	5	1731	0
36	3	1736	0	1736	42	5	1778	0
37	3	1783	0	1783	42	5	1825	0
38	3	1830	0	1830	42	5	1872	0
39	3	1877	0	1877	42	5	1919	0
40	3	1924	0	1924	42	5	1966	0
41	3	1971	0	1971	42	5	2013	0
42	3	2018	0	2018	42	5	2060	0
43	3	2065	0	2065	42	5	2107	0
44	3	2112	0	2112	42	5	2154	0
45	3	2159	0	2159	42	5	2201	0
46	3	2206	0	2206	42	5	2248	0
47	3	2253	0	2253	42	5	2295	0
48	3	2300	0	2300	42	5	2342	0
49	3	2347	0	2347	42	5	2389	0
50	3	2394	0	2394	42	5	2436	0
51	3	2441	0	2441	42	5	2483	0
52	3	2488	0	2488	42	5	2530	0
53	3	2535	0	2535	42	5	2577	0
54	3	2582	0	2582	42	5	2624	0

55	3	2629	0	2629	42	5	2671	0
56	3	2676	0	2676	42	5	2718	0
57	3	2723	0	2723	42	5	2765	0
58	3	2770	0	2770	42	5	2812	0
59	3	2817	0	2817	42	5	2859	0
60	3	2864	0	2864	42	5	2906	0
61	3	2911	0	2911	42	5	2953	0
62	3	2958	0	2958	42	5	3000	0
63	3	3005	0	3005	42	5	3047	0
64	3	3052	0	3052	42	5	3094	0
65	3	3099	0	3099	42	5	3141	0
66	3	3146	0	3146	42	5	3188	0
67	3	3193	0	3193	42	5	3235	0
68	3	3240	0	3240	42	5	3282	0
69	3	3287	0	3287	42	5	3329	0
70	3	3334	0	3334	42	5	3376	0
71	3	3381	0	3381	42	5	3423	0
72	3	3428	0	3428	42	5	3470	0
73	3	3475	0	3475	42	5	3517	0
74	3	3522	0	3522	42	5	3564	0
75	3	3569	0	3569	42	5	3611	0

CB 40-6

Cmax = 4786

MC1 =

1	1	0	0	0	47	0	47	2
2	1	0	47	47	47	0	94	2
3	1	0	94	94	47	0	141	2
4	1	0	141	141	47	0	188	2
5	1	0	188	188	47	0	235	2
6	1	0	235	235	47	0	282	2
7	1	0	282	282	47	0	329	2
8	1	0	329	329	47	0	376	2
9	1	0	376	376	47	0	423	2
10	1	0	423	423	47	0	470	2
11	1	0	470	470	47	0	517	2
12	1	0	517	517	47	0	564	2
13	1	0	564	564	47	0	611	2
14	1	0	611	611	47	0	658	2
15	1	0	658	658	47	0	705	2
16	1	0	705	705	47	0	752	2
17	1	0	752	752	47	0	799	2
18	1	0	799	799	47	0	846	2
19	1	0	846	846	47	0	893	2
20	1	0	893	893	47	0	940	2
21	1	0	940	940	47	0	987	2
22	1	0	987	987	47	0	1034	2
23	1	0	1034	1034	47	0	1081	2
24	1	0	1081	1081	47	0	1128	2
25	1	0	1128	1128	47	0	1175	2
26	1	0	1175	1175	47	0	1222	2
27	1	0	1222	1222	47	0	1269	2
28	1	0	1269	1269	47	0	1316	2
29	1	0	1316	1316	47	0	1363	2
30	1	0	1363	1363	47	0	1410	2
31	1	0	1410	1410	47	0	1457	2
32	1	0	1457	1457	47	0	1504	2
33	1	0	1504	1504	47	0	1551	2
34	1	0	1551	1551	47	0	1598	2
35	1	0	1598	1598	47	0	1645	2
36	1	0	1645	1645	47	0	1692	2
37	1	0	1692	1692	47	0	1739	2
38	1	0	1739	1739	47	0	1786	2
39	1	0	1786	1786	47	0	1833	2
40	1	0	1833	1833	47	0	1880	2
41	1	0	1880	1880	47	0	1927	2
42	1	0	1927	1927	47	0	1974	2
43	1	0	1974	1974	47	0	2021	2
44	1	0	2021	2021	47	0	2068	2
45	1	0	2068	2068	47	0	2115	2
46	1	0	2115	2115	47	0	2162	2
47	1	0	2162	2162	47	0	2209	2
48	1	0	2209	2209	47	0	2256	2
49	1	0	2256	2256	47	0	2303	2
50	1	0	2303	2303	47	0	2350	2
51	1	0	2350	2350	47	0	2397	2
52	1	0	2397	2397	47	0	2444	2
53	1	0	2444	2444	47	0	2491	2
54	1	0	2491	2491	47	0	2538	2
55	1	0	2538	2538	47	0	2585	2
56	1	0	2585	2585	47	0	2632	2
57	1	0	2632	2632	47	0	2679	2
58	1	0	2679	2679	47	0	2726	2
59	1	0	2726	2726	47	0	2773	2
60	1	0	2773	2773	47	0	2820	2
61	1	0	2820	2820	47	0	2867	2
62	1	0	2867	2867	47	0	2914	2
63	1	0	2914	2914	47	0	2961	2
64	1	0	2961	2961	47	0	3008	2
65	1	0	3008	3008	47	0	3055	2
66	1	0	3055	3055	47	0	3102	2
67	1	0	3102	3102	47	0	3149	2
68	1	0	3149	3149	47	0	3196	2
69	1	0	3196	3196	47	0	3243	2
70	1	0	3243	3243	47	0	3290	2
71	1	0	3290	3290	47	0	3337	2
72	1	0	3337	3337	47	0	3384	2
73	1	0	3384	3384	47	0	3431	2
74	1	0	3431	3431	47	0	3478	2
75	1	0	3478	3478	47	0	3525	2
76	1	0	3525	3525	47	0	3572	2
77	1	0	3572	3572	47	0	3619	2
78	1	0	3619	3619	47	0	3666	2
79	1	0	3666	3666	47	0	3713	2
80	1	0	3713	3713	47	0	3760	2
81	1	0	3760	3760	47	0	3807	2
82	1	0	3807	3807	47	0	3854	2
83	1	0	3854	3854	47	0	3901	2
84	1	0	3901	3901	47	0	3948	2
85	1	0	3948	3948	47	0	3995	2
86	1	0	3995	3995	47	0	4042	2
87	1	0	4042	4042	47	0	4089	2
88	1	0	4089	4089	47	0	4136	2
89	1	0	4136	4136	47	0	4183	2
90	1	0	4183	4183	47	0	4230	2
91	1	0	4230	4230	47	0	4277	2
92	1	0	4277	4277	47	0	4324	2
93	1	0	4324	4324	47	0	4371	2
94	1	0	4371	4371	47	0	4418	2
95	1	0	4418	4418	47	0	4465	2
96	1	0	4465	4465	47	0	4512	2
97	1	0	4512	4512	47	0	4559	2
98	1	0	4559	4559	47	0	4606	2
99	1	0	4606	4606	47	0	4653	2
100	1	0	4653	4653	47	0	4700	2

MC2 =

1	2	47	0	47	44	47	91	3
2	2	94	0	94	44	3	138	3
3	2	141	0	141	44	3	185	3
4	2	188	0	188	44	3	232	3
5	2	235	0	235	44	3	279	3
6	2	282	0	282	44	3	326	3
7	2	329	0	329	44	3	373	3
8	2	376	0	376	44	3	420	3
9	2	423	0	423	44	3	467	3
10	2	470	0	470	44	3	514	3
11	2	517	0	517	44	3	561	3
12	2	564	0	564	44	3	608	3
13	2	611	0	611	44	3	655	3
14	2	658	0	658	44	3	702	3
15	2	705	0	705	44	3	749	3
16	2	752	0	752	44	3	796	3
17	2	799	0	799	44	3	843	3
18	2	846	0	846	44	3	890	3
19	2	893	0	893	44	3	937	3
20	2	940	0	940	44	3	984	3
21	2	987	0	987	44	3	1031	3
22	2	1034	0	1034	44	3	1078	3
23	2	1081	0	1081	44	3	1125	3
24	2	1128	0	1128	44	3	1172	3
25	2	1175	0	1175	44	3	1219	3
26	2	1222	0	1222	44	3	1266	3
27	2	1269	0	1269	44	3	1313	3
28	2	1316	0	1316	44	3	1360	3
29	2	1363	0	1363	44	3	1407	3
30	2	1410	0	1410	44	3	1454	3
31	2	1457	0	1457	44	3	1501	3
32	2	1504	0	1504	44	3	1548	3
33	2	1551	0	1551	44	3	1595	3
34	2	1598	0	1598	44	3	1642	3
35	2	1645	0	1645	44	3	1689	3
36	2	1692	0	1692	44	3	1736	3
37	2	1739	0	1739	44	3	1783	3
38	2	1786	0	1786	44	3	1830	3
39	2	1833	0	1833	44	3	1877	3
40	2	1880	0	1880	44	3	1924	3
41	2	1927	0	1927	44	3	1971	3
42	2	1974	0	1974	44	3	2018	3
43	2	2021	0	2021	44	3	2065	3
44	2	2068	0	2068	44	3	2112	3
45	2	2115	0	2115	44	3	2159	3
46	2	2162	0	2162	44	3	2206	3
47	2	2209	0	2209	44	3	2253	3
48	2	2256	0	2256	44	3	2300	3
49	2	2303	0	2303	44	3	2347	3
50	2	2350	0	2350	44	3	2394	3
51	2	2397	0	2397	44	3	2441	3
52	2	2444	0	2444	44	3	2488	3
53	2	2491	0	2491	44	3	2535	3
54	2	2538	0	2538	44	3	2582	3
55	2	2585	0	2585	44	3	2629	3
56	2	2632	0	2632	44	3	2676	3
57	2	2679	0	2679	44	3	2723	3
58	2	2726	0	2726	44	3	2770	3
59	2	2773	0	2773	44	3	2817	3
60	2	2820	0	2820	44	3	2864	3
61	2	2867	0	2867	44	3	2911	3
62	2	2914	0	2914	44	3	2958	3
63	2	2961	0	2961	44	3	3005	3
64	2	3008	0	3008	44	3	3052	3
65	2	3055	0	3055	44	3	3099	3
66	2	3102	0	3102	44	3	3146	3
67	2	3149	0	3149	44	3	3193	3
68	2	3196	0	3196	44	3	3240	3
69	2	3243	0	3243	44	3	3287	3
70	2	3290	0	3290	44	3	3334	3
71	2	3337	0	3337	44	3	3381	3
72	2	3384	0	3384	44	3	3428	3
73	2	3431	0	3431	44	3	3475	3
74	2	3478	0	3478	44	3	3522	3
75	2	3525	0	3525	44	3	3569	3
76	2	3572	0	3572	44	3	3616	3
77	2	3619	0	3619	44	3	3663	3

78	2	3666	0	3666	44	3	3710	3
79	2	3713	0	3713	44	3	3757	3
80	2	3760	0	3760	44	3	3804	3
81	2	3807	0	3807	44	3	3851	3
82	2	3854	0	3854	44	3	3898	3
83	2	3901	0	3901	44	3	3945	3
84	2	3948	0	3948	44	3	3992	3
85	2	3995	0	3995	44	3	4039	3
86	2	4042	0	4042	44	3	4086	3
87	2	4089	0	4089	44	3	4133	3
88	2	4136	0	4136	44	3	4180	3
89	2	4183	0	4183	44	3	4227	3
90	2	4230	0	4230	44	3	4274	3
91	2	4277	0	4277	44	3	4321	3
92	2	4324	0	4324	44	3	4368	3
93	2	4371	0	4371	44	3	4415	3
94	2	4418	0	4418	44	3	4462	3
95	2	4465	0	4465	44	3	4509	3
96	2	4512	0	4512	44	3	4556	3
97	2	4559	0	4559	44	3	4603	3
98	2	4606	0	4606	44	3	4650	3
99	2	4653	0	4653	44	3	4697	3
100	2	4700	0	4700	44	3	4744	3

MC3 =

1	3	91	0	91	42	91	133	0
2	3	138	0	138	42	5	180	0
3	3	185	0	185	42	5	227	0
4	3	232	0	232	42	5	274	0
5	3	279	0	279	42	5	321	0
6	3	326	0	326	42	5	368	0
7	3	373	0	373	42	5	415	0
8	3	420	0	420	42	5	462	0
9	3	467	0	467	42	5	509	0
10	3	514	0	514	42	5	556	0
11	3	561	0	561	42	5	603	0
12	3	608	0	608	42	5	650	0
13	3	655	0	655	42	5	697	0
14	3	702	0	702	42	5	744	0
15	3	749	0	749	42	5	791	0
16	3	796	0	796	42	5	838	0
17	3	843	0	843	42	5	885	0
18	3	890	0	890	42	5	932	0
19	3	937	0	937	42	5	979	0
20	3	984	0	984	42	5	1026	0
21	3	1031	0	1031	42	5	1073	0
22	3	1078	0	1078	42	5	1120	0
23	3	1125	0	1125	42	5	1167	0
24	3	1172	0	1172	42	5	1214	0
25	3	1219	0	1219	42	5	1261	0
26	3	1266	0	1266	42	5	1308	0
27	3	1313	0	1313	42	5	1355	0
28	3	1360	0	1360	42	5	1402	0
29	3	1407	0	1407	42	5	1449	0
30	3	1454	0	1454	42	5	1496	0
31	3	1501	0	1501	42	5	1543	0
32	3	1548	0	1548	42	5	1590	0
33	3	1595	0	1595	42	5	1637	0
34	3	1642	0	1642	42	5	1684	0
35	3	1689	0	1689	42	5	1731	0
36	3	1736	0	1736	42	5	1778	0
37	3	1783	0	1783	42	5	1825	0
38	3	1830	0	1830	42	5	1872	0
39	3	1877	0	1877	42	5	1919	0
40	3	1924	0	1924	42	5	1966	0
41	3	1971	0	1971	42	5	2013	0
42	3	2018	0	2018	42	5	2060	0
43	3	2065	0	2065	42	5	2107	0
44	3	2112	0	2112	42	5	2154	0
45	3	2159	0	2159	42	5	2201	0
46	3	2206	0	2206	42	5	2248	0
47	3	2253	0	2253	42	5	2295	0
48	3	2300	0	2300	42	5	2342	0
49	3	2347	0	2347	42	5	2389	0
50	3	2394	0	2394	42	5	2436	0
51	3	2441	0	2441	42	5	2483	0
52	3	2488	0	2488	42	5	2530	0
53	3	2535	0	2535	42	5	2577	0
54	3	2582	0	2582	42	5	2624	0
55	3	2629	0	2629	42	5	2671	0
56	3	2676	0	2676	42	5	2718	0
57	3	2723	0	2723	42	5	2765	0
58	3	2770	0	2770	42	5	2812	0
59	3	2817	0	2817	42	5	2859	0
60	3	2864	0	2864	42	5	2906	0
61	3	2911	0	2911	42	5	2953	0
62	3	2958	0	2958	42	5	3000	0
63	3	3005	0	3005	42	5	3047	0
64	3	3052	0	3052	42	5	3094	0
65	3	3099	0	3099	42	5	3141	0
66	3	3146	0	3146	42	5	3188	0
67	3	3193	0	3193	42	5	3235	0
68	3	3240	0	3240	42	5	3282	0
69	3	3287	0	3287	42	5	3329	0
70	3	3334	0	3334	42	5	3376	0
71	3	3381	0	3381	42	5	3423	0
72	3	3428	0	3428	42	5	3470	0
73	3	3475	0	3475	42	5	3517	0
74	3	3522	0	3522	42	5	3564	0
75	3	3569	0	3569	42	5	3611	0
76	3	3616	0	3616	42	5	3658	0
77	3	3663	0	3663	42	5	3705	0

78	3	3710	0	3710	42	5	3752	0
79	3	3757	0	3757	42	5	3799	0
80	3	3804	0	3804	42	5	3846	0
81	3	3851	0	3851	42	5	3893	0
82	3	3898	0	3898	42	5	3940	0
83	3	3945	0	3945	42	5	3987	0
84	3	3992	0	3992	42	5	4034	0
85	3	4039	0	4039	42	5	4081	0
86	3	4086	0	4086	42	5	4128	0
87	3	4133	0	4133	42	5	4175	0
88	3	4180	0	4180	42	5	4222	0
89	3	4227	0	4227	42	5	4269	0
90	3	4274	0	4274	42	5	4316	0
91	3	4321	0	4321	42	5	4363	0
92	3	4368	0	4368	42	5	4410	0
93	3	4415	0	4415	42	5	4457	0
94	3	4462	0	4462	42	5	4504	0
95	3	4509	0	4509	42	5	4551	0
96	3	4556	0	4556	42	5	4598	0
97	3	4603	0	4603	42	5	4645	0
98	3	4650	0	4650	42	5	4692	0
99	3	4697	0	4697	42	5	4739	0
100	3	4744	0	4744	42	5	4786	0

APPENDIX E: CASE STUDY 3 - MFPP PROBLEMS

This appendix contains a results statistics for case studies.

Column 1: Shows current job number which seizes the machines Job Number

Column 2: Shows operation number of jobs in process in column 1

Column 3: Shows Arrival Time of job in column 1

Column 4: Shows Waiting Time for a job to be loaded on the machine

Column 5: Shows start time of a process

Column 6: Shows Processing Time of job on machines

Column 7: Shows Machine Idle Time

Column 8: Shows Finish time

Column 9: Shows Next Machine on which finished job to be processed

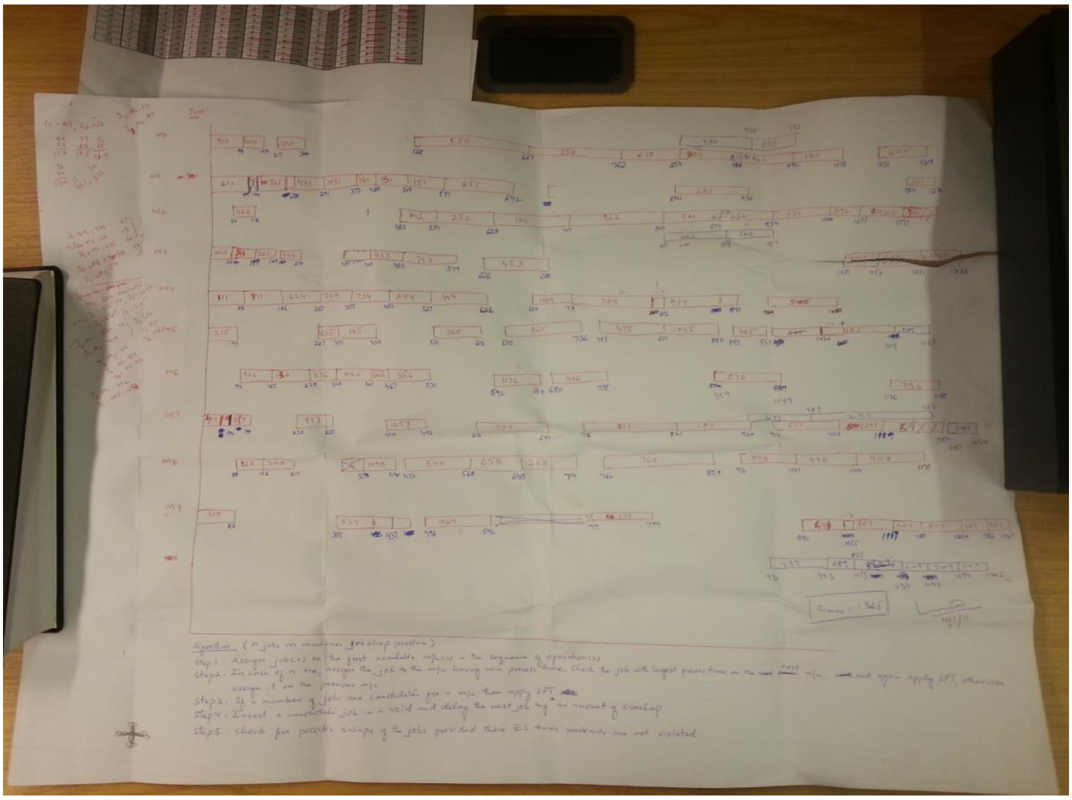
(Note: If the table splits, continue to next page).

```
Cmax = 300.2000
MC1 =
  5.0000  1.0000     0     0           0     0     0     0  2.0000
  6.0000  1.0000     0     0            0     0     0     0  2.0000
  7.0000  1.0000     0     0           0     0     0     0  2.0000
  8.0000  1.0000     0     0           0     0     0     0  2.0000
  1.0000  1.0000     0     0           0  66.400  0     0  66.4000  2.0000
  2.0000  1.0000     0  66.400     0  66.400  0  66.4000     0 132.8000  2.0000
  3.0000  1.0000     0 132.800     0 132.800 0  66.4000     0 199.2000  2.0000
  4.0000  1.0000     0 199.200     0 199.2000  66.4000     0 265.6000  2.0000
MC2 =
  5.0000  2.0000     0     0     0  20.2000     0  20.2000  3.0000
  6.0000  2.0000     0  20.2000  20.2000  20.2000     0  40.4000  3.0000
  7.0000  2.0000     0  40.4000  40.4000  20.2000     0  60.6000  3.0000
  8.0000  2.0000     0  60.6000  60.6000  20.2000     0  80.8000  3.0000
  1.0000  2.0000  66.4000  14.4000  80.8000     0     0  80.8000  3.0000
  2.0000  2.0000 132.8000     0 132.8000     0  52.0000 132.8000  3.0000
  3.0000  2.0000 199.2000     0 199.2000     0  66.4000 199.2000  3.0000
  4.0000  2.0000 265.6000     0 265.6000     0  66.4000 265.6000  3.0000
MC3 =
  5.0000  3.0000  20.2000     0  20.2000     0  20.2000  20.2000  4.0000
  6.0000  3.0000  40.4000     0  40.4000     0  20.2000  40.4000  4.0000
  7.0000  3.0000  60.6000     0  60.6000     0  20.2000  60.6000  4.0000
  8.0000  3.0000  80.8000     0  80.8000     0  20.2000  80.8000  4.0000
  1.0000  3.0000  80.8000     0  80.8000  31.0000     0 111.8000  4.0000
  2.0000  3.0000 132.8000     0 132.8000  31.0000  21.0000 163.8000  4.0000
  3.0000  3.0000 199.2000     0 199.2000  31.0000  35.4000 230.2000  4.0000
  4.0000  3.0000 265.6000     0 265.6000  31.0000  35.4000 296.6000  4.0000
MC4 =
  5.0000  4.0000  20.2000     0  20.2000  27.8000  20.2000  48.0000  5.0000
  6.0000  4.0000  40.4000  7.6000  48.0000  27.8000     0  75.8000  5.0000
  7.0000  4.0000  60.6000  15.2000  75.8000  27.8000     0 103.6000  5.0000
  8.0000  4.0000  80.8000  22.8000 103.6000  27.8000     0 131.4000  5.0000
  1.0000  4.0000 111.8000  19.6000 131.4000     0     0 131.4000  5.0000
  2.0000  4.0000 163.8000     0 163.8000     0  32.4000 163.8000  5.0000
  3.0000  4.0000 230.2000     0 230.2000     0  66.4000 230.2000  5.0000
  4.0000  4.0000 296.6000     0 296.6000     0  66.4000 296.6000  5.0000
MC5 =
  5.0000  5.0000  48.0000     0  48.0000  27.8000  48.0000  75.8000  6.0000
  6.0000  5.0000  75.8000     0  75.8000  27.8000     0 103.6000  6.0000
  7.0000  5.0000 103.6000     0 103.6000  27.8000     0 131.4000  6.0000
  8.0000  5.0000 131.4000     0 131.4000  27.8000     0 159.2000  6.0000
  1.0000  5.0000 131.4000  27.8000 159.2000     0     0 159.2000  6.0000
  2.0000  5.0000 163.8000     0 163.8000     0  4.6000 163.8000  6.0000
  3.0000  5.0000 230.2000     0 230.2000     0  66.4000 230.2000  6.0000
  4.0000  5.0000 296.6000     0 296.6000     0  66.4000 296.6000  6.0000
MC6 =
  5.0000  6.0000  75.8000     0  75.8000  3.6000  75.8000  79.4000     0
  6.0000  6.0000 103.6000     0 103.6000  3.6000  24.2000 107.2000     0
  7.0000  6.0000 131.4000     0 131.4000  3.6000  24.2000 135.0000     0
  8.0000  6.0000 159.2000     0 159.2000  3.6000  24.2000 162.8000     0
  1.0000  6.0000 159.2000  3.6000 162.8000  3.6000     0 166.4000     0
  2.0000  6.0000 163.8000  2.6000 166.4000  3.6000     0 170.0000     0
  3.0000  6.0000 230.2000     0 230.2000  3.6000  60.2000 233.8000     0
  4.0000  6.0000 296.6000     0 296.6000  3.6000  62.8000 300.2000     0
```

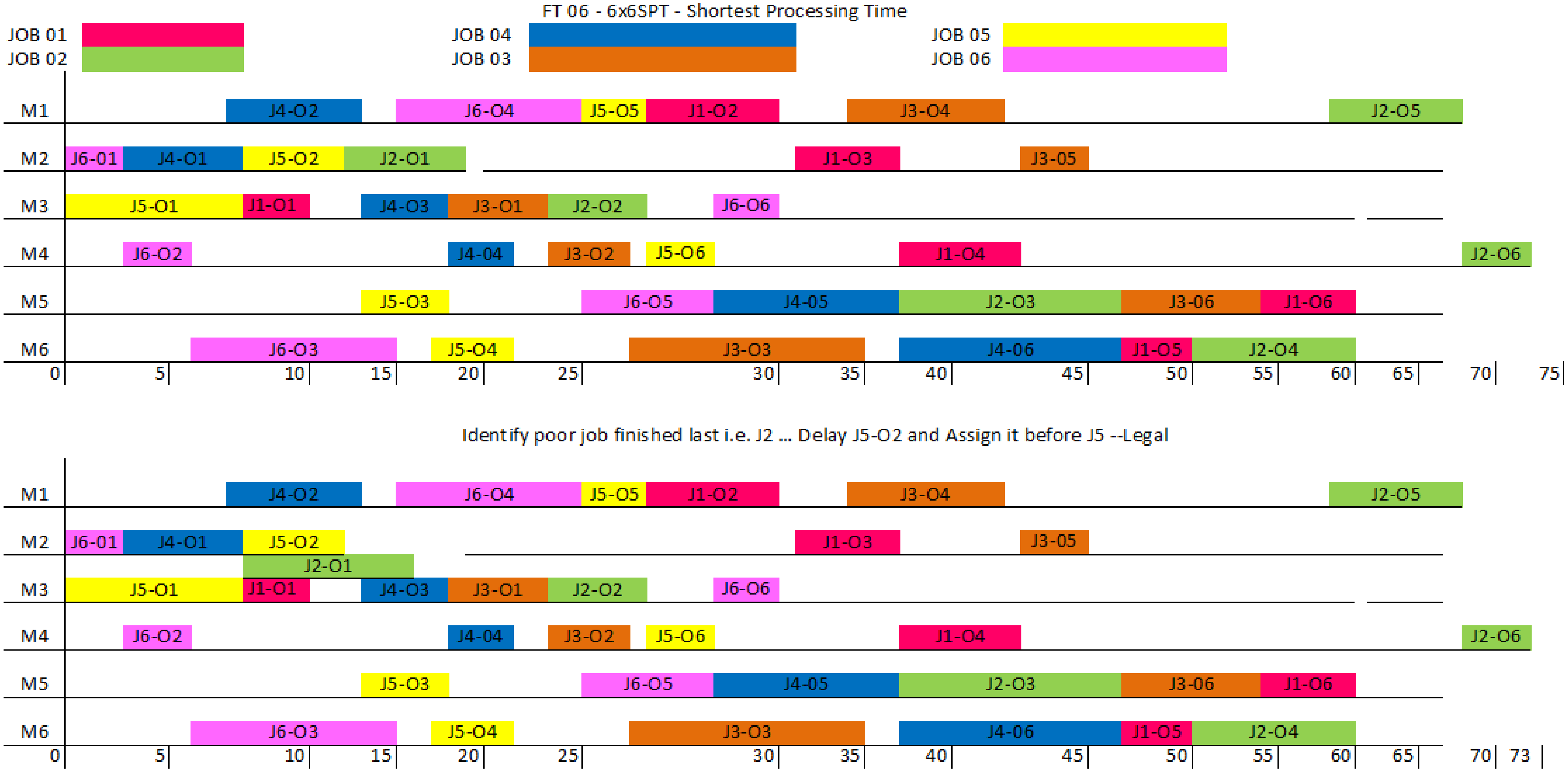
APPENDIX F: ATTEMPTS

This appendix shows the Makespan or the result (Gantt charts) for FT06 and LA02 benchmark job shop scheduling problem with some of the new procedures.

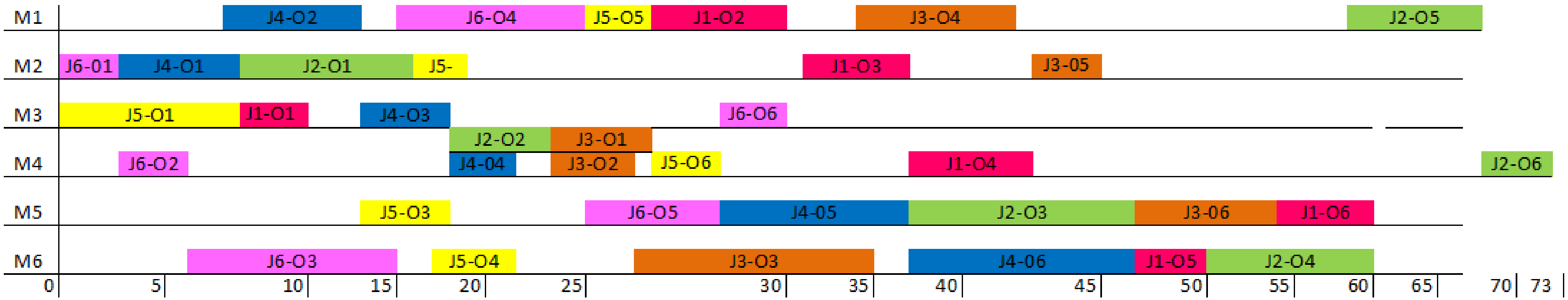
The number of attempts listed in Chapter 4 were applied to different size of the problems and were solved manually using drawing sheets. An example is shown in the figure below (snapshot of FT10: 10x10 problem). Practically, it was a laborious job, even a simple job solution procedure took hours, but has helped in understanding the behavior of the problems. The reproduction of these attempts in excel need months. Therefore, the application of new procedure attempts on FT06 and LA02, which are larger problems is provided in this appendix. These attempts and the attempts in Chapter 4 quit fairly explain the process that how these procedures resulted in Index Based Heuristic (IBH).



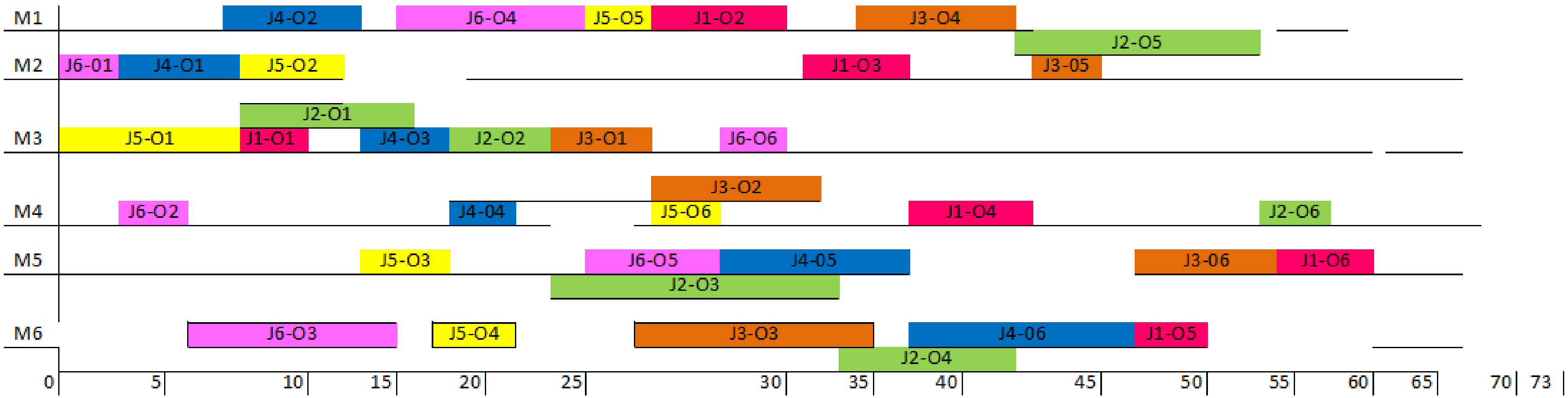
Attempt on FT06 (6x6) – Exchange procedure



Assign J2-O2 before J3-O1 on M3 - Results in Overlapping and Apply Delay on J3-O2 - Legal with increase in Complexity

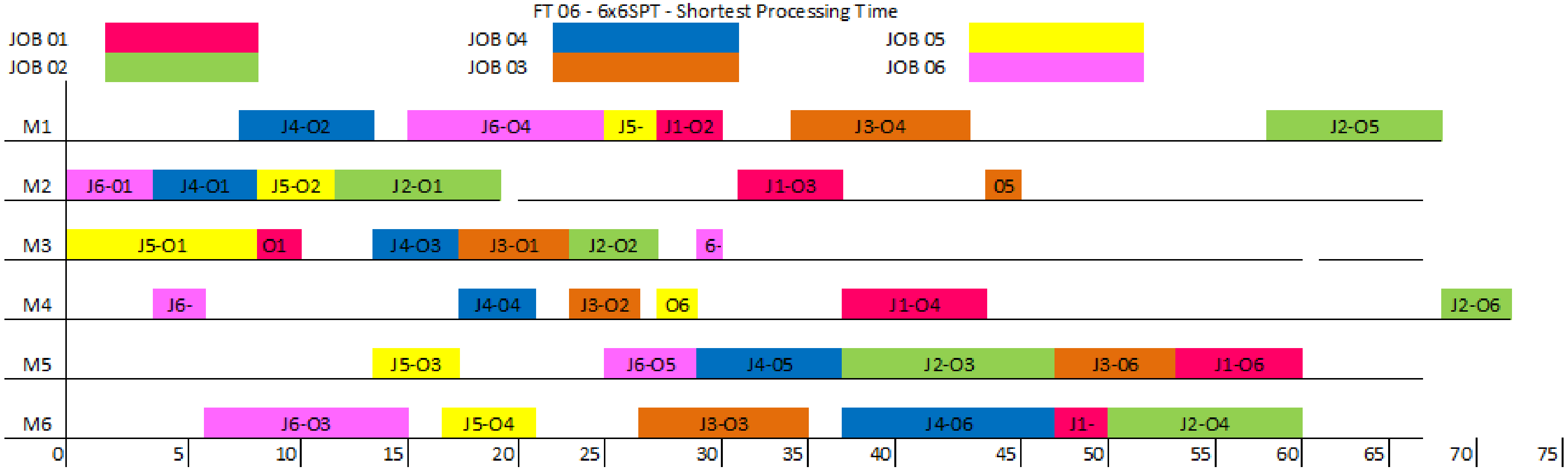


Delay J3-O2 on M4 will overlap J5-06 delay J5-06 & Assign J2-O3 at the end of J2-O2 and Delay overlapped J06 -O5 . This lead into so many overlapping and violation of precedence constraints that It made it impossible to logically program it.

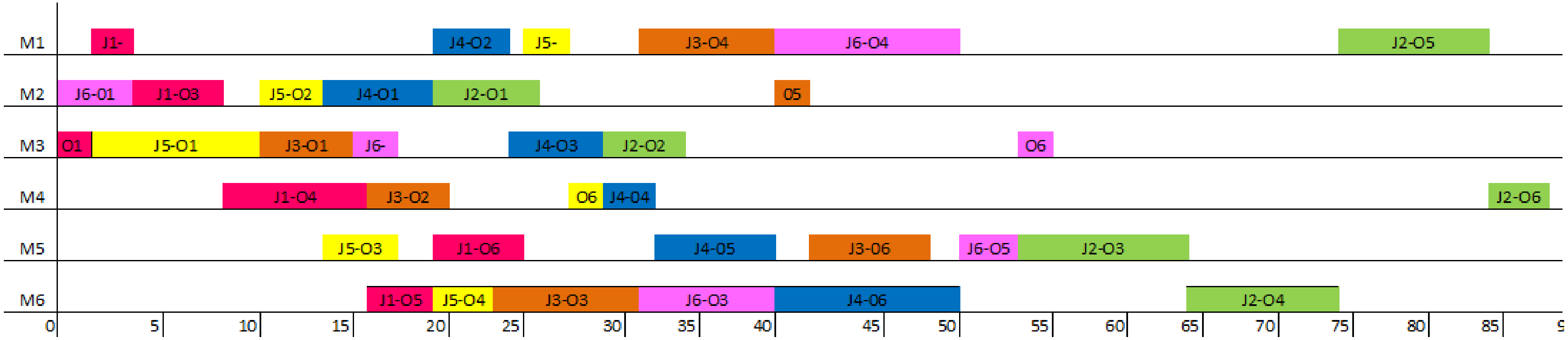


Conclusion: The priority of on job and delay lead in complexities and made it difficult to program. The Delay was very effective in small problem, however, it increased the complexities. Hence, it is impossible to program such procedure although manually it is possible and will lead a feasible schedule

Attempt on FT06 – Changing priority



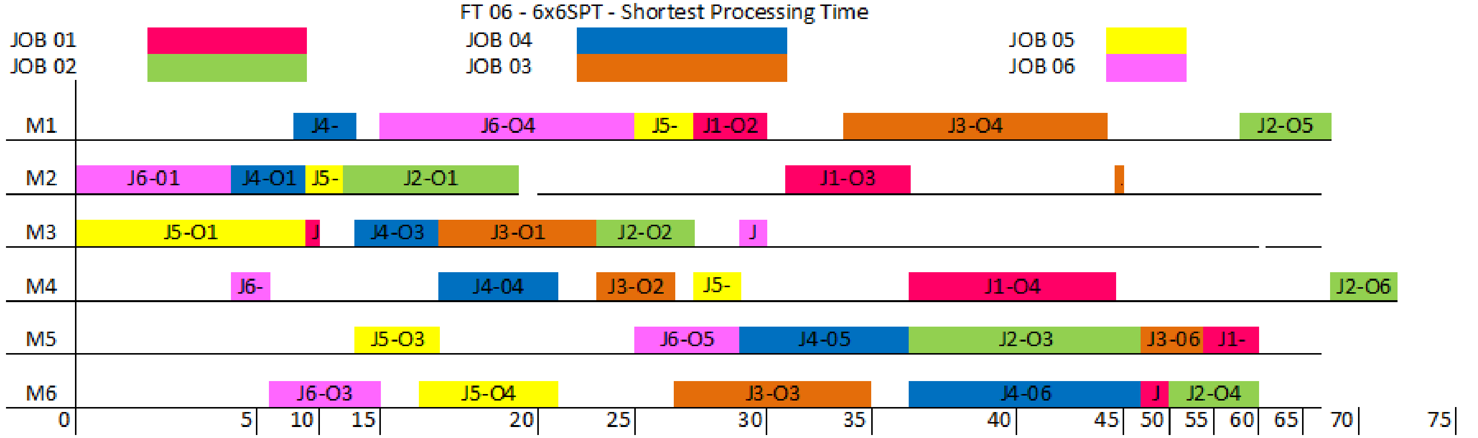
Using Attempt 04 technique: Results in the following schedule



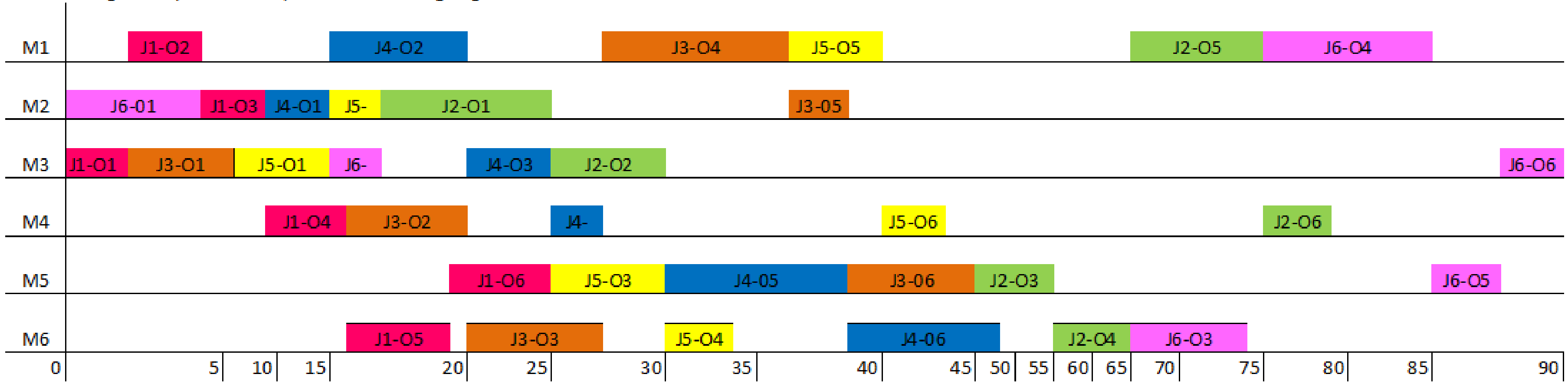
Conclusion: Although it yielded a feasible schedule, but the result is worst than the actual.

Attempt on FT06: Priority Exchange of 50% jobs

Assign half job ties on the same machine on SPT and rest of jobs ties on the same machine on LPT

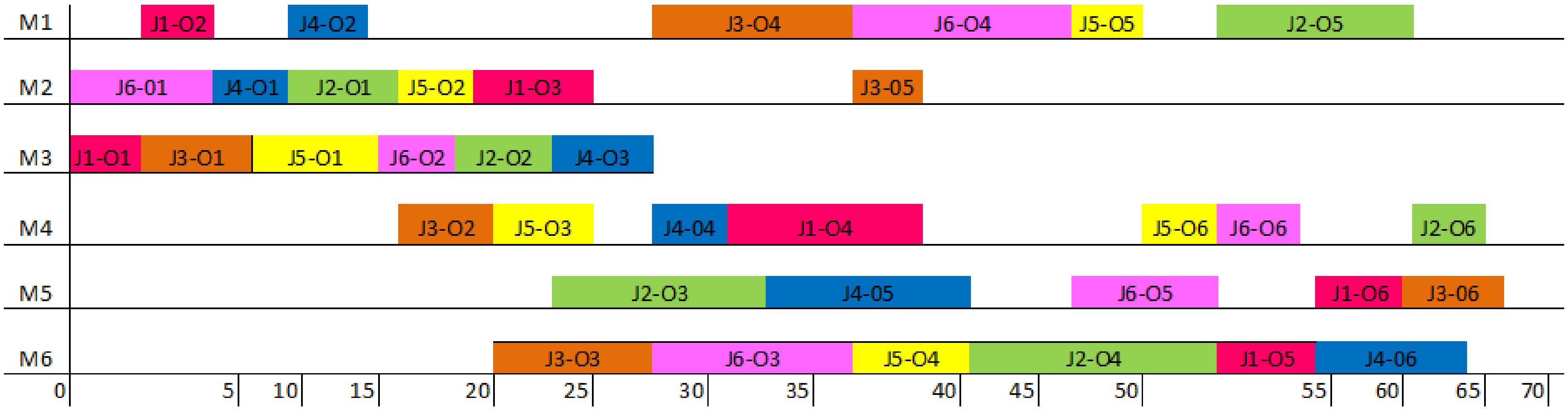


Using Attempt 04 technique modified: Assigning Order J1-J3-J1-J2-J4-J6



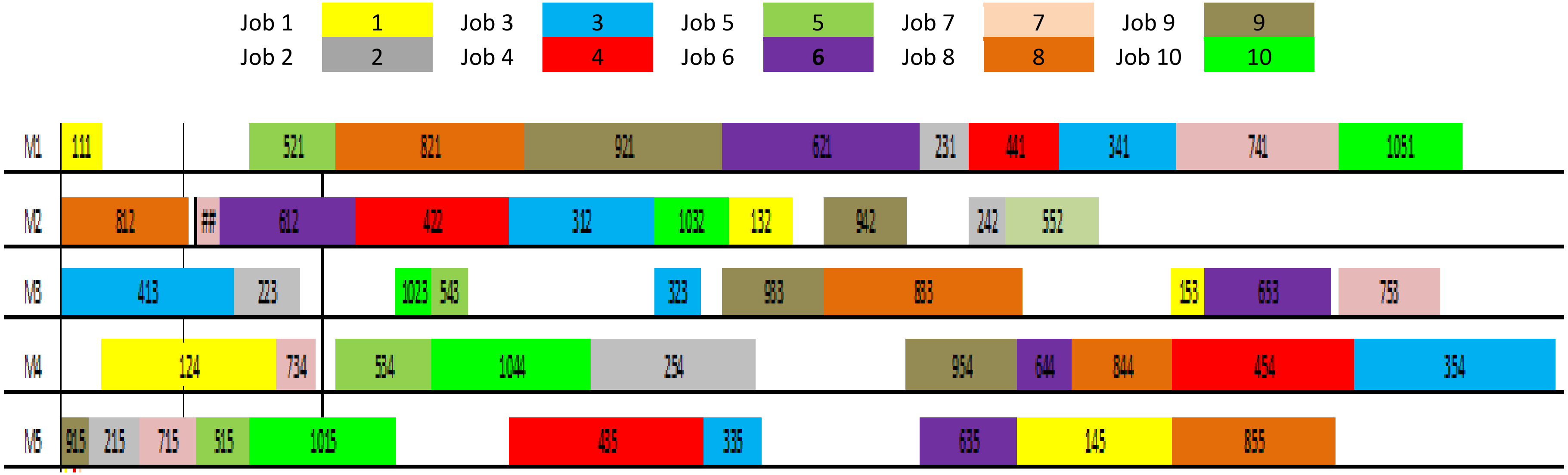
Conclusion: again it yielded a feasible schedule, but the result is worst than the actual.

Attempt on FT06 swapping jobs or prioritizing jobs in ascending order for in each operation shown below:
Operation 1: J1,J6,J3,J4,J2,J5
Operation 2: J1, J5, J6, J3, J2, J4
Operation 3: J4, J5, J1, J3, J6, J2
Operation 4: J4, J5, J1, J3, J2, J6
Operation 5: J3, J1, J5, J6, J4, J2
Operation 6: J5,J6, J2, J1, J3, J4



Conclusion: This swapping attempt improved the results and yielded a feasible schedule. This attempt was tried on another problem (LA02 – 10x5) in order to check whether it will be effective on a larger number of jobs or not.

Attempt on Lawrence (1984) – LA02 : 10 x 5 JSSP with a Makespan value of 655.
SPT rule gives a Makespan of 1022.
Applying the Swapping Technique which prioritizing jobs in ascending order for in each operation. For this problem the final schedule is shown below:



Conclusion: The swapping technique yielded better solution. At this point an idea of taking effect of other job on the seized job on a machine was considered. Which resulted in normalization of the processing time and finally resulted in Index Based Heuristics (IBH).

APPENDIX G: LIST OF PUBLICATION FROM THIS RESEARCH

Journal Publication:

1. MAQSOOD, I. Hussain, S., KHAN, MK. and WOOD, AS., (2012). A novel heuristic for job shop scheduling, *International Journal of Customer Relationship Marketing and Management* (*IJCRMM*) **(In Press)**

2. MAQSOOD, S. Noor, S., KHAN, MK. and WOOD, AS., (2012). Hybrid Genetic Algorithm (GA) for job shop scheduling problems and its sensitivity analysis. Int. J. Intelligent Systems Technologies and Applications. Vol. 11, Issue 1/2, PP. 49-62

3. MAQSOOD, S., KHAN, MK. and WOOD, AS., (2011). A novel heuristic for low batch manufacturing process scheduling optimization with reference to process engineering, Special issue of Chemical Product and Process Modelling. Vol. 6, Issue 2, Article 8.

Conference Publication:

4. MAQSOOD, S., NOOR, S., KHAN, MK. and WOOD, AS., (2011). Sensitivity analysis of genetic algorithms for job shop scheduling problems. *26th International Conference of CAD/CAM, Robotics & Factories of the Future, Kuala Lumpur, Malaysia.*

5. MAQSOOD, S., HUSSAIN, I., KHAN, MK. and WOOD, AS., (2011). A novel Index Based Heuristic for job shop scheduling problems. *26th International Conference of CAD/CAM, Robotics & Factories of the Future, Kuala Lumpur, Malaysia.*

6. MAQSOOD, S., KHAN, MK. and WOOD, AS., (2011). A novel heuristic for low batch manufacturing process scheduling optimization. *International Conference of Computer Aided Process Engineering (CAPE). School of Engineering, Design & Technology. University of Bradford, UK.*

7. MAQSOOD, S., KHAN, MK. and WOOD, AS., (2010). A review of AI technique for manufacturing scheduling. *The 25th International conference of CAD/CAM, Robotics & Factories of the Future. Pretoria, South Africa.*