Ministry of High Education
Benha High Institute of Technology
Mechanical Engineering Technology Department

INVESTIGATING GROUP SCHEDULING IN A FLOW-LINE BASED MANUFACTURING CELL

A Thesis Submitted
In Partial Fulfillment of the Requirements for The Degree of
Master of Engineering and Technology

By

MAGDY EL-SAYED ABD-EL-RAHMAN HELAL
B.Sc. in Mechanical Engineering, Manufacturing Technology

Under Supervision of

Prof. AHMED S. HUZAYYIN

Professor of Mechanical Engineering
Dean of Benha High Institute Of Technology

Dr. Eng. MERVAT A. BADR

Mechanical Engineering Department

The National Research Center – Cairo

Dr. Eng. HANY M. ASSAF

Modern Academy for Computer Science

& Management Technology – Maady-Cairo

1999

Ministry of High Education
Benha High Institute of Technology
Mechanical Engineering Technology Department

A Master Thesis Titled

INVESTIGATING GROUP SCHEDULING IN A FLOW-LINE BASED MANUFACTURING CELL

By

MAGDY EL-SAYED ABD-EL-RAHMAN HELAL
B.Sc. in Mechanical Engineering, Manufacturing Technology

Approved By :

Prof. Farouk Ahmed Al-Gayar
Prof. Of Industrial Engineering
Faculty of Engineering
Helwan University

Prof. Abd-El-Latif Mohammad Hareedy
Prof. Of Industrial Engineering
Faculty of Engineering
Ain Shams University

Prof. Ahmed Soliman Huzayyin
Professor and Dean
Benha High Institute of Technology

Dr. Mervat Abd-El-Sttar Badr
Researcher, Mechanical Engineering Dept.
National Research Center

1999

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

ACKNOWLEDGMENT

I wish to express my deepest gratitude to **Prof. Ahmed S. Huzayyin** for his supervision, valuable advises, and his patience throughout this work.

I wish also to express my appreciation and sincere thanks to **Dr. Eng. Mervat A. Badr** for her continuous follow-up, guidance, and support during all stages of the research.

Thanks also to **Dr. Eng. Hany M. Assaf** for his encouragement at the beginning of the work.

And I'm grateful to **Ass. Prof. Mohamed I. Etman**; Faculty of Engi. & Tech. Helwan Uni. for support and important help He offered me in my first steps.

I'm also grateful to everybody in the department of human resources and trainingship, and the process planning department at El-Nasr for Automotive Manufacturing Co. in Helwan – Cairo, for all help and generosity they gave me.

ABSTRACT

Production scheduling models in which parts to be processed are classified into part-families, based on the principles of group technology is referred to as group scheduling (GS). The creation of part-families leads to the creation of a two-phase scheduling problem. First phase is to schedule part-families, and second is to schedule jobs within each family.

Benefits of such approach include setup time reduction and the simplification of the scheduling problem. This is suitable for the current trends in manufacturing, which indicate tendency towards batch production systems, larger product mix, reduced throughput times, and the wide application of group technology and cellular manufacturing systems. A typical application of GS is the scheduling of the static flow line manufacturing cell.

This research studies GS in a static flow-line manufacturing cell that is dedicated for the processing of a number of part-families. A selected number of GS heuristics are investigated and compared to each other with respect to makespan and total flow time, separately. Heuristics are classified, according to the amount of calculations involved, into single-pass methods, multiple pass methods and the iterative improvement techniques.

A number of modifications are proposed in order to explore the relative capabilities of the three classes of heuristic and to investigate the characteristics of the GS model. A recursive procedure for timetabling and calculating makespan and total flow time in multi-family cells is proposed. The procedure is capable of accounting for the possibility of the existence of the zero processing times in the multi-family cells. In addition, a case study was carried out to explore the applicability of GS in an existing typical batch production system.

Results of the research showed that the proposed modification could improve the performance of the GS heuristics under study. It was also found that the interaction between the two phases of scheduling in GS should be considered in developing GS heuristics. The iterative improvement techniques were found appropriate for GS not only because of their superiority over the simple methods but because they can handle the interaction between the two scheduling phases of GS as well.

Of the iterative methods, the tabu search heuristic is found to be preferable to the simulated annealing heuristic. Tabu search provides the ability to control its behaviour by the flexibility to consider different search-based information in defining its components so as to improve its performance.

Results also showed that the zero processing times have to be considered during timetabling calculations in multi-family cells, otherwise erroneous and misleading information would be obtained. Meanwhile it does not seem effective to consider the zero-processing times in the structure of the heuristics.

It is also found that due to the zero processing times, makespan should not be defined as the time span from the start of the first job on the first machine to the completion of the last job on the last machine. Instead it has to be defined as the largest completion time given that completion times for the zero-time jobs are set to zero. Makespan is not necessarily associated with the last job or the last machine. In addition, optimizing makespan can lead to a relatively good total flow time while the inverse is not true.

The case study showed that it is possible to apply GS in a traditional existing flow shops without formulating manufacturing cells physically.

NOMENCLATURES

A_i : First scheduling index in family phase in Hitomi.

A_{ij} : First scheduling index in job phase in Hitomi.

A_i^x : First scheduling index in family phase for subproblem x in CDS.

A_{ij}^x : First scheduling index in job phase for subproblem x in CDS.

AP_o : Initial acceptance probability in SA.

AP_x : Acceptance probability in iteration x in SA.

B_i : Second scheduling index in family phase in Hitomi.

B_{ij} : Second scheduling index in job phase in Hitomi.

B_i^x : Second scheduling index in family phase for subproblem x in CDS.

B_{ij}^x : Second scheduling index in job phase for subproblem x in CDS.

C_j : Completion time of job j.

C_{max} : Makespan.

d_j : Due date of job j.

GP : Switch variable between phases in SA.

F : Number of parts families.

F_j : Flow time of job j.

F_{max} : Maximum flow time.

i : Families index.

(i) : Family in position i in sequence.

j : Jobs index.

J_{ij} : Job j in family i.

(j) : Job in position j in sequence.

k : Machines index.

L_j : Lateness of job j.

L_{max} : Maximum lateness.

$\overline{L}$: Mean lateness.

M : Number of available machines.

N : Number of jobs.

N_T : Number of tardy jobs.

n_i : Number of jobs in family i.

O_{jk} : An operation for job j on machine k in traditional models.

P_{jk} : Processing time of job j on machine k in traditional models.

P_{ik} : Sum of processing times of all jobs in family i on machine k.

P_{ijk} : Processing time of job j within family i on machine k in GS.

r : Temperature reduction factor in SA..

r_j : Release date of Job j.

s : A move in TS

S_{ik} : Setup time of family i on machine k.

$S(x)$: Set of moves applicable to a trial solution x.

T : Set of tabu moves; the tabu-list.

T_i : Sequence index for family phase in NEH.

T_{ij} : Sequence index for job phase in NEH.

T_j : Tardiness of job j.

T_{max} : Maximum tardiness.

$\overline{T}$: Mean tardiness.

X : Iterations counter in SA.

$X(s)$: Set of solution accessible from x by S(x).

Y : Number of searches per iteration in SA.

ε : Reduction factor of the acceptance probability in SA.

AFM : Relative total flow time associated with a makespan.

AMF : Relative makespan associated with a total flow time.

ITM : Intermediate Term Memory in TS.

LTM : Long Term Memory in TS.

RELF : Relative total flow time.

RLEM : Relative makespan.

CONTENTS

CHAPTER 1

CHAPTER 1

INTRODUCTION

The topic of production scheduling has received renewed attention due to the changes in the business and technological environment. Current trends in manufacturing indicate a tendency towards reduced throughput times, lower inventory levels, smaller lot sizes, and the adoption of the innovative manufacturing concepts such as Total Quality Control (TQC), Just-In-Time (JIT), besides the wide applications of Group Technology (GT) and Cellular Manufacturing Systems (CMS). In addition, there is a trend away from the pure mass production organization towards batch production systems [1,2,3].

The open worldwide competition, increased capital cost, wide range of customer expectations, and shorter product life cycles, are factors that have dramatic impact upon manufacturing. Industrial firms have to produce in a larger product mix, smaller volumes, and shorter production runs. This in turn has dramatically increased the perceived importance of scheduling. Thus, improved production scheduling techniques incorporating the new manufacturing circumstances are essential to cope with the changing market place [2,4,5,6,7,8].

Industrial engineering has introduced the concept of GT to rationalize component design and manufacturing [9]. GT is a proven technique that has invalidated the inverse relationship between batch size and manufacturing costs. It has made available, for a small batch producer, economies that were earlier believed possible for mass production systems only [10].

Main advantages attributed to the application of GT are the reduced setup times and costs, the possibility of flow-shop pattern which in turn reduces the costs of material handling and buffering and simplifies production control, and the possibility to develop cellular layouts and formulating manufacturing cells [3,5,11,12,13].

Meanwhile, the basic efforts in GT are to identify families of parts that require similar processing on a set of machines. These machines are then grouped into manufacturing cells. That is: a part family is a set of similar jobs in terms of setup and processing requirements. The manufacturing cell can be regarded as a group of machines located in close proximity and dedicated for the manufacture of a specific number of part families [10,12,14]. Cells have been found to lead to the usual benefits rooted to GT while combining the flexibility of a job shop and the efficiency of the flow shop [15,16].

Manufacturing cells can be of two configurations: job shop cells, and flow line cells. A flow line layout has definite advantages over job shop layout. It implies simplified flows, minimum material handling and greater control of cell activities [17,18]. In fact, one of the reasons justifying a changeover of a general job shop to a CMS is the possibility of creating flow line cells and the use of efficient scheduling and sequencing procedures [15].

Production scheduling models associated with the application of GT is referred to as Group Scheduling (GS) [3,5,11,13]. GS is applied where parts to be processed are classified into different families, to take advantage of the similar processing requirements and the common setup times. A typical application of GS is the scheduling of the static flow-line cell [19].

The creation of part families leads to the creation of a two-phase scheduling model; first phase is to schedule part families, and second is to schedule jobs within each family. Scheduling is greatly simplified with the GS model in addition to the reduction of the setup times [11,15,20,21]. Results generally indicated that GS approach yields superior performance over the traditional single-phase models [14,22].

The traditional models can be regarded as a kind of GS in which there is only one family consisting of all the jobs, or alternatively, each family contains one job. However, for such a traditional situation consider N jobs to be processed through a number of M machines. It is required to identify the best sequence of processing of these jobs. If, for simplicity, the sequence of processing is maintained the same on all machines, which is termed permutation scheduling, then the number of possible sequences will be N!, only one of them is to be identified as the best.

If the N jobs are classified into F families each containing n_i jobs, then the number of permutation schedules to be considered will be $F!\times\prod_{i=1}^{F} n_i$. In a typical example this is about 15% of N! [11]. Thus, scheduling is remarkably simplified by the GS application.

The other main advantage offered by GS is the setup time reduction realized by processing of jobs in the same family in succession and having one common setup for them. Including setup time in processing time is a classical assumption in production scheduling. However, scheduling models that separate setup time from processing time were found to lead to better results than those with setup time included [21], which gives significance to GS.

Further, in real practice, similar jobs are often combined together to avoid changeover times. This use of informal part families is an application of GT. More advanced usage is to create formal part families, dedicate clusters of machines to these families, without rearranging of the equipment, and explicitly recognize part families in the scheduling process [23]. This means that even thought a cell is not formed, GS concepts can still be applied effectively with the existing shop layout [5].

This leads to the possibility to achieve advantages of GT and CMS by the use of GS, without formulating cells physically.

Nevertheless, although simpler than an equivalent traditional problem, GS problem is non-polynomial complete (NP-complete) and an optimal solution can, in practice, be found for small sized problems only. Researchers have adopted the heuristic approaches to produce near optimal solutions [21].

This research studies group scheduling in a static flow-line cell that is dedicated for the processing of a number of part-families. Selected GS heuristics are investigated and compared to each other. A number of modifications and suggestions are proposed in order to explore the characteristics of the GS model and the capabilities of the heuristics. Heuristics are classified according to the amount of computational efforts involved, into three main categories: the single-pass methods that generate a single solution, the multiple-pass methods that generate a finite number of solutions and the iterative improvement techniques which starts with an initial solutions and work iteratively to improve it.

And since it is possible to apply GS without rearranging the equipment into cells, this work is applicable to flow shops as well.

CHAPTER 2

CHAPTER 2

LITERATURE REVIEW

This chapter presents a review of GS literatures. Since GS is a form of the general scheduling problem, a brief review of production scheduling will be presented. Efforts to solve the scheduling problem in a flow shop are summarized hence to show how the traditional work was modified for GS applications. Afterward, GS model is defined and previous work is reviewed and presented.

2.1 PRODUCTION SCHEDULING

Scheduling is one of production decisions concerned with timing [24]. It is the function of determining an optimal implementation time plan for performing the necessary jobs [11]. The schedule is the sequence by which jobs are to be processed. It is defined as the listing of jobs to be processed through a workshop, and their respective start dates as well as other related information [26]. This is done after the production items and the quantities to be manufactured in specified time periods have been decided by production planning, and the production processes for those items have been determined by process planning [3,26].

The job is a task consisting of a collection of operations arranged in the technological order. It is a part, or a product, completed through a single or a number of machines, on each of them an operation such as turning, drilling…etc., is performed [3,11,26].

The scheduling problem had evolved tied to the scientific management in the early 20th century. In that time Henry Gantt developed the Gantt chart for controlling jobs and shop operations. The chart has been used as a visual aid in controlling machine loading in manufacturing shops, and hence the evolution of the problem of scheduling [3]. This may be the reason that most of terms used in the study of scheduling are related to manufacturing and industry, although scheduling problem appears in various fields [24].

Generally, there are three main categories of production scheduling situations [3,5,11,20,27,28]:

1. **Single machine scheduling**; determining the order of processing of jobs on a single machine.
2. **Flow shop scheduling**; scheduling in a flow shop, where the sequence of machines is the same for all the processed jobs.
3. **Job shop scheduling**; scheduling in a job shop where the sequence of machines differs for each job.

If the set of jobs available for scheduling does not change over time, the system is called static. If new jobs arrive over time the system is dynamic. Static models have proven more tractable than dynamic models. Moreover, static models have often captured the essence of the more complex, dynamic systems, and the analysis of static problems has frequently been useful in the study of the more general situations [26].

2.1.1 The General Scheduling Problem

The general scheduling problem can be stated as follows [3,5,20,27]:

1. A set of N jobs has to be processed.
2. A set of M machines is available.
3. The processing of job j (j = 1,2…N), on machine k (k = 1,2…M) is termed an operation.

4. For each operation there is an associated processing time; P_{jk} , which is the time needed for processing job j on machine k.
5. For each job there may be a release date r_j; which is the time at which job j is ready for being processed.
6. There may be a due date d_j at which job j should be completed.
7. The flow pattern or the order of machines for any job may or may not be fixed for all jobs; (cases of flow shop or job shop respectively).

The following assumptions appear frequently in literatures [3,11,24,27]:

1. Machines are always available and never break down.
2. There is only one machine of each type in the shop.
3. All jobs are available simultaneously at the commencement of processing.
4. Processing times are deterministic and known in advance.
5. Setup times are independent of the sequence of processing and are included in the processing times.
6. Transportation times are ignored or included in processing times.
7. The job consists of a strictly ordered sequence of operations.
8. Each machine can handle one and only one operation at a time.
9. Each operation can be performed by only one machine at a time.
10. No preemption is allowed: once a job is started it should be completed.
11. No relative priorities among jobs.

2.1.2 Complexity of the Scheduling Problem

The complexity of a scheduling algorithm refers to the execution time required to reach a solution. This time is usually expressed as a function of the number of jobs N. An algorithm is said to have a complexity of the order of the N^3;($O(N^3)$) if there exist a constant c such that the function cN^3 bounds its execution time. An algorithm whose complexity is bounded by a polynomial in N is a polynomial-time algorithm.

This algorithm is expected to be efficient and the associated problem is easy to solve. For the majority of the production scheduling problems there are no polynomial-time algorithms have been known. Such problems are called non polynomial complete (NP-complete), or NP-Hard [11].

In addition, the problem of scheduling is of combinatorial nature, that is the optimal solution is to be selected from among a large number of feasible alternatives. It is difficult to determine the optimal schedule in a real situation within a reasonable period of time due to the difficulty of acquiring the accurate information. Even, with the complete information available the task of optimal scheduling is not easy because the number of schedules to be considered is not small [11,29].

Consider the scheduling of N jobs on M machines. This is a combinatorial problem since there are $(N!)^M$ alternative solutions among which one is the optimal with respect to some measure of performance, and it can theoretically be found in a finite number of computational iterations. However, for example, in a small problem of scheduling 5 jobs on 8 machines, there exists $(5!)^8 = 4.3x10^{16}$ possible schedules. Using a high-performance computer that can evaluate one alternative in one microsecond, it will take about 1363 years to find the optimal solution [3].

A simplification can be made by considering the permutation schedules. A permutation schedule is one with the same job order kept on all machines. This will decrease the number of possible alternatives to N!. But, permutation scheduling can not guarantee optimality, and it may still be difficult to locate optima efficiently [5,19,20,26].

Accordingly, search for optimal solutions by complete enumeration procedures is not practical. And it is wiser to use effective theorems, rules, and heuristic algorithms rather than optimization and enumerative methods. Several theorems and algorithms have been already developed. The collection of research work concerned with the mathematical models and theoretical analysis related to scheduling is called the theory of scheduling [3,24].

2.1.3 The Theory of Scheduling

The theory of scheduling includes a variety of techniques that are useful in solving scheduling problems. The study of theory began in the early 1950's. An article by S.M. Johnson in 1954 is acknowledged as pioneering work. It presented an efficient optimal algorithm for solving the problem of scheduling N jobs on two machines in a flow shop, and generalized the method to some special cases of scheduling N jobs on three machines [24,27].

Jackson in 1955 and Smith in 1956 gave various optimal rules for single machine problems. These efforts formed the basis for much of the development of the classical scheduling theory. In the following years, several kinds of general-purpose operations research techniques were applied. Meanwhile heuristic methods were being developed for problems, which were proven difficult. By the late of the 1960s, the solid body of theory had emerged [27].

2.1.4 Scheduling Criteria

The goal of production scheduling is to define the optimal sequence of processing. Such a decision is accomplished with respect to a certain measure of performance, or a scheduling criterion. A measure of performance is usually a function of the set of completion times of jobs. If it is a non-decreasing function of completion times and is required to be minimized, the criterion is termed regular. Most of the scheduling criteria are regular. Following are some important related quantities employed in the scheduling criteria definitions [26]:

1. **Job completion time** (C_j). Time at which all processing of job j is finished.
2. **Job flow time** (F_j). Amount of time job j spends in the shop. $F_j = C_j - r_j$.

3. **Job lateness** (L_j). Amount of time by which the completion time of job j exceeds its due date. $L_j = C_j - d_j$.
4. **Job tardiness** (T_j). $T_j = \max\{L_j, 0\}$.

Job lateness can be positive or negative. Negative lateness represents better service than requested, while positive lateness represents poorer services. In many situations, distinct penalties and other costs will be associated with positive lateness, but no benefits will be associated with negative lateness. Therefore it is often helpful to work with a quantity that measures only positive lateness, which is tardiness [26].

Some of the important criteria are the following [3,5,11,20,24,26,27]:

1. **Maximum flow time**: $F_{max} = \max_{j=1}^{N}\{F_j\}$
2. **Mean flow time:** $\overline{F} = \frac{1}{N}\sum_{j=1}^{N} F_j$
3. **Makespan.** In static situations where $r_j = 0$ for all jobs, flow time for each job is its completion time. The maximum flow time equals the greatest completion time which is denoted by C_{max}.

$$C_{max} = \max_{j=1}^{N}\{C_j\}$$

C_{max} is known as makespan. It is the time span from the start of the first job on the first machine to the completion of the last job on the last machine.

4. **Maximum lateness or tardiness**:

$$L_{max} = \max_{j=1}^{N}\{L_j\} \quad \text{and} \quad T_{max} = \max_{j=1}^{N}\{L_j, 0\}$$

5. **Mean lateness or tardiness**.

$$\overline{L} = \frac{1}{N}\sum_{j=1}^{N} L_j \qquad \text{and} \qquad \overline{T} = \frac{1}{N}\sum_{j=1}^{N} T_j$$

6. **The number of tardy jobs.** If u_j is a binary variable that equals 1 if $C_j > d_j$ and 0 otherwise, then the number of tardy jobs is $N_T = \sum_{j=1}^{N} u_j$

Several other criteria exist. The selection of a criterion is based on the broad objective of the decision-maker. Makespan is the simplest to optimize and is commonly used [26].

2.2 FLOW SHOP SCHEDULING

In a flow shop M different machines exist, and each job is consisting of M operations, each requires a different machine. It is characterized by the unidirectional flow of work through the machines. The flow shop scheduling problem is relatively tractable compared to job shop scheduling [26].

First step of development for the solution of the problem dates back to Johnson’s work in 1954 for the 2-machine flow shop scheduling and its extension to the specially structured 3-machine problems, in which the second machine is dominated by the first and/or the third one [3,11,27].

2.2.1 Johnson’s Efficient Rule

Minimizing makespan for the 2-machine flow shop problem is the basic problem in the field of flow shop scheduling. It is called Johnson’s problem. Johnson’s rule to solve this problem states that job x precedes job y in an optimal sequence if $\text{Min}\{P_{x1}, P_{y2}\} \leq \text{Min}\{P_{x2}, P_{y1}\}$. In practice an optimal sequence is directly constructed with an adaptation of this rule [26]. An implementation of Johnson's rule with respect to makespan is described as follows [3]:

Step 1. Find the minimum processing time among the unscheduled jobs. Break tie arbitrarily.

Step 2. If it requires machine 1 (2) position the associated job in the first (last) free position in the sequence. If all positions are filled then stop.

Step 3. Remove the assigned job from consideration and return to Step 1.

In applying Johnson's rule, it is observed that the last job (N^{th}) can't be begun on machine 2 until all jobs have completed processing on machine 1. Hence one possible lower bound for the makespan is $L_1 = \sum_{j=(1)}^{(N)} P_{(j)1} + P_{(N)2}$ where (j) represents the job in position j in the schedule. Another lower limit L_2 exists. Observe that non of the jobs can begin processing on machine 2 until the first job in sequence has completed processing on machine 1, then $L_2 = \sum_{j=(1)}^{(N)} P_{(j)2} + P_{(1)1}$. The higher of the two limits is controlling.

The summations in both lower limits expressions are constant for any sequence. Only $P_{(1)1}$ and $P_{(N)2}$ are the affecting factors. Consequently, and as the sequence is developed gradually, it becomes logical to choose the lowest processing time P_{jk} and place job j first if k = 1, or last if k = 2 and proceed similarly with the remaining jobs [24]. This is how the Johnson's algorithm works [24].

Johnson had proven that his optimal 2-machine algorithm can be applied with respect to makespan if the maximum processing time on machine 2 is less than or equal to the minimum processing time on one or both machines 1 and 3. An artificial 2-machine scheduling problem is created by summing the processing times for each job on machines 1 and 2 (to be the processing time on first fictitious machine) and for each job on machines 2 and 3 (to be the processing time no second fictitious machine). Johnson's rule is then applied to this artificial 2-machine problem to obtain an optimal solution for the original problem.

2.2.2 Flow Shops with More than Three Machines

This is the general case. It is NP-complete. The optimal solution for it minimizing makespan is found using the branch and bound method developed by Ignall and Scharage in 1965. Still, this is impractical in real situations when the problem size increases. Besides, only permutation schedules are considered. Consequently, heuristic methods were developed to obtain near optimal solutions, in much less computational efforts. Heuristics are based on Johnson's rule in many cases [3,24,26].

Of the numerous heuristics are Petrove's method, developed in 1966, which is the direct extension of Johnson's efficient rule to the general flow shop scheduling problem, the CDS heuristic developed by Campbell, Dudeck, and Smith in1970, and the NEH heuristic developed by Nawaz, Enscore and Ham in 1983. CDS and NEH were identified as the best performing heuristics in flow shop scheduling [5,15,21,23,28,30,31].

2.2.2.1 Optimal solution for the general flow shop problem

The branch and bound method is an implicit enumeration algorithm for iteratively finding optimal solutions to discrete combinatorial problems by repeating branching and bounding procedures. The application of the method to solving the large-scale scheduling problems assures optimality. However, only permutation schedules are considered and hence this is a sub-optimal solution in the true sense [3]. The two fundamental procedures follow [3,11].

1. **The branching procedure**. Branching is represented by a tree similar to that shown in Fig.2.1. At level 1, each job becomes a node. At each node a lower bound on makespan is calculated. The node resulting in the smallest lower bound is selected for further branching by appending the remaining N-1 jobs to it hence moving to level 2 with N-1 nodes. Thus each node represents a partial schedule of the jobs and complete schedule is found at the Nth level.

2. **The bounding procedure**. Bounding is the process of calculating a lower bound on makespan for each partial schedule generated at each node. The node with the lowest bound is promising and is considered for further branching.

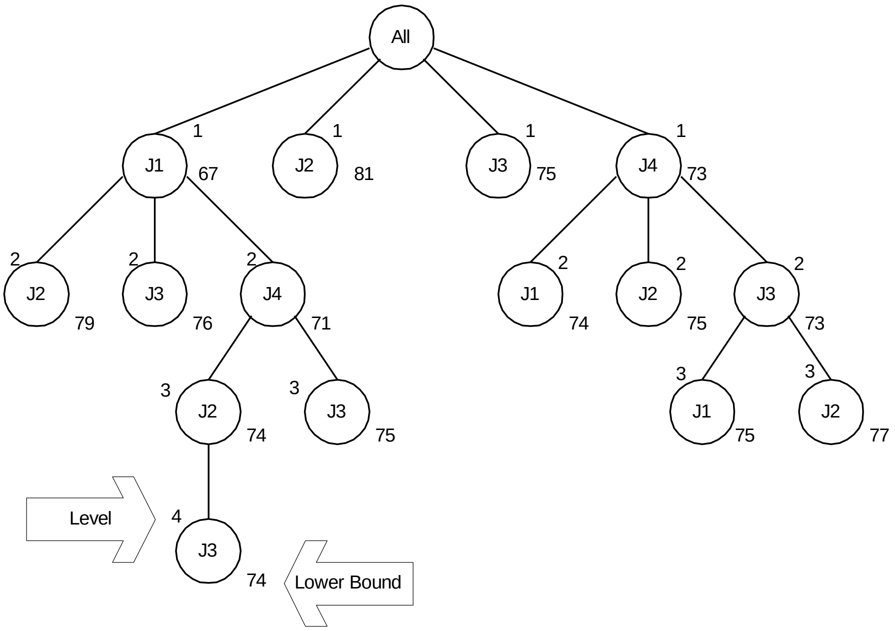


Fig.2.1 The branching tree for a four-job flow shop scheduling problem

2.2.2.2 Petrov's method

This method easily produces a fairly good job schedule. A single schedule is generated through the following steps [3]:

Step 1. For j = 1,2,…,N, calculate the two fictitious processing times:

$$P_j^1 = \sum_{k=1}^{h} P_{jk} \qquad \text{and} \qquad P_j^2 = \sum_{k=h'}^{M} P_{jk}$$

Where h = M/2, h'=h+1 for even M, and h = h'= (M+1)/2 for odd M.

Step 2. Apply Johnson's algorithm to this artificial 2-machine problem.

2.2.2.3 The CDS algorithm

This is a multiple application of Johnson's rule. Its power lays in two properties (1) using Johnson's efficient rule, (2) creating several schedules i.e. several chances of finding the optimal solution [28]. CDS provides for the generation of M-1 schedules through the construction of M-1 artificial 2-machine problems. In the k^th problem (k =1,2,…,M-1), the following two artificial processing times are calculated for each job j (j = 1,2,…,N):

$$P_{j1}^{k} = \sum_{m=1}^{k} P_{jm} = \text{Processing time for job j on first fictitious machine}$$

$$P_{j2}^{k} = \sum_{m=M+1-k}^{M} P_{jm} = \text{Processint time for job j on second fictitious machine}$$

The CDS algorithm is implemented in the following steps [31]:

Step 1. Set k = 1, for the first artificial problem.

Step 2. Construct the k^th 2-machine problem by calculating the two artificial processing times.

Step 3. Apply Johnson's rule to the k^th problem and obtain the k^th schedule and calculate its makespan.

Step 4. If k < M-1 then set k = k + 1 and return to Step 2.

Step 5. Identify the schedule with the minimum makespan from among the M-1 schedules as the best schedule.

2.2.2.4 The NEH algorithm

This heuristic assumes that a job with a higher total processing time needs more attention than a job with a lower total processing time. The schedule is developed gradually by appending jobs one by one to the existing partial schedule. The method is performed in the following steps [15,23,32]:

Step 1. Compute for each job j the summation $P_j = \sum_{k=1}^{M} P_{jk}$ and arrange jobs in the descending order of P_j.

Step 2. Pick the first two jobs in the order in Step 1 and find the best schedule for these two jobs by calculating makespan for the two possible combinations of them.

Step 3. Pick the job next in order and find the best its position by inserting it in each position in the existing partial schedule and checking makespan, keeping the relative positions of the scheduled jobs.

Step 4. If there is no more jobs then stop, otherwise return to Step 3.

2.3 TIMETABLING IN FLOW SHOP SCHEDULING

The following approaches are employed for timetabling and calculating makespan [3,11]:

1. **Graphical approach.** By representing the schedule on Gantt chart, timing data can be read. Fig.2.2 shows the Gantt chart for a schedule for 4 jobs on 5 machines. Completion times for jobs are read on the horizontal axis. Makespan is the completion time of the last job on the last machine. Fig.2.3 shows a part of a Gantt chart that explains how to calculate the start and completion times of a job. The rule is that a job can not be started on a machine unless the preceding job in

the schedule is completed on this machine (part A in Fig.2.3), or the job itself is finished on the previous machine (part B in Fig.2.3), the larger is taken.

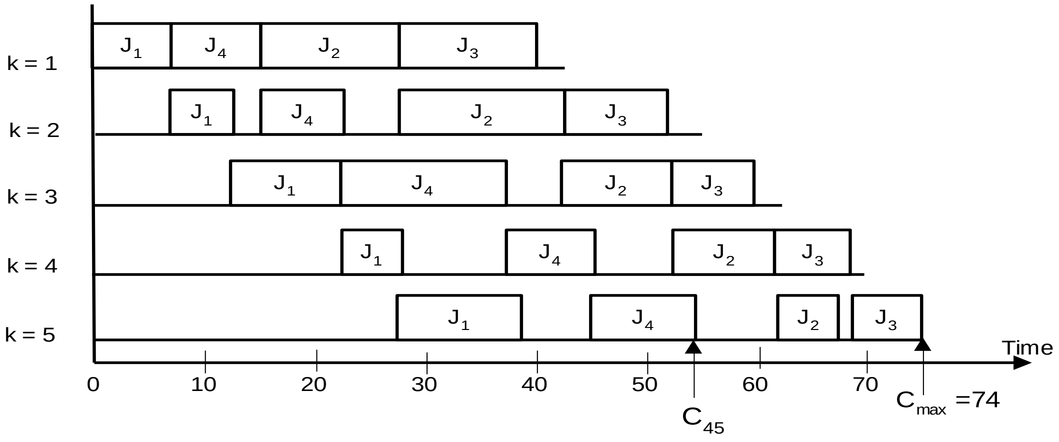


Fig.2.2 Gantt chart for optimal schedule of 4 jobs on 5 machines

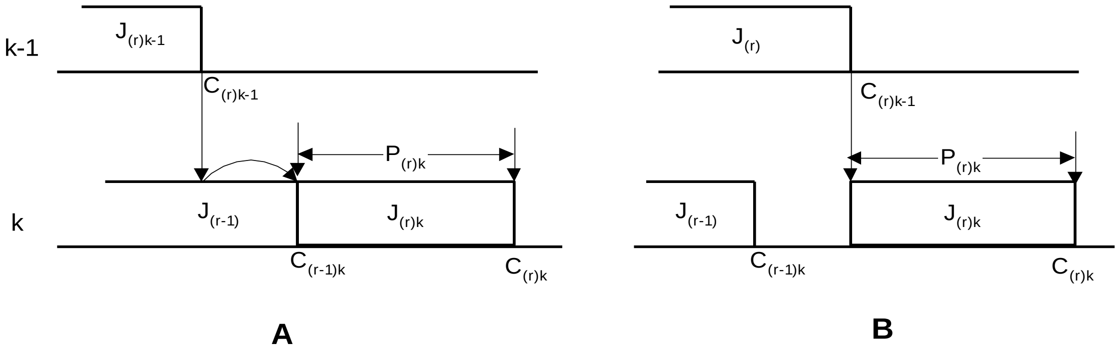


Fig. 2.3 Basics of calculating the completion time of the job in position r on machine k

2. **Mathematical formulae.** Referring to Fig.2.3, the completion time of job in position r on machine k can be calculated by the following recursive formula:

$$C_{(r)k} = \max\left\{C_{(r-1)k}, C_{(r)k-1}\right\} + P_{(r)k}$$

Position r is represented by (r). Actually, this is the mathematical representation of the rule used in the graphical method shown in Fig.2.3. Makespan is then given by:

$$C_{max} = \max\left\{C_{(N-1)M}, C_{(N)M-1}\right\} + P_{(N)M}$$

Another formulation of this recursive formula is reported in [17,18,30]. Letting σ be a partial schedule of some of the jobs, C(σ,k) be completion time of σ on machine k, and $C(\sigma_a,k)$ be completion time of job a on machine k after job a is appended to σ, the completion time of job a is computed by:

$$C(\sigma_a, k) = \max\left\{C(\sigma, k); C(\sigma_a, k-1)\right\} + P_{a,k}$$

If the total flow time of the jobs in σ is $F_{\sigma,}$ then the total flow time after appending job a to σ is calculated by the following formula:

$$F_{\sigma a} = F_\sigma + C(\sigma_a, M)$$

2.4 THE GROUP SCHEDULING MODEL

In the group scheduling; GS model, the N jobs are classified into F part families each contains n_i jobs where $\sum_{i=1}^{F} n_i = N$. Let the family index be i (i=1,2…F) and jobs in family i indexed by j (j =1,2…n_i). Jobs are to be processed on M machines. The setup time of family i on machine k (k =1,2…M) is denoted by S_{ik} and the processing time of job j in family i on machine k is P_{ijk}. The machines are assumed grouped into a manufacturing cell.

To show how the scheduling problem is simplified by GS consider a set of jobs; N = 10. Conventionally there are N! = 10! = 3,628,800 feasible schedules at each machine or 3,628,800 permutation schedules to be investigated. In GS, letting F = 3; and the size of each family such that n_1 = 4, n_2 = 3, n_3 = 3. The number of permutation schedules will be $F! \times \prod_{i=1}^{F} n_i! = 3! \times [4! \times 3! \times 3!]$ = 5184 that is only 14.82 % of N!.

Still, the GS problem is NP-complete. Efforts have been exerted to solve the problem in various environments. In the context of GS in flow-line cells and in flow shops as well, traditional flow shop scheduling algorithms have been modified for GS application where algorithms are to be executed in two stages for the two phases of GS [19].

2.4.1 Branch and Bound Solution for Group Scheduling

In 1976, Hitomi & Ham employed the branch and bound technique to obtain the optimal solution for the GS problem in a static flow shop. The scheduling criterion is minimizing makespan. Their work was based on the earlier work of Ignall & Schrage developed in 1965. The modified version is a two-stage application of Ignall & Sharage's model [19,21,23].

In 1985, Hitomi et-al. [11] explained the use of the branch & bound optimizing methodology, emphasizing that, in GS, both optimal family and job sequences must be determined simultaneously, and hence a new type of branch & bound procedures is required. The procedure according to [11] is described as follows. Comparing this version with that presented in Sse. 2.2.2.1 explains the two-phase nature of GS.

1. **Branching procedure**. In GS, there occur two kinds of nodes: family nodes and job nodes. Branching of families and branching of jobs are both required. The branching of families is made first. Then jobs within each family are branched from each of the family nodes created. The branching of the jobs in each family is repeated until all positions in that family are filled. Actually, job branching in each family is an application of traditional branch and bound method. Afterward, new family nodes are created by branching the unscheduled families at the best found job node. Then job branching is performed and so on. The tree starts with family nodes and ends with job nodes. The branching tree will look like that in Fig.2.4.

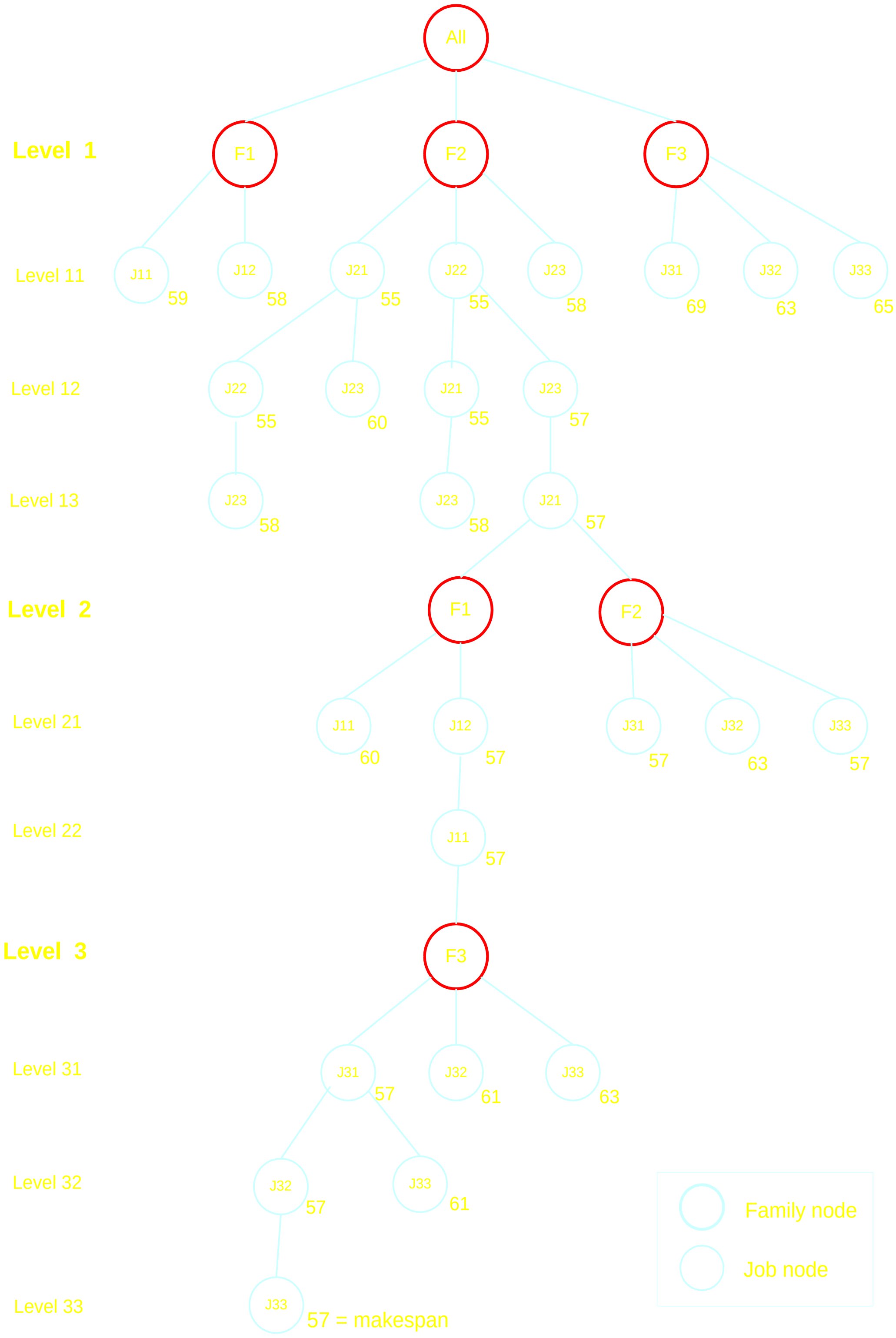


Fig.2.4 The branching tree for the group scheduling approach

2. **The bounding procedure.** The bounding procedure is the process of calculating a lower bound on makespan for the partial schedules generated at each job node. The node with the lowest lower bound is considered for more branching by appending the unscheduled families to it to formulate new family nodes. No lower bound is calculated at the family nodes.

2.4.2 Simple Heuristic Solution for the Group Scheduling Problem

Hitomi et-al. [11] introduced three optimizing algorithms for multi-stage especially structured GS problems, in which some well-defined relationships hold among the family setup times and job processing times at each machine. The problem according to them is reduced into an artificial two-machine problem to be solved by the application of Johnson's efficient rule. Further, they generalized Petrove’s heuristic to GS application to a obtain near-optimal solution. In this generalization a sequencing index employing the summation of the family setup times and job processing times, is used to develop an artificial two-machine problem when developing the family schedule. In finding the job sequences within families, Petrov’s method is employed traditionally to the jobs within each family.

In 1988, Hitomi [13] used the modified Petrov's method to obtain a near optimal solution for scheduling F part families in a flow shop. The objective is to minimize makespan. The heuristic is used in the job phase at first to find jobs sequences within each family, then a families sequence is developed given the jobs sequences developed before. In the family phase, the sequencing indices for each family are the summation of the family setup time and the processing times of the jobs in it.

In 1988, Grasso et-al. [21] reported that better results than those obtained using algorithms with setup time included, were obtained if setup times are separated from processing times in the heuristics, and this gave rise to the significance of GS.

They reviewed the work of Hitomi and co-workers to solve the GS problem in flow shops. Hence they state that the solution of GS problem can be simplified into the separated determination of job sequences within families, and the determination of families sequence. This does not take into account the possible interaction between the two phases of scheduling and hence leads to sub-optimal results. However, this simplification is useful in order to derive rapid and efficient GS heuristics from the setup time included heuristics. Consequently they proposed a four-level general framework for constructing GS heuristics employing the setup-time included procedures. The framework can be presented as follows:

Step 1: Determine a good job sequence in each family utilizing a setup-time included sequencing algorithm.

Step 2: On the basis of the methodology selected to derive the sequencing indices, define the sequencing indices for each family.

Step 3: Calculate the values of the sequencing indices for each family.

Step 4: Determine a good families' sequence utilizing the setup-time included procedure to the sequencing indices.

The effectiveness of this model depends on the methodology of defining the sequencing indices, and on the selected setup-time included heuristic that need not necessarily be better performing in the conventional scheduling. In addition Grasso et-al. classify the scheduling heuristics in two classes: single-shot that generate a single schedule, and multi-shot that generate several schedules one schedule of them is to be selected as best. They provided different variations for the Johnson's rule and the CDS heuristics modified according to the proposed general framework using different formulations of the sequencing indices.

In 1990 Allison [19] followed another approach to develop GS heuristics by combining two different traditional methods using one of them in each GS phase for scheduling F part families in a flow cell. He indicated that some researchers use the flow shop scheduling heuristics for GS starting with job phase and then switch to the family phase. Others start with family phase and then job phase. Alison classified the scheduling heuristics, as in [21] into single-pass, and multiple-pass heuristics, stating that single-pass heuristics are generally inferior, but require less computational efforts and he suggests a compromise approach to combine the two types of heuristics.

The question addressed by Alison is that where should the superior method (the multiple-pass) be employed; in job phase or in the family phase? He combined CDS and Petrov's method in four ways to study answering this question. The four combinations are Petrov/CDS, CDS/Petrov, Petrov/Ptreov and CDS/CDS, in each, the first is used in job phase and the second is used in family phase. The objective is minimizing makespan. Results showed that investing greater computational efforts by the use of the multiple-pass methods, in sequencing families yields better results, which means that the family phase is more important than the job phase. The number of jobs in each family does not strongly affect the relative performance of the heuristics.

In 1991, Wommerlov and Vakharia [23] considering the scheduling of a flow line cell that is dedicated for processing of a number of part families, they provided modified versions of CDS and NEH heuristics for GS situations. In both, the first phase is the families sequencing and the second is sequencing job within the families. The two methods were compared with the original heuristics and with some dispatching rules. It was found that the conversion from traditional scheduling to GS is generally advantageous in all operating environments considered, and in addition, picking the “wrong” scheduling procedure is less serious for procedures considering the part-family membership, than for job rules that ignore the part-families information, which is an added advantage to the GS model.

2.4.2.1 Hitmi’s heuristic

Phase 1: Job sequencing in each family

1- For each job j in family i, calculate the sequencing indices:

$$A_{ij} = \sum_{k=1}^{h} P_{ijk} \qquad \text{and} \qquad B_{ij} = \sum_{k=h'}^{M} P_{ijk}$$

where $h = M/2, h' = h+1$ for even M, and $h = h' = (M+1)/2$ if M is odd

2- Apply Johnson' s rule to A_{ij} and B_{ij} to obtain a sequence of jobs within each family.

Phase 2: Families Sequencing

1- For each family i, calculate the two artificial family processing times:

$$A_i = \sum_{k=1}^{h} \left[S_{ik} + \sum_{j=1}^{n_i} P_{ijk} \right] \qquad \text{and} \qquad B_i = \sum_{k=h'}^{M} \left[S_{ik} + \sum_{j=1}^{n_i} P_{ijk} \right]$$

2- Apply Johnson' s rule to A_i and B_i to obtain a family sequence.

2.4.2.2 GS-CDS heuristic

The heuristic constructs M-1 artificial 2-machine problem in the family phase and M-1 2-machine problems within each family in the job phase. Make denotes makespan. Families are treated as fictitious jobs having processing times as given in Step 3. In the job phase the procedure is actually an application of the traditional CDS for the jobs within each family, given the family sequence found in phase 1. It is applied in each family independent of the other families.

Phase 1: Family sequencing

Step 1. Set x = 1, where x is the subproblem index. Let $Make^{o} = \infty$.

Step 2. If $x \geq M$ then switch to Phase 2, else go to Step 3.

Step 3. Calculate for each family i two artificial family processing times:

$$A_i^x = \sum_{k=1}^{x} \left[S_{ik} + \sum_{j=1}^{n_i} P_{ijk} \right] \quad , \quad B_i^x = \sum_{k=M-x+1}^{M} \left[S_{ik} + \sum_{j=1}^{n_i} P_{ijk} \right]$$

Step 4. Apply Johnson's rule and generate a families sequence. Let the total flow time be Makex

Step 5. If $Make^{x-1} > Makex$ then $Make^{x} = Makex$, and keep the sequence. Else $Make^{x} = Make^{x-1}$

Step 6. Set x = x + 1 , and go to Step 2

Phase 2: Job sequencing within each family

Step 1. Set i = 1

Step 2. If $i \geq F + 1$ then stop, else go to Step 3

Step 3. Set x = 1 , $Make^{o} = \infty$

Step 4. If $x \geq M$ then go to Step 8, else go to Step 5

Step 5. Calculate for each job j in family i , for the x^{th} subproblem, the two artificial processing times on the first and second artificial machines as follows:

$$A_{ij}^{x} = \sum_{k=1}^{x} P_{ijk} \quad \text{and} \quad B_{ij}^{x} = \sum_{k=M-x+1}^{M} P_{ijk}$$

Step 6. Apply Johnson's rule and find a job sequence.
Let total flow time be Makex.

Step 7. If $Make^{x-1} > Makex$ then let $Make^{x} = Makex$ and keep the sequence; Else let $Flow^{x} = Flow^{x-1}$. Set x = x + 1, and go to Step 4

Step 8. Set i = i + 1 and go to Step 2

2.4.2.3 GS-NEH heuristic

As in the CDS, families in phase 1 are treated as fictitious jobs having processing times P_{ik}, as in Step 1. The families sequence is kept during the second phase. In the second phase the traditional NEH is applied to jobs within each family independently of the other families.

Phase 1: Family Sequencing

Step 1. Calculate for each family i:

$$P_{ik} = \left[\sum_{j=1}^{n_i} P_{ijk} + S_{ik} \right] \qquad \text{and} \qquad T_i = \sum_{k=1}^{M} P_{ik}$$

Step 2. Arrange families in descending order of T_i.

Let ω be an index for this ordered list of families.

Step 3. Pick the first and second two families in the list and find the best sequence for them. Set $\omega = 3$.

Step 4. If $\omega = F + 1$ then switch to Phase 2, else go to Step 5.

Step 5. Pick the family in the ω^{th} position in the list. Find the best position for it by inserting it in each of the ω positions in the partial sequence found in the previous trial, without changing the relative positions of the previously assigned families. Set $\omega = \omega + 1$ and go to step 4.

Phase 2: Job sequencing within families.

Step 1. Set $i = 1$

Step 2. If $i = F + 1$ then stop, else go to Step 3.

Step 3. For each job j in family i calculate:

$$T_{ij} = \sum_{k=1}^{M} P_{ijk}$$

Step 4. Rank jobs in descending order of T_{ij}. Let θ be an index for this ordered list.

Step 5. Pick the two jobs in the first and second positions in the list, and find the best sequence from the two possible sequences for the two jobs by calculating makespan for them. Set $\theta = 3$.

Step 6. If $\theta \geq n_i + 1$ then go to Step 8, else go to Step 7.

Step 7. Pick the job in the θ^{th} position in the list and find the best sequence by inserting it in each of the θ positions in the partial sequence found in the previous trial, without changing the relative positions of the previously assigned jobs. Set $\theta = \theta + 1$ and go to Step 6.

Step 8. Set $i = i + 1$, and go to Step 2.

2.4.3 Iterative Improvement Techniques to Solve the GS Problem

The drawback of the simple GS heuristics is that they are not able to consider the possible phase's interaction of GS. In addition, the number of solutions generated is small while the problem is combinatorial. With the increase of computer capabilities, researchers developed iterative improvement techniques to solve the GS problem. The iterative methods seem able to consider the phase's interaction. Besides, the number of solutions investigated is larger. Of the generic techniques applied are the simulated annealing (SA), the tabu search (TS) and the genetic algorithm methods.

2.4.3.1 Simulated annealing heuristic

Simulated annealing (SA), is a randomized improvement algorithm that has been used to derive near global optimal solutions for combinatorial intractable problems. It was originally developed as a simulation model for a physical annealing process (hence the name "simulated annealing"). Simply speaking it is an iterative improvement technique, in which an initial solution is repeatedly improved by making small local alternations until no such alternation yields a better solution. SA randomizes this procedure in a way that allows for occasional changes that worsen the solution in an attempt to reduce the probability of becoming stuck in a poor but locally optimal solution. The basic concepts of SA were developed by Kirkpatrick et-al. in 1983 [15, 17, 33, 34].

In 1993, Sridhar and Rajendran [17] proposed a SA heuristic, for scheduling a single family flow cell with the objective of minimizing total flow time. The heuristic is in two steps: first is to generate an initial good sequence using a flow shop algorithm which they believe to be a good initial seed generator with respect to flow time. They prefer to use a good initial solution to using a random one as usually done with SA applications. The second level is an iterative improvement procedure by the SA. They concluded that iterative improvement heuristics are effective in tackling problems that are computationally intractable. They recommended using an acceptance probability with SA that is dependent on the change in the objective function value.

In 1990, Vakharia and Chang [15] provided another SA based heuristic for the scheduling in a static flow line cell. The cell is dedicated for the processing of a number of part families. The objective is to minimize makesapn. They start with a random initial schedule and iteratively improved it using the SA approach. They used an acceptance probability that is independent of the change in the objective function value. The proposed SA heuristic is structured to spend 90% of the iterations in the job phase and 10% in family phase. The heuristic is presented later.

Principles of SA can be presented as follows. It is based on the analogy between the annealing of a solid and the optimization of combinatorial problems. Solids are annealed by raising the temperature to a maximal value at which particles randomly arrange in the liquid phase, followed by cooling to force particles into a low-energy state of a regular lattice. At high temperatures all possible states can be reached. Lowering the temperate decreases the number of accessible states and the system finally will be frozen into its ground state. In combinatorial optimization, a similar situation takes place. The system may occur in many different configurations. Any configuration has a cost that is given by the value of a relevant cost function. Similar to the simulation of the annealing of solids, one can statistically model the evolution of the system that has to be optimized into a state (configuration) that corresponds to the minimum value of the cost function [17].

In its usual form, SA algorithm starts off from an arbitrary initial configuration. In each iteration, by slightly perturbing the current configuration a new configuration is generated. The difference in cost between the two configurations is compared with an acceptance criterion that tends to accept improvements but also admits, in a limited way, deteriorations in cost. Initially, the acceptance criterion is taken such that deteriorations are accepted at a high probability. As the optimization process proceeds, the acceptance criterion is modified such that the probability of accepting deteriorations decreases. At the end of the process the acceptance probability is zero [33].

Temperature is simulated as a control parameter that acts like an iteration counter for the algorithm. Temperature is successively reduced by means of a reduction factor. When temperature reaches a pre-specified value (the freezing temperature) the procedure is terminated. At every temperature step, iterations are carried out for a number of times in search for better solutions [17].

The strength of the method lies in the fact that inferior solutions are accepted with a certain acceptance probability (AP) with the hope that the algorithm can clear local optimal troughs and find global optima. The standard and original choice for the acceptance criterion is given at any temperature step t by the Metropolis formula as [33]:

$$AP_x = \begin{cases} \exp\left(\frac{-\Delta}{X}\right) & \text{if } \Delta > 0 \\ 1 & \text{if } \Delta \leq 0 \end{cases}$$

Where Δ is the change in cost between the current configuration and the new one generated from it. The advantage of having the acceptance probability dependent on the change of the cost function is that solutions that cause drastic changes in the cost are avoided at the lower temperature changes [17].

The algorithm is generic and needs to be modified in the context of the problem in hand. A generic SA algorithm is given below for a minimization problem [17].

```
Step 1.  Get an initial solution S.
Step 2.  Get an initial temperature T > 0.
Step 3.  While not yet frozen do the following.
         Step 3.1.  Perform the following loop L times.
                    Step 3.1.1  Pick a random neighbor $S'$ of S.
                    Step 3.1.2  Let Δ = cost( $S'$ ) – cost( S ).
                    Step 3.1.3  If $\Delta \leq 0$ then set S = $S'$
                    Step 3.1.4  If Δ > 0 then set S = $S'$ with probability
                                exp(-Δ/T)
         Step 3.2.  Set T = rT , (where r is a reduction factor)
Step 4.  Return S
```

SA was applied in [15] for GS starting with an initial schedule developed randomly. The initial schedule is perturbed using a pair-wise interchange of jobs or families in order to find a neighbor solution to the current one. During search process, an inferior solution may be accepted to replace the current solution, according to the value of the acceptance probability. The acceptance probability is simply set to an initial value and is reduced each iteration by a constant reduction factor such that it reaches zero at the end of the iterations. The acceptance probability is independent of the change in the objective function value.

A switch parameter to direct the search process either to family phase or to the job phase is employed. This parameter is expected to have significant effect on the performance of the heuristic. Its value of 0.1 used in [15] leads to spending 90% of the iterations to the job phase, and 10% to the family phase. Parameters and variables used in this SA algorithm in [15] follow.

X : The number of iterations (X = 25)

Y : Number of searches per iteration (Y = 50)

AP_o : Initial value for the acceptance probability (AP_o = 0.5)

GP : Switch variable between the two phases.
A random number is sampled, if less than GP, the family sequence is perturbed, else job sequence is perturbed.

ε : The reduction factor of the acceptance probability $\varepsilon = AP_o / X$. The acceptance probability for iteration x is $AP_x = AP_{x-1} - AP_0 / X\ \varepsilon$

The SA heuristic is described as follows where Make is makespan.

Step 1. Set X, Y, AP_o, and GP. Set $\varepsilon = AP_o / X$.

Step 2. Generate a random schedule. This includes a complete sequence for all jobs (Ω^o), a family sequence (τ) and a sequence for jobs within each part family (μ_f); where f = 1, 2 , ... , F and let this be the current solution with a makespan $Make^o$. Let Ω^* represent the incumbent solution with makespan $Make^*$. Let $\Omega^* = \Omega^o$ and $Make^* = Make^o$. Set x = 0.

Step 3. Let x = x + 1. If x > X then stop, else set y = 0 and continue.

Step 4. Set y = y + 1. If y > Y then set $AP_{x+1} = AP_x - \varepsilon$ and go to Step 3, Else go to step 5.

Step 5. Generate a random number ν ($0 \le \nu \le 1$). If $\nu \ge GP$, then go to Step 7. Else go to Step 6

Step 6. In this step the order of jobs within each family will not change. Generate a random number $\nu1$ ($1 \le \nu1 \le F$). Interchange the family in position $\nu1$ with that in position $\nu1+1$ (if $\nu1 = F$, interchange the family in position F with that in position 1) and generate a family sequence τ^1. Based on τ^1 specify a new complete job sequence Ω^1 and calculate $Make^1$.

(a) If $Make^1 \ge Make^*$ then go to (b), else let $\Omega^* = \Omega^1$ in the incumbent solution, set $Make^* = Make^1$ and go to (b).

(b) If $Make^1 \ge Make^o$ then go to (c), else let $\tau = \tau^1$, $\Omega^o = \Omega^1$ in the current solution and set $Make^o = Make^1$ and go to Step 4.

(c) Generate a random number $\nu2$ ($0 \le \nu2 \le 1$). If $\nu2 \ge AP_x$ then go to Step 4, else let $\Omega^o = \Omega^1$, $\tau = \tau^1$ in the current solution, and set $Make^o = Make^1$ and go to Step 4.

Step 7. In this step the sequence of part families stays the same. Generate a random number $\nu1$ ($1 \le \nu1 \le N$), where N is the total number of jobs in all the families. Let f_1 be the family in which job $\nu1$ is included. Interchange the job in position $\nu1$ with that in position $\nu1+1$ (if $\nu1$ is the last in the family f_1, interchange job in position $\nu1$ with that in

Position 1 for the same family f_1) in Ω^o. Let the new sequence be $\mu^1_{f_1}$ for family f_1 and the new complete sequence be Ω^1 with $Make^1$.

(a) If $Make^1 \ge Make^*$ then go to (b), else let $\Omega^* = \Omega^1$ in the incumbent solution, set $Make^* = Make^1$ and go to (b).

(b) If $Make^1 \ge Make^o$ then go to (c), else let $\mu_{f_1} = \mu^1_{f_1}$, $\Omega^o = \Omega^1$ in the current solution, and set $Make^{o} = Make^{1}$. Go to Step 4.

(c) Generate a random number $\nu 2$ ($0 \leq \nu 2 \leq 1$). If $\nu 2 \geq AP_{x}$ then go to Step 4, else let $\mu_{f_1} = \mu^{1}{}_{f_1}$ and $\Omega^{o} = \Omega^{1}$ in the current solution, and set $Make^{o} = Make^{1}$ and go to Step 4.

2.4.3.2 Tabu search heuristic

In 1993, Kapov and Vakharia [12] developed an iterative algorithm based on the tabu search (TS) approach, for GS in a flow line cell dedicated for the processing of a number of part families. The objective is the minimization of makespan. TS is a meta-strategy developed to improve the solvability of the hard combinatorial optimization problems. The proposed algorithm iterates between the two phases of scheduling, keeping a limited track of the search trajectory in order to guide the search out of the local optimum. Short-term-memory containing information about a predefined number of recent iterations is employed so that a gain in a recent iteration is not wasted in the next near iterations, and hence avoiding being trapped in local optima.

In addition, a long-term memory is used to restart or rerun the procedure for a predefined number of times (e.g. 5), by the generation of a new initial family sequence using the information gathered during the search iterations in the long term memory in the previous run. However, in [12] job sequences within families are randomly set after the development of the new family seed solution, which does not seem logical.

TS has its origins in combinatorial optimization procedures applied to some non-linear problems in the late 1970s, and subsequently applied to a divers collection of problems. It is an adaptive procedure with the ability to make use of other methods, which it directs to overcome the limitations of local optimality. It helps guide such methods (may be as a subroutine), to continue exploration without falling back into a local optimum. Latest search and computational comparisons involving travelling salesman problem, graph theory, integrated circuit design and timetabling problems has likewise disclosed the abilities of the TSto obtain high quality solutions with modest computational effort, generally dominating alternative methods tested [35,36].

The strategic principles of the TS in a broader sense have been laid out in [35, 37]. These are summarized hereafter.

Consider a combinatorial optimization problem given by : Min $c(x) : x \in X$. Where X is the set of vectors that can be feasible solutions, and c(x) is the value of a relevant penalty function designed to assure optimality or, simply, it is the objective function.

Given a trial solution $x \in X$, let s be a *move* that leads from one trial solution x to another solution in the neighborhood of x. Simple definition of a move is that it is a transition between solutions [36]. For each $x \in X$ there is a set S(x) that consists of all those moves $s \in S$ applicable to x, that is:

$$S(x) = \{ s \in S : x \in X(s) \}$$

and

$$X(s) = \{ x \in X : s \in S(x) \}$$

Consider the following simple hill climbing heuristic.

Step 1. Select an initial $x \in X$

Step 2. Select some $s \in S(x)$ such that $c(s(x)) < c(x)$. If no such s exists then x is a local optimum and method stops, else continue to Step 3

Step 3. Let $x = s(x)$ and return to Step 2

This means that: start with a solution x and apply moves to it. If exists, take the new solution s(x) resulting from applying s to x such that the value of the objective function of s(x) is better than that of x , and let it be the current solution. Otherwise, this x is a local optimum and the search process stops. The algorithm is very simple but the local optimum may not be the global optimum. TS then guides such a heuristic to continue exploration without falling in a local optimum, and to overcome the absence of feasible moves.

In order to avoid local optimum, a subset T (called tabu-list), of S is created whose elements are called tabu (forbidden) moves. That is some moves applicable to x may be prevented if they are included in T. Elements of T are the moves (or solutions) those violates a set of tabu conditions (e.g. linear inequalities or logical relationships) i.e.

$$T(x) = \{ s \in S : s(x) \text{ violates the tabu conditions} \}$$

The tabu conditions are defined in the context of the application. The list T reflects the recent move history of the search [36]. The size of T is t. It may be fixed and may be variable. Thus elements of T are determined based on historical information from the search process, extending back to t iterations in the past.

Given T and employing an evaluation function denoted by OPTIMUM that is used to help selecting new current solutions, a simple TS heuristic follows.

Step 1. Select an initial $x \in X$ and let the best solution be $x^* = x$.
Set the iteration counter k = 0 and begin with T empty.

Step 2. If S(x) – T is empty, go to Step 4.
Otherwise, set k = k +1 and select $s_k \in S(x) - T$ such that
$s_k(x) = \text{OPTIMUM}(s(x) : s \in S(x) - T)$

Step 3. Let $x = s_k(x)$. If $c(x) < c(x^*)$ then let $x^* = x$.

Step 4. If the chosen number of iterations has elapsed either in total or since x^* was last improved, or if $S(x) – T = \Phi$ upon reaching this step directly from Step 2, then stop. Else update T and return to Step 2.

In the above heuristic, in iteration k there is a solution x on which the applicable moves are performed. The move, which leads to the best result with respect to OPTIMUM, is chosen. This move $s_k(x)$ is used to update T while the solution reached by it becomes the new current solution. That is; at each iteration the best move is chosen not an improving one. This is reasonable since the previous current solution has been consumed up and all moves applicable to it were performed. Keeping it means being in a local optimum. There is nothing to do but to move from this current solution to another one, whatever it is, hoping that from this new one, a better solution could be accessible. A natural choice of OPTIMUM is given by selecting $s_k(x)$ such that:

$$c(s_k(x)) = \text{Minimum} (c(s(x)) \ : \ s \in S(x) - T)$$

This simple rule that selects the minimum c(s(x)) subject to tabu conditions has in fact proved successful in a variety of applications [35]. A similar straightforward but effective form of the set T is given by:

$$T = \left\{ s^{-1} : s = s_h \text{ for } h > k - t \right\}$$

Where k is the iteration index and s^{-1} is the inverse of move s, thus $s^{-1}(s(x)) = x$. That is T is the set of those moves that would reverse (undo) one of the moves made in the t most recent iterations. Consequently, the goal more general is to avoid returning to a previous solution state, e.g. to a previously visited solution where the best available move for leaving it will be the same as before. Each iteration, T is updated each iteration by setting $T = T - s^{-1}_{k-t} + s^{-1}_{k}$. The minus and plus signs indicate deleting and appending elements to T. Upon appending a new element to T the oldest element is removed.

Another important component of TS is the aspiration level function. The role of the aspiration level function is to provide added flexibility to choose good moves, by allowing the tabu status of a move to be overridden if this aspiration level is fulfilled. The aspiration level for a specific tabu move is fulfilled if $c(s(x)) < \text{Best}(c(x))$, i.e. the OPTIMUM function value for that move is better than the overall best existing value. This tabu move is then performed.

Intermediate and long term memories (ITM and LTM) are another two components of the TS, the functions of which are to achieve regional intensification and global diversification of the search. The tabu-list T fulfills the function of a short term memory.

ITM records and compares features of the best trial solutions generated during a particular period of search. Features that are common to all or the majority of the best trial solutions are taken to be a regional attribute of a good solution. The method then seeks new solutions that exhibit these features by correspondingly restricting the available moves during a subsequent period of regional search intensification. For example, in the traveling salesman problem any current solution will incorporate some of the total edges in the problem. After some initial number of iterations, the method can identify those edges which are often contribute to the good solutions, hence discarding the other edges not incorporated in any tour and devoting itself to the resulting smaller problem.

LTM diversifies the search based on principles that are roughly the reverse of those for the ITM. The latter focuses more intensively on regions that contain good solutions as experienced during the search process. LTM guides the process to regions that contrast with those examined so far. For a traveling salesman problem, a LTM is a count of the number of times each edge appears in the trial tours generated during the search process, and new good starting solutions are generated such that tending to avoid those edges most recently used before, and search at new regions.

Kapov and Vakharia [12] define the elements of the TS as applied to GS as follows. Let Ω be a complete feasible schedule that consists of a sequence of part-families and a sequence of jobs within each family. There exist two neighborhoods for Ω:

N_i (Ω) Obtained from Ω by exchanging families in positions i and i + 1,
Where i = 1,…,F-1, keeping the order of jobs within families.
If I = F then exchange the last and the first families in Ω.

N_j (Ω) Obtained from Ω by exchanging jobs j and j + 1 in family i, keeping the oreder of families unchanged, where i = 1,…,F; j = 1,…, n_i –1.
If j = n_i then exchange the last and first jobs in the family i.

Let the transition from Ω to N_i(Ω) termed f-move (for family-move), and transition from Ω to N_j(Ω) termed j-move (for job-move). A value of a move is the difference between makespans after and before the move. Iteration is completed when the whole neighborhood of a current schedule is evaluated and the best move is identified and performed.

Two types of tabu-list are used to contain information necessary to forbid a number of recent moves, say t moves. The f-tabu-list contains families that were moved from position i + 1 to position i in the t recent iterations, hence a family being in the list can't be moved back to position i + 1. Similarly, j-tabu-list is used for j-moves. The procedure will iterate between the two types of moves. When there are no improvements in a predefined number of f-moves, j-moves are performed. If no improvement during a predefined number of j-moves, return to f-moves, and so on until a stopping criterion is reached.

Both fixed tabu-list size and variable tabu-list size were addressed in [12]. The variable list size was found better and was recommended. It is operated as follows. Given an initial tabu-list size, if there is no improvement in the prescribed number of iterations, decrease the list size to intensify the search in the current region. Following that, when there is no improvement in the prescribed number of iterations, increase the list size to diversify the search.

Specifically, the heuristic starts by performing f-moves with the initial f-tabu list size of Int(F/2). If there is no improvement in the last 5F iterations, the list size is decreased to Int(F/3). After 2F iterations performed without improvement the list size is increased to Int(F/0.5). After 3F iterations without improvement, the initial list size is retained and the process switches the job phase and j-moves are performed. The initial j-list size is Int(N/F). After Int(N/3) iterations without improvement decrease the list size to Int(N/2F). After more Int(N/3) iterations without improvement increase the list size to Int(N/0.5F) and after another Int(N/3) without improvement retain the initial size and return to the family phase.

Kapov and Vakharia used ITM and LTM. Both were called LTM. Both are based on frequencies (i, p), denoting the number of times a family i occupied position p in trial schedules during the search process. Initially it is a zero F x F matrix. Each time a new current solution is constructed, the entries of the frequency matrix corresponding to families and their respective positions in the current schedule are increased by one.

LTM is used to create a new starting family sequence. The heuristic is restarted a number of times with the new starting solution generated using LTM. LTM based on maximal frequencies; termed LTM_MAX, (actually ITM) is used to provide search intensification by creating a new starting family sequence by following the procedures below:

1 . Take the maximal entry in the LTM matrix, say (i_1 , p_1) and fix the family i_1 in the position p_1.
2 . Delete the row i_1 and the column p_1.
3 . Repeat 1 and 2 until a new family sequence is cereated.
4 . Jobs are randomly scheduled within families.

A LTM_MIN was used instead of LTM_MAX, involving taking the minimal entry in the matrix instead of the maximal entry. This corresponds to the use of LTM in the basic TS. LTM_MAX was found preferable in [12].

The commonly used criterion for defining the aspiration level function was employed in [12]. It is to perform a tabu move if the resulting makespan is better than the best previously found.

The TS heuristic using variable tabu list sizes and a LTM_MAX is described below. The number of the LTM restarts is five, that is the heuristic is rerun five times at the beginning of each time a new initial solution is generated using LTM_MAX.

Step 1. Initialize the f-tabu-size and j-tabu-size. Set the required number of LTM restarts. Set LTM matrix = 0.

Step 2. If LTM = 0 then generate a random families sequence, Else generate a families sequence using LTM_MAX. Complete the schedule by randomly generating a jobs sequence within each family. Let this be current solution Ω^o with makespan Make^o. Let Ω^* represent the incumbent solution with makespan Make*.

Set $\Omega^* = \Omega^o$, Make* = Make^o and LTM = LTM + 1.

Step 3. Start counting iteration for family exchange. Set f-iter = 0.

Step 4. Stopping criterion for family exchanges:

(a) If no improvement in the last 5F iterations with the initial f-list size then decrease the f-list size and go to (b).

(b) If no improvement in the last 2F iterations with the decreased f-list size then increase the f-list size and go to (c).

(c) If no improvement in the last 3F iterations with the increased size then set the list size to its initial value and go to Step 6.

IF at any point there is an improvement then go to Step 5

Step 5. **Family exchange phase of search.**

Evaluate completely the neighborhood N_i (Ω^o) and select the best exchange of families. Denote the new complete sequence by Ω^1 and its makespan by $Make^1$. If $Make^1 < Make^*$ then set $\Omega^*=\Omega^1$ and $Make^* = Make^1$. Set $\Omega^o = \Omega^1$, $Make^o = Make^1$ and go to Step 4.

Step 6. Start counting iterations for job exchanges.

Set j-iter = 0 and go to Step 7.

Step 7. Stopping criteria for job exchanges. Set j-iter = j-iter + 1.

(a) If no improvement in the last Int(N/3) iterations with the initial j-list size then decrease the j-list size and go to (b).

(b) If no improvement in the last Int(N/3) iterations with the decreased j-list size then increase the j-list size and go to (c).

(c) If no improvement in last Int(N/3) iterations with the increased size then set the list size its initial value and go to Step 9.

If at any point there is an improvement then go to Step (8)

Step 8. **Job exchange phase of search.**

Evaluate completely the neighborhood N_j (Ω^o) and select the best exchange of jobs. Denote the new complete sequence by Ω^1 and its makespan by $Make^1$. If $Make^1 < Make^*$ then set $\Omega^*=\Omega^1$,

$Make^* = Make^1$. Set $\Omega^o = \Omega^1$, $Make^o = Make^1$ and go to step 7.

Step 9. If the incumbent solution was changed during the job exchange phase of search (Step 8), then go to Step 3. If the required number of LTM restarts has been performed then stop the search, else go to Step 2.

2.4.3.4 Genetic algorithm

In 1994, Sridhar and Rajendran [18] proposed a genetic algorithm for GS in a flow line cell. The cell is dedicated for the processing of a number of part families. The objective is minimizing makespan, followed by minimizing the total flow time and finally the bi-criteria of minimizing the makespan and total flow time. The algorithm is relatively complex, executed in three steps. In the first step, two conventional flow shop heuristics; the NEH, which minimizes makespan, and another one referred to as RC (developed by Rajendaran and Chaudhuri in 1992), which minimizes the total flow time, are used to develop two families sequences. By applying pair-wise adjacent interchange to each sequence, two family chromosomes are formulated, each consists of F sequences.

Step two is job sequencing within families. NEH and RC are modified for cell scheduling such that the sequencing indices in each of NEH and RC are divided by the number of the non-zero operations for the job in hand. The modification is supposed to account for the zero processing times in the cell. Each of the heuristics is then, applied separately and independently to each family to generate two job sequences in each family. Then the complete job sequence chromosomes, which are the complete schedules for all jobs are developed, given the families chromosomes developed earlier. Hence, 8 complete job chromosomes are developed, four of them minimize makespan and four minimize total flow time. In the third step, matching and search procedure of the genetic algorithm approach is followed by mixing chromosomes to formulate new sequences (generations) in order to find better schedules.

The genetic algorithm search procedure is applied in the family level for a number of iteration, and then is applied to the job level for a number of iterations. Besides, heuristics are used within each family independently of the other families. Thus, the procedure does not seem to account effectively for the phases’ interaction.

2.4.4 Timetabling In Group Scheduling

Makespan is the commonly used scheduling criterion in GS. However, it was indicated in [26] that in general flow shop scheduling, minimizing makespan was not observed to be a particularly good method for minimizing the total scheduling costs. Sridhar and Rajendran [18] report that the flow time objective is a more significant objective than makespan. In [17] as well, they state that the minimizing total flow time is more relevant objective in the flow line based manufacturing cells.

According to [17] and [30], makespan refers to the schedule completion time (time at which the last job finishes its final operation) while total flow time refers to the sum of jobs' completion times. The minimization of total flow time results in minimum in-process inventory, stable utilization of resources, rapid turnaround of jobs, while minimizing makespan leads to minimizing production run length associated with uneven turnaround of jobs.

The methods for calculating makespan and timetabling for GS are the same as for traditional flow shop problems. Only, the part family membership is considered. Graphical approach uses Gantt chart to represent the schedule, the start and completion times for each job and the start times of the families setups as well as makespan, are read on the horizontal axis.

Hitomi [13] proposed a recursive formulation for timetabling for a number of part families in a flow shop in which makespan is given by:

$$C_{max} = \sum_{i=1}^{F} \left(Q_{(i)M} + \sum_{j=1}^{n_i} D_{(i)(j)k} \right)$$

Where Q_{ik} is the summation of family i setup time on machine k and the processing time of all jobs in it on machine k, given by:

$$Q_{ik} = S_{ik} + \sum_{j=1}^{n_i} P_{ijk} .$$

and

$$D_{(i)(j)k} = \begin{cases} C_{(i)(1)k-1} - C_{(i-1)(n_{i-1})k} - S_{(i)k} > 0 \text{ for } j = 1 \\ C_{(i)(j)k-1} - C_{(i)(j-1)k} > 0 \text{ for } j = 2,3,\ldots,n_i \\ 0, \quad \text{Otherwise} \end{cases}$$

The completion time of job in position j within the family in position i on machine k is calculated by:

$$C_{(i)(j)k} = \sum_{h=1}^{i-1} \left(Q_{(h)k} + \sum_{j=1}^{n_h} D_{(h)(i)k} \right) + S_{(i)k} + \sum_{h=1}^{j} \left(D_{(i)(h)k} + P_{(i)(h)k} \right)$$

Kapov and Vakharia in [12] proposed another recursive formulation for timetabling in a multi-family flow cell. However, in an attempt made by the author to implement this procedure, calculations could not be continued at step 5. In that step the start times for the last jobs in families 2 through F are needed. But these values are calculated later in step 7. Till step 5 only the start times for up to the last job in the first family is known. The procedures, however, is listed below.

(1) The starting time of the first job of the first family in the sequence is equal to The family setup time: $\text{Start}_{1,1,1} = S_{1,1}$.

(2) For i = 1; j =1, k = 2,…,M:

$$\text{Start}_{1,1,k} = \max\{S_{1,k}; \text{Start}_{1,1,k-1}+P_{1,1,k-1}\}$$

(3) For i = 1; j = 2,…,n_1 ; k = 1:

$$\text{Start}_{1,j,1} = \text{Start}_{1,j-1,1} + P_{1,j-1,1}$$

(4) For i = 1; j =2,…,n_1 ; k = 2,…,M:

$$\text{Start}_{1,j,k} = \max\{\text{Start}_{1,j-1,k} + P_{1,j-1,k} ; \text{Start}_{1,j,k-1} + P_{1,j,k-1}\}$$

(5) For i = 2,…,F ; j = 1; k = 1:

$$\mathrm{Start}_{i,1,1} = \mathrm{Start}_{i-1,n_{i-1},1} + P_{i-1,n_{i-1},1} + S_{i,1}$$

(6) For i = 2,…,F ; j = 1 ; k =2,…,M:

$$\mathrm{Start}_{i,1,k} = \max\left\{\mathrm{Start}_{i-1,n_{i-1},k} + P_{i-1,n_{i-1},k} + S_{i,k} ; \mathrm{Start}_{i,1,k-1} + P_{i,1,k-1}\right\}$$

(7) For i = 2,…,F ; j = 2,…,n_i ; k = 1:

$$\mathrm{Start}_{i,j,1} = \mathrm{Start}_{i,j-1,1}+P_{i,j-1,1}$$

(8) For i=2,…,F ; j = 2,…,n_i ; k = 2,…,M:

$$\mathrm{Start}_{i,j,k} = \max\{\mathrm{Sart}_{i,j-1,k} + P_{i,j-1,k} ; \mathrm{Sart}_{i,j,k-1} + P_{i,j,k-1}\}$$

Sridhar and Rajendran [17,18,30] notified that, in manufacturing cells jobs will not have to visit all machines, then processing times for some jobs on some machines are equal to zero, and this is a feature of the manufacturing cell makes it different the general flow shop[1]. Consequently, they reformulated a flow shop recursive timetabling formulation to be used in flow cells by accounting for the possibility of the zero times. Their work considers a single-family cell. The original formulation is found in Sec. 2.3. The modified formulation follows [18].

```
Set C; completion time of job a on the previous machine, equal to zero.
For k = 1 to M Do:
    If $P_{ak} > 0$ then
        Completion time of job a on machine k is:
            $C(\sigma_a , k) = \max \{ C; C(\sigma , k) \} + P_{ak}$
let $C = C(\sigma_a , k)$
    Else
        $C(\sigma_a , k) = C(\sigma , k)$
    End if
The total flow time of jobs in $\sigma_a$ is updated as:
        $F_{\sigma a} = F_{\sigma} + C$
And makespan of the partial schedule $\sigma_a$ is given by
```

---

[1] In 1974, Baker mentioned the possibility of the zero processing times in the general flow shops.

$$C(\sigma_a) = \max \{ C(\sigma) ; C \}$$

Sridhar and Rajendran in [17,30] provide a mathematical example for scheduling a single part family, to show the effect of not considering the zero processing times. They concluded that the consequence of not using a modified recursive procedure would be erroneous results and false engagement of machines with jobs that need no processing on them, and misleading estimations for makespan and total flow time.

Another suggestion in [18,30] is to consider the zero times in the definition of the sequencing indices in the simple GS heuristics. The idea is to divide the index by the number of the non-zero operations for the job. The index is then termed effective sequencing index. They provide no explanation for this suggestion. Besides the modification is used in scheduling jobs within families and can not be used in the family phase if more than one family is scheduled. Called Rajendran' s modification, this suggestion is tested in next chapters.

Rajendran in [30] proposes a heuristic for scheduling in a single-family flow line cell with respect to the bi-criteria of minimizing total flow time and minimizing makespan. The heuristic is developed for the flow shop scheduling and then modified for the flow line cell. The basic concept of the heuristic to is that partial schedule σ_a is preferred to partial schedule σ_b if $\sum_{k=1}^{M} P_{ak} \leq \sum_{k=1}^{M} P_{bk}$ where P_{jk} is the processing time of job j on machine k, and σ_j is the partial schedule after appending job j to it. The relation seeks to identify and append the job with the lower value of the sum of processing times over all machines that is most likely to have the earliest completion time on the last machine.

Another preference relationship used is that σ_a is preferred to σ_b if $\sum_{k=1}^{M} \left[(M - k + 1) P_{ak} \right] \leq \sum_{k=1}^{M} \left[(M - k + 1) P_{bk} \right]$. An initial seed solution is found at the beginning using the traditional NEH heuristic, and the preference relations are then employed in a pair-wise interchange fashion to search for the better sequence. Rajendran then modifies his heuristic for the application in the flow cell as follows. Instead of arranging jobs in the descending order of the summation of the processing times of jobs over the machines (in using NEH), the effective index is computed by dividing the summation by the number of the non-zero-time operations for the job involved. And similarly each sum in the preference relationships is divided by the number of the non-zero processing times for job a or b as appropriate.

In [18] the effective indices were also employed to modify the NEH and the RC methods used in the job phase in the proposed genetic algorithm (See section 2.4.3.4). In the other hand, the heuristic used to generate an initial solution for the SA heuristic in [17] for a single part family, was not modified although the zero processing times were in prospect.

CHAPTER 3

CHAPTER 3

PROPOSED MODIFICATIONS TO THE GROUP SCHEDULING HEURISTICS

Based on the literature review presented in Chapter 2, a number of GS heuristics are selected to study the relative performance of the different types of GS heuristics. Heuristics are classified according to the amount of calculations involved, into single-pass, and multiple-pass methods [19,21]. A third class is to be considered to include the iterative improvement techniques. Two aspects are given more interest: the presence of the zero processing times, and the two phase's nature of the GS model. A number of modifications are proposed to improve the performance and investigate the capabilities of the heuristics with respect to the features of the GS model. A procedure for timetabling in a multi-family flow cell considering the zero processing times is proposed as well. The main assumptions apply to the study are:

1. A static flow line cell is composed of M machines and is dedicated to the manufacture of N jobs classified into F part families is considered;
2. Cell and part families have been identified satisfactorily;
3. Machines are continuously available;
4. Only permutation schedules are considered;
5. Minor setup times are included in job processing times;
6. Family setup times are not sequence dependent;
7. No preemption is allowed and no relative priorities among jobs.

The selected methods and the proposed modifications are compared to each other with respect to makespan and total flow time separately. Heuristics are classified and the proposed modifications are listed below.

1. **The single-pass heuristics**. Hitomi's heuristic [13] is studied. It generates a single schedule. Hitomi is the GS version of Petrov's method, which is the direct

extension of Johnson's efficient rule to the general flow shop problem. It does not take into account the phase's interaction.

2. The multiple-pass heuristics. The two best performing simple flow shop scheduling heuristics, as modified for GS, are studied. These are the CDS and the NEH methods. The modified versions are presented in [15,23]. Neither of them can account for phase's interaction of GS.

3. The iterative improvement techniques. The simulated annealing (SA) [15] and the tabu search (TS) [12], heuristic approaches are studied. The iterative methods iterate between the two scheduling phases so that phase's interaction is dealt with.

3.1 THE SINGLE-PASS METHODS

3.1.1 HIT-M

This is a modified version of Hitomi's method described in Sec. 2.4.2.1. The modification adopts Rajendran's modification suggested in [17,18,30] (See Sec. 2.4.4) that is supposed to account for the zero processing times in the structure of the heuristics. Step 1 in Phase 1 in the original Hitomi's method in Sec 2.4.2.1 will be as follows.

Step 1. For each job j in family i, calculate the sequencing indices:

$$A_{ij} = \frac{\sum_{k=1}^{h} P_{ijk}}{\text{No. of non-zero operations of job j on the h machines}}$$

$$B_{ij} = \frac{\sum_{k=h'}^{M} P_{ijk}}{\text{No. of non-zero operations of job j on the remaining machines}}$$

where $h = M/2, h' = h + 1$ for even M , and $h = h' = (M+1)/2$ for odd M

3.2 THE MULTIPLE PASS METHODS

Three new versions of CDS (Sec. 2.4.2.2) and one of NEH (Sec. 2.4.2.3) are proposed. In the listings below, "Flow" denotes the total flow time. "Make" is to replace "Flow" when using the heuristics for minimizing makespan.

3.2.1 CDS-M-1

This is the original CDS employing Rajendran's modification. The two sequencing indices in Step 5 in Phase 2 of the original CDS (Sec.2.4.2.2) are to be divided by the number of the non-zero operations for the job in hand.

3.2.2 CDS-M- 2

This is an iterative CDS heuristic. It returns to the family phase after completing job sequencing to search for a better schedule given the jobs sequence within each family. If the families sequence could be changed, the heuristic returns to the job phase given the new families sequence, and so on. CDS-M-2 is supposed to be able to take phases' interaction in consideration.

Phase 1: Families sequencing

Step 1. Set x = 1, where x = 1,2,…, M-1. Let $\text{Flow}^{o} = \infty$.

Step 2. If x ≥ M then switch to phase 2, else go to Step 3.

Step 3. Calculate for each family i the two artificial processing times:

$$A_i^x = \sum_{k=1}^{x} \left[S_{ik} + \sum_{j=1}^{n_i} P_{ijk} \right] \quad , \quad B_i^x = \sum_{k=M-x+1}^{M} \left[S_{ik} + \sum_{j=1}^{n_i} P_{ijk} \right]$$

Step 4. Apply Johnson's rule and generate a families sequence.

Let the total flow time be Flowx

Step 5. If Flow $^{x-1}$ > Flowx then let Flow^x = Flowx, and keep the sequence.

Else let Flow^x = Flow $^{x-1}$

Step 6. Set x = x + 1 , and go to Step 2

Phase 2: Jobs sequencing within each family

Step 1. Set i = 1.

Step 2. If i $\geq$ F + 1 then switch to Phase 3, else proceed.

Step 3. Set x = 1 and Flow^o = ∞.

Step 4. If x $\geq$ M then go to Step 8, else go to Step 5

Step 5. Calculate for each job j in family i the two artificial processing times

$$A_{ij}^x = \sum_{k=1}^{x} P_{ijk} \quad \text{and} \quad B_{ij}^x = \sum_{k=M-x+1}^{M} P_{ijk}$$

Step 6. Apply Johnson's rule and find a jobs sequence.

Let total flow time be Flowx.

Step 7. If Flow^{x-1} > Flowx then let Flow^x = Flowx and keep the sequence;

Else let Flow^x = Flow^{x-1}. Set x = x + 1, and go to Step 4

Step 8. Set i = i + 1 and go to Step 2

Phase 3: Families resequencing

Step 1. Keep the complete schedule found in Phase 2, if coming from Phase 2, (or in Phase 4 if coming from Phase 4).

Let the total flow time be Flow. Set x = 1 and let Flow^o = Flow.

Step 2. IF x ≥ M and a change in the complete schedule occurred in Phase 3 then switch to Phase 4, Else if x ≥ M and no change occurred in Phase 3 then stop, Else proceed.

Step 3, 4,5, and **6** are the same as in phase 1.

Phase 4: Jobs resequencing within each family

Step 1. Keep the complete schedule found in Phase 3. Let total flow time be Flow. Set i = 1.

Step 2. If i ≥ F+1 and a change in the complete schedule occurred in Phase 4 then switch to phase 3, else If i ≥ F+1 and no change occurred in Phase 4 then stop, else go to step 3.

Step 3. Set x = 1, and let Flow^{o} = Flow.

Step 4, 5, 6, 7, and **8** are the same as in Phase 2.

3.2.3 CDS-M-3

This is CDS-M-2 employing Rajendran’s modification. The scheduling indices in Phases 2 and 4 in CDS-M-2 are divided by the number of the non-zero operations for the job in hand.

In the implementation of the CDS, families in each family phase are treated as jobs, by computing the artificial processing time for each family i on each machine k as $P_{ik} = \sum_{j=1}^{n_i} P_{ijk} + S_{ik}$. The families sequence is maintained during job phases, and the index i will denote the position of the family in the sequence during the job phase. While working within the ith family, the rest of families (i+1, i+2,…, F) are empty that is there is no complete schedule until the end of the heuristic. This approach does not apply to Phases 3 and 4 in CDS-M-1 and CDS-M-3, in which there are complete schedules at the beginning of them.

3.2.4 NEH-M

This is the original NEH [15] employing Rajendran's modification in the scheduling indices. In Phase 2 of NEH (Sec. 2.4.2.3), Step 3 becomes as follows.

Step 3. Compute for each job j in family i:

$$T_{ij} = \frac{\sum_{k=1}^{M} P_{ijk}}{\text{number of non - zero operations for job j}}$$

As done with CDS, in implementing NEH the families in Phase 1 are treated as jobs, by computing the processing time for each family i on each machine k as $P_{ik} = \sum_{j=1}^{n_i} P_{ijk} + S_{ik}$. In phase 2, index i denotes the position of the family in the sequence. During job phase, in sequencing within family i, the rest of families (i+1, i+2,…, F) are empty.

3.3 THE ITERATIVE IMPROVEMENT TECHNIQUES

Three new versions of SA and two of TS are proposed. Two initial solutions will be used for each method; a random initial solution and a relatively good initial solution generated by the original Hitomi's heuristic. Heuristics are listed in next subsections with respect to total flow time and for the case of using a random initial solution,

The GP parameter in SA method controls the amount of search efforts given to each scheduling phase. Its value of 0.1 in [15] leads to spending 10% of efforts to the family phase and 90% to the job phase. Alison [19] stated that the family phase in GS is more worthy. To investigate this, GP will be given values of 0.1, 0.3, 0.5, 0.7, and 0.9. This is used for all the SA versions.

3.3.1 SA-M-1

In this version, slight modifications in Steps 6 and 7 in the original SA are suggested hoping to increase the efficiency of the search process. The idea is to prevent reversing (cancellation) of the last performed perturbation during the current perturbation operation, hence to avoid wasting efforts and time.

A sample execution of the original SA showed that out of 1250 search operations (50 searches per iteration for 25 iterations), 125 families perturbations and 1125 jobs perturbations are performed (GP = 0.1). Of the 125 trials on families, an average of 63 (51%), random numbers were repeated successively in Step 6. Assume a trial families sequence as 1,4,5,3,2. If in Step 6, $\nu 1$ is 3 then the sequence is perturbed to be 1,4,3,5,2. If in the succeeding search, $\nu 1$ is 3 again, then the third and fourth families are interchanged and the sequence will be back to 1,4,5,3,2. Hence the first perturbation operation was reversed and wasted. That is nearly 25% of the perturbation operations in Step 6 were canceled. Similarly, about 7% of job perturbations are wasted.

The significance of the suggested modification is to avoid wasting efforts and iterations and hence to enlarge the search area. In the family phase, as before, about 25% enlargement is possible. In other words about 25% more chance to find the best schedule is made available. Main parts in Steps 6 and 7 in the original SA heuristic (Sec.2.4.3.1) are modified to become as follows:

Step 6. ① In carrying out this step the order of jobs within each family will not change. Generate a random number $\nu 1$ ($1 \le \nu 1 \le F$).

If $\nu 1$ = last $\nu 1$ and the previous perturbation was a families interchange, then go back to ①. **Else** **If** ν1 = last ν1 and the previous perturbation was a jobs interchange, and no change in the current sequence has occurred in that perturbation, then go back to ①. **Else** proceed as the original Step 6 in SA.

Step 7. ① In carrying out this step the sequence of part families stays the same. Generate a random number ν3 ($1 \leq \nu3 \leq N$).

If ν3 = last ν3 and the previous perturbation was a jobs interchange, then go back to ① **Else** **If** ν3 = last ν3 and the previous perturbation was a families interchange, and no change in the current sequence has occurred in that perturbation, then go back to ①. **Else** proceed as the original Step 7 in SA.

3.3.2 SA-M-2

In this version of SA, a change dependent acceptance probability is employed. The standard acceptance probability for the SA approach as reported in [33] (See Sec. 2.4.3.1) is used. This form is closer to the generic SA than the SA heuristic proposed in [15]. Accordingly, the parameter X will be the maximum temperature, which is to be reduced by a temperature reduction factor r at each step. The value for r is 0.9. Freezing temperature is 1.62 so that using X = 25 and r = 0.9, there will be 25 temperature steps corresponding to the 25 iterations in the original SA.

The acceptance probability will be calculated at each temperature step. The remaining variables including the initial value of the acceptance probability will take the same values as in the original SA. SA-M-2 is described as follows.

Step 1. Set X, Y, AP_o, and GP. Let r = 0.9.

Step 2. Generate a random initial schedule. This includes a complete sequence for all jobs (Ω^o), a family sequence (τ) and a sequence for jobs within each part family (μ_f); where f = 1, 2 , ... , F. Let this be the current solution with a total flow time $Flow^o$. Let Ω^* represent the incumbent solution with total flow time $Flow^*$, and set $\Omega^* = \Omega^o$ and $Flow^* = Flow^o$.

Step 3. Let X = rX. If $X \leq 1.62$ then stop, else set y = 0 and continue.

Step 4. Set y = y + 1. If y > Y then go to Step 3, else go to Step 5.

Step 5. Generate a random number ν ($0 \leq \nu \leq 1$).

If $\nu \geq GP$, then go to Step 7, else go to Step 6

Step 6. In carrying out this step, the sequence of jobs within each family will not change. Generate a random number $\nu 1$ ($1 \leq \nu 1 \leq F$). Interchange the family in position $\nu 1$ with that in position $\nu 1+1$ (if $\nu 1 = F$, then interchange the family in position F with that in position 1) and generate a family sequence τ^1. Based on τ^1 specify a new complete job sequence Ω^1 and calculate its total flow time $Flow^1$.

(a) If $Flow^1 > Flow^*$ then go to (b)

Else let $\Omega^* = \Omega^1$, set $Flow^* = Flow^1$ and go to (b).

(b) If $Flow^1 > Flow^o$ then let $\Delta = Flow^1 - Flow^o$, calculate the acceptance probability $AP_x = EXP(- \Delta / X)$ and go to (c). Else let $\tau = \tau^1$ and $\Omega^o = \Omega^1$ in the current solution, and set $Flow^o = Flow^1$ and go to Step 4.

(c) Generate a random number $\nu 2$ ($0 \leq \nu 2 \leq 1$). If $\nu 2 \geq AP_x$ then go to Step 4, else let $\Omega^o = \Omega^1$, $\tau = \tau^1$ in the current solution, set $Flow^o = Flow^1$ and go to step 4.

Step 7. In carrying out this step, the sequence of families is not changed. Generate a random number $\nu 3$($1 \leq \nu 3 \leq N$), where N is the total number of jobs. Let f_1 be the family in which job $\nu 3$ is included. Interchange job in position $\nu 3$ with that in position $\nu 1+1$ (if $\nu 3$ is the last in the family f_1, interchange job in positions $\nu 3$ with that in

Position 1 family f_1) in Ω^o. Let the new sequence be $\mu^1_{f_1}$ for family f_1 and the new complete sequence be Ω^1 with total flow time $Flow^1$.

(a) If $Flow^1 \geq Flow^*$ then go to (b), else let $\Omega^* = \Omega^1$ in the incumbent solution, set $Flow^* = Flow^1$ and go to (b).

(b) If $Flow^1 \geq Flow^o$ then let $\Delta = Flow^1 - Flow^o$, and calculate the acceptance probability $AP_x = EXP(- \Delta / X)$ and go to (c).
Else let $\mu_{f_1} = \mu^1_{f_1}$,$\Omega^o = \Omega^1$ in the current solution, and set $Flow^o = Flow^1$ and go to Step 4.

(c) Generate a random number $\nu 2$ ($0 \leq \nu 2 \leq 1$). If $\nu 2 \geq AP_x$ then go to Step 4, else let $\mu_{f_1} = \mu^1_{f_1}$ and $\Omega^o = \Omega^1$ in the current solution, and set $Flow^o = Flow^1$ and go to Step 4.

3.3.3 SA-M-3

In this version, the control made on the behaviour of the random numbers in SA-M-1 is added to SA-M-2. Main parts of Steps 6 and 7 in SA-M-2 are modified similar to Steps 6 and 7 in SA-M-1.

3.3.4 TS-M-1

In this version, when generating the new restart schedule, the current jobs sequences within families are kept instead of being randomly regenerated. The concept is to make use of the search efforts in the job phase. LTM is used for the same purpose in the family phase. Step 2 in the original TS (Sec2.4.2.2) is modified to be as follows.

Step 2. **If** LTM = **0** then generate a random families sequence,
Else generate a families sequence using LTM_MAX and keep the current jobs sequence within each family.

Let this be the current solution Ω^o with a total flow times $Flow^o$.
Let Ω^* represent the incumbent solution with total flow time $Flow^*$.
Set $\Omega^* = \Omega^o$, $Flow^* = Flow^o$ and LTM = LTM + 1.

3.3.5 TS-M-2

In this version, LTMs for jobs within each family are developed and used to generate new restart jobs sequences within families based on these LTMs as made in the family phase. Actually, the original TS is completed rather than being modified.

For jobs in each family i (i = 1,2 …, F) a LTM termed LTM_i. LTM_i is a frequency matrix of the size n_i x n_i will contain information about the number of times a job occupied a certain position in the trial solutions. Step 2 in the original TS will be as follows below. Proposed LTMs will be used based on the maximal frequenceies as made with the families LTM.

Step 2. **If** LTM = **0**, generate a random families sequence, **Else** generate a families sequence using LTM_MAX, and for each family, generate a jobs sequence using LTM_MAX_i .Let this be a current solution Ω^o with a total flow times $Flow^o$. Let Ω^* represent the incumbent solution with total flow time $Flow^*$. Set $\Omega^* = \Omega^o$, $Flow^* = Flow^o$.
Set LTM = LTM + 1 and $LTM_i = LTM_i + 1$ for all i.

3.4 PROPOSED TIMETABLING PROCEDURE

For the case of a cell dedicated for processing of a number of part-families, the following timetabling procedure is proposed. It can compensate for the presence of the zero processing times. A formulation of the procedure disregarding the zero times is first presented, then the procedure with the consideration of the zero times is presented. For both formulations let $\text{Start}_{i,j,k}$ be the start time for job j in family i on machine k, and $\text{Setstart}_{i,k}$ the family setup time start for family i on machine k.

3.4.1 Proposed Timetabling Procedure Disregarding the Zero Times

1st. **For i = 1, 2,..., F , For j = 1, For k = 1, 2,..., M**

$$\text{Start}_{i,1,k} = \max \begin{cases} \text{Start}_{i,1,k-1} + P_{i,1,k-1} \\ \text{Start}_{i-1,n_{i-1},k} + P_{i-1,n_{i-1},k} + S_{i,k} \end{cases}$$

$$\text{Setstart}_{i,k} = \text{Start}_{i,j,k} - S_{i,k}$$

For j = 2,3,…,n_i , For k = 1,…,M

$$\text{Start}_{i,j,k} = \max \begin{cases} \text{Start}_{i,j-1,k} + P_{i,j-1,k} \\ \text{Start}_{i,j,k-1} + P_{i,j,k-1} \end{cases}$$

$$\text{Setstart}_{i,k} = \text{Start}_{i,jjjj,k} - S_{i,k}$$

$$\text{Makespan} = \text{Start}_{F,n_F,M} + P_{F,n_F,M}$$

$$\text{Total Flow Time} = \sum_{i=1}^{F} \sum_{j=1}^{n_i} \text{Finish}_{i,j,M}$$

3.4.2 Proposed Timetabling Procedure Considering the Zero Times

2nd. **For i = 1, 2,..., F , For j = 1, For k = 1, 2, ... , M**

$$\text{Start}_{i,1,k} = \begin{cases} \max \begin{cases} \text{Start}_{i,1,kk} + P_{i,1,kk} \\ \text{Start}_{ii,jj,k} + P_{ii,jj,k} + S_{i,k} \times Z_1 \end{cases} & \text{If} \quad P_{i,1,k} > 0 \\ 0 & \text{Otherwise} \end{cases}$$

Where : kk Last machine that job 1 in family i ($J_{i,1}$) was processed on.

jj The job preceeds job $J_{i,1}$ on machine k.

ii Family containing job jj.

Z_1 Binary variable such that $= \begin{cases} 1 & \text{If } P_{i,1,k} > 0 \\ 0 & \text{If } P_{i,1,k} = 0 \end{cases}$

For j = 2,3,…,n_i , For k = 1,…,M

$$\text{Start}_{i,j,k} = \begin{cases} \max \begin{cases} \text{Start}_{i,j,kkk} + P_{i,j,kkk} \\ \text{Start}_{iii,jjj,k} + P_{iii,jjj,k} + S_{i,k} \times Z_2 \end{cases} & \text{If } P_{ijk} > 0 \\ 0 & \text{Otherwise} \end{cases}$$

$$\text{Setstart}_{i,k} = \text{Start}_{i,jjjj,k} - S_{i,k}$$

Where : kkk Last machine job j in family i ($J_{i,j}$), visited.

jjj The job preceeds job $J_{i,j}$ on machine k.

iii Family containing job jjj.

jjjj First job in family i having a non-zero time on machine k.

Z_2 Binary variable such that $= \begin{cases} 1 & \text{If iii} < \text{i} \\ 0 & \text{otherwise} \end{cases}$

$$\text{Makespan} = \max_{k=1,2,\ldots,M} \left\{ \text{Start}_{v,t,k} + P_{v,t,k} \right\}$$

$$\text{Total Flow Time} = \sum_{i=1}^{F} \sum_{j=1}^{n_i} \text{Finish}_{i,j,lk}$$

Where : t Last job processed on machine k

v Family containing job t

lk Last machine in the cell that job J_{ij} was processed on.

3.4.3 Consequences of Not-Considering the Zero Times

To show the effects of not taking the possibility of the zero times in consideration during timetabling in multi-family cells, the following GS sample problem is presented. A 3-families, 4-machines and 5-jobs per family GS problem is solved twice employing the proposed timetabling procedure in its two formulations as given in Secs. 3.4.1 and 3.4.2 respectively. Data for the problem (shown in Table 3.1) is generated as explained in next section, such that 20% of jobs need no processing on some machines (zero processing times).

Table 3.1 Basic data for the 3-family, 4-machines, and 5-jobs per family, GS example

Family	Job	Machine 1	Machine 2	Machine 3	Machine 4
F_1	S_{1k}	7	7	17	8
	J_{11k}	4	7	0	0
	J_{12k}	0	3	6	5
	J_{13k}	6	4	0	5
	J_{14k}	0	2	4	10
	J_{15k}	0	5	6	6
F_2	S_{2k}	3	12	2	11
	J_{21k}	5	4	10	0
	J_{22k}	7	0	10	1
	J_{23k}	9	9	0	5
	J_{24k}	0	9	0	0
	J_{25k}	5	9	0	8
F_3	S_{3k}	6	15	9	16
	J_{31k}	2	0	9	3
	J_{32k}	4	9	10	1
	J_{33k}	3	3	6	10
	J_{34k}	0	6	4	0
	J_{35k}	4	3	0	1

A feasible schedule for this GS problem is generated by the original Hitomi's heuristic method as **F_1**(J_{14}, J_{12}, J_{15}, J_{13}, J_{11}), **F_3** (J_{31}, J_{33}, J_{32}, J_{34}, J_{35}), **F_2** (J_{22}, J_{21}, J_{25}, J_{23} J_{24}). The time tables for the schedule is shown in Table 3.2 for disregarding the zero processing times, and in Table 3.3 when considering the zero processing times in the calculations. Gantt charts for the two cases are shown in Fig.3.1.

The shaded cells in Table 3.3 contain the start and finish times that are different from the corresponding values in Table 3.2. These differences are due to the compensation for the zero times. The shaded numbers are the correct values that are obtained by eliminating the zero time-jobs from their locations in the schedule. Basically, the effect of the proposed timetabling procedure is to eliminate the zero time-jobs. In Fig.3.1 the numbers above the horizontal bars indicate the locations of the zero-jobs. These are shown in part [a]. In part [b] these jobs are eliminated. Comparing the two tables and the two parts [a] and [b] in Fig.3.1, the following observations are true.

Table 3.2 Time table for the data of Table 3.1 without considering zero times

Family	Job	Machine 1		Machine 2		Machine 3		Machine 4	
		Start	Finish	Start	Finish	Start	Finish	Start	Finish
F_1	S_{1K}	0	7	0	7	0	17	13	21
	J_{14K}	7	7	7	9	17	21	21	31
	J_{12K}	7	7	9	12	21	27	31	36
	J_{15K}	7	7	12	17	27	33	36	42
	J_{13K}	7	13	17	21	33	33	42	47
	J_{11K}	13	17	21	28	33	33	47	47
F_3	S_{3K}	17	23	28	43	34	43	47	63
	J_{31K}	23	25	43	43	43	52	63	66
	J_{33K}	25	28	43	46	52	58	66	76
	J_{32K}	28	32	46	55	58	68	76	77
	J_{34K}	32	32	55	61	68	72	77	77
	J_{35K}	32	36	61	64	72	72	77	78
F_2	S_{2K}	36	39	64	76	74	76	78	89
	J_{22K}	39	46	76	76	76	86	89	90
	J_{21K}	46	51	76	80	86	96	96	96
	J_{25K}	51	56	80	89	96	96	96	104
	J_{23K}	56	65	89	98	98	98	104	109
	J_{24K}	65	65	98	107	107	107	109	109

1. Job J31 is reported in Table 3.2 and Fig.3.1 [a] to finish on machine 2 at time 43 hence it can start on machine 3 only at time 43. As this job is the first in family 3 then setup time for the family on machine 3; S33 will start at time 34 to finish at 43. Machine 3 is then idle for one time unit after finishing J13 and J11 according to the schedule, from 33 to 34. But this job J31 does not need processing on machine 2 (P312 = 0), thus this finishing time of 43 for J31 is meaningless. Hence J31 does not have to wait until 43 to start on machine 3. Instead, J31 can be removed from the schedule at this location and hence it can be shifted to start at 42 on machine 3 while S33 will start at 33 to finish at 42, as shown in Table 3.3 and in Fig.3.1 [b].

2. Job J22 starts on machine 3 at time 76 after being finished on machine 2 at time 76, to finish at 86 as reported in Table 3.2 and Fig.3.1 [a]. But P222 = 0. Hence J22 can start on machine 3 once the machine is free that happens at time 73. Then the job is finished at time 83 not at time 86, which is shown in Table 3.3 and Fig.3.1 [b]. Idle time on machine 3 before this job is removed as well.

Table 3.3 Time table for the data of Table 3.1 considering zero processing times

Family	Job	Machine 1		Machine 2		Machine 3		Machine 4	
		Start	Finish	Start	Finish	Start	Finish	Start	Finish
F_1	S_{1K}	0	7	0	7	0	17	13	21
	J_{14K}	0	0	7	9	17	21	21	31
	J_{12K}	0	0	9	12	21	27	31	36
	J_{15K}	0	0	12	17	27	33	36	42
	J_{13K}	7	13	17	21	0	0	42	47
	J_{11K}	13	17	21	28	0	0	0	0
F_3	S_{3K}	17	23	28	43	33	42	47	63
	J_{31K}	23	25	0	0	42	51	63	66
	J_{33K}	25	28	43	46	51	57	66	76
	J_{32K}	28	32	46	55	57	67	76	77
	J_{34K}	0	0	55	61	67	71	0	0
	J_{35K}	32	36	61	64	0	0	77	78
F_2	S_{2K}	36	39	64	76	71	73	78	89
	J_{22K}	39	46	0	0	73	83	89	90
	J_{21K}	46	51	76	80	83	93	0	0
	J_{25K}	51	56	80	89	0	0	90	98
	J_{23K}	56	65	89	98	0	0	98	103
	J_{24K}	0	0	98	107	0	0	0	0

3. Job J25 starts on machine 4 at 96 and finishes at 104. This is because it is to be finished on machine 3 at 96 as shown in Table 3.2 and Fig.3.1 [a], although it is not processed on that machine. It is clear that J25 and also J23 can be started on machine 4 at 90 and 98 respectively. This is shown in Table 3.2 and Fig.3.1 [a]. In addition, machine 4 is released at 103 instead of 109 hence freeing more idle time.

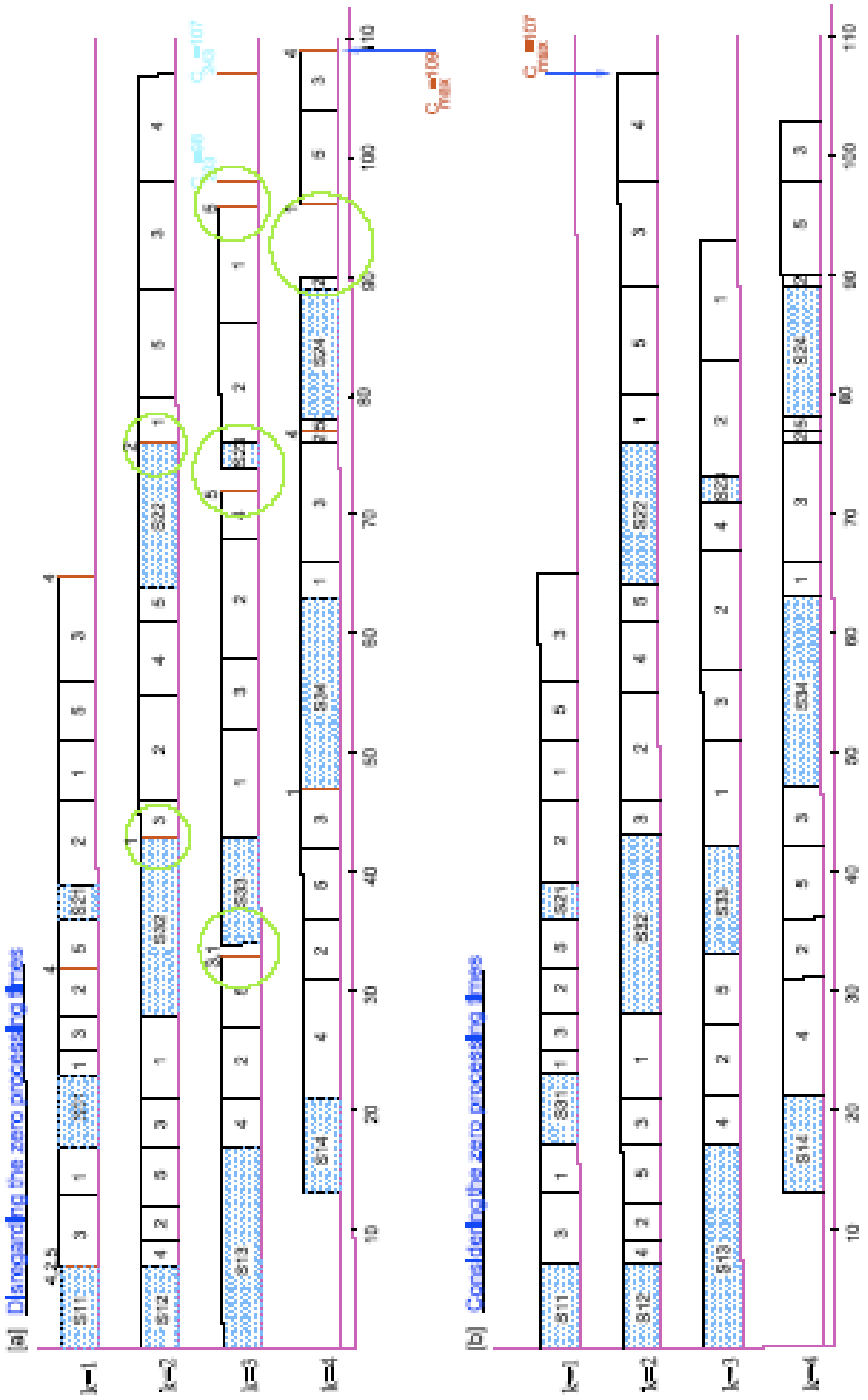


Fig.3.1 Gantt charts for sample problem of the data in Table 3.1

4. Machine 3 is reported in Table 3.2 and Fig.3.1 [a] to be busy with J24 until time 107 which is not true since P243 = 0. The same can be said for J233 that needs no processing on machine 4 although that machine is falsely reported in Table 3.2 to be engaged with J233 until time 98.

5. Makespan in Fig.3.1 [a] is the finish time of J24 (109) on machine 4. In Fig.3.1 [b] makespan is the finish time of J24 (107) on machine 2. The total flow time when disregarding the zero times is calculated as 1085. When considering the zero times and making these corrections to jobs start and completion times, it is corrected to be 1043.

Consequently, it can be said that neglecting the possibility of the zero times in multi-family manufacturing cells will result in erroneous jobs completion times, false and overestimated values for makespan and total flow time, and misleading information about machine utilization and availability. The correct information is obtained when taking the zero processing times in account during the timetabling calculations. Start and finish times for the zero-times jobs should be set to zero so as not to affect the start times of the following jobs in the schedule. Besides, machine idle times can be reduced as well.

Another consequence of using the proposed modified time tabling procedure is that makespan is found to occur on any machine not necessarily on the last machine and not necessarily with the last job in the schedule. Makespan when zero processing times exist, is not always correct to be defined as the time span from the start of the first job on the first machine to the completion of the last job on the last machine [3,17]. Instead it is sufficient to be defined as the largest completion time. Given that the start and completion times for the zero time jobs are set to zero, then makespan can be defined mathematically as in Sec.2.1.4.

The consequences of not taking the zero processing times in consideration are true regardless of the level of performance of the scheduling methodology employed. Solving the same sample 3x4x5 problem using the proposed TS-M-1 for both the total flow time and makespan objectives, the same observations were found true. The corrections to total flow time and makespan are found as shown in Table 3.4.

Table 3.4 Corrections made to total flow time and makespan by the Proposed timetabling procedure considering the zero times in a 3x4x5 problem

Method	**Disregarding The zero times**	**Considering the zero times**	**Change (%)**
Hitomi			
Total flow time	1085	1043	3.87
Makespan	109	107	1.35
TS-M-1 for total flow time			
Total flow time	1049	1010	3.72
Makespan	109	107	1.35
TS-M-1 for makespan			
Total flow time	1051	1024	2.57
Makespan	101	107	0

For a larger size problem (5x5x5), the same observations are true as well. For the same heuristics shown in Table 3.4, the corrections to makespan and total flow time are found as shown in Table 3.5.

Table 3.5 Corrections made to total flow time and makespan by the Proposed timetabling procedure considering the zero times in 5x5x5 problem

Method	**Disregarding the zero times**	**Considering the zero times**	**Change (%)**
Hitomi			
Total flow time	2876	2605	9.42
Makespan	195	194	0.51

TS-M-1 for total flow time			
Total flow time	2649	2301	13.14
Makespan	195	193	1.03
TS-M-1 for makespan			
Total flow time	2772	2389	13.82
Makespan	186	181	2.69

3.5 COMPARISON OF THE GROUP SCHEDULING HEURISTICS

For Carrying out the comparison among the described GS heuristics, GS problems of various sizes are randomly generated. Data configuration is similar to that used in [12,15]. 30 problems for each of 8 problem sizes were generated as described below. Written as ($F \times M \times n_i$) the generated problem sizes are (3×3×3), (3×4×5), (4×4×4), (6×5×4), (5×5×5), (6×6×6), (5×6×8), and (8×8×8).

Processing times for jobs are integer random variables uniformly distributed in U(1, 10). Most researchers have used this distribution in their experimentation [30]. To generate the zero processing times a uniform random number is sampled, if it is less than or equal to 0.2 a zero processing time is used. Hence 20% of job processing times in the cell are set to zero. This percentage is used in [17,18,30]. Nevertheless, a family can not be empty.

For each problem size the family setup times are integer random variables uniformly distributed in the following three sets: U(1,20), U(1,50) and U(1,100) so that to study the impact of the different values for the family setup time to job processing time ratios; S/R of 2, 5 and 10 respectively.

To compare the relative performance of the heuristics, a measure of performance is established as follows. The total flow time and makespan obtained by the original Hitomi's heuristic for each S/R ratio are standardized to be 100%. Then the average total flow time or the average makespan for the other heuristics are related to that of Hitomi. For instance, let F_{Hitmoi} and F_X represent the average total flow times obtained by Hitomi and the X heuristic, respectively, then the relative total flow time for X is denoted by RELF_x (relative makespan is RELM_X) and is given by:

$$\mathrm{RELF}_X = \left(\frac{F_X}{F_{\mathrm{Hitomi\ -1}}} \right) \times 100$$

Hence, a value below 100 will indicate that X outperforms Hitomi and is preferred to it. And generally lower values are for better performance. In addition, for each scheduling criterion the other criterion is recorded as a side result for the comparison. The computational times in seconds are recorded as well. Results of the comparison are discussed in Chapter 4.

CHAPTER 4

CHAPTER 4

ANALYSIS AND DISCUSSION OF RESULTS

In this chapter the results of the comparison of the GS heuristics described in Chapter 3 are analyzed and discussed. All procedures were coded in Quick BASIC 4.5. The computational experiments were performed on a 100 MHz Pentium IBM compatible personal computer. The complete set of results is tabulated in Appendix A for total flow time and Appendix B for makespan.

For convenience, the iterative improvement methods in this chapter and in the tables of results will be sometimes referred to as explained in Appendix A.

4.1 RESULTS WITH RESPECT TO TOTAL FLOW TIME

4.1.1 The Single and Multi-Pass Methods

Basically the modifications to the simple (single and multi-pass) methods are concerned with testing Rajendran's modification (Sec. 2.4.4) [18,30]. As shown in Fig.4.1, Rajendran's modification is generally ineffective for all the simple methods. This is shown in Fig.4.1 for the 5x5x5 problem size as an example. It can be found from the tables in Appendix A that this is true for all problem sizes.

It is shown in Fig. 4.1 also that the proposed CDS-M-2 is the best CDS version. This is the result of its iterative behaviour that can handle the scheduling phases' interaction. But this is limited by the finite number of solutions generated by CDS and is incurring longer CPU times. For example, for the largest problem at S/R = 10, the original CDS takes 2.95 sec while CDS-M-2 takes 6.84 sec. This

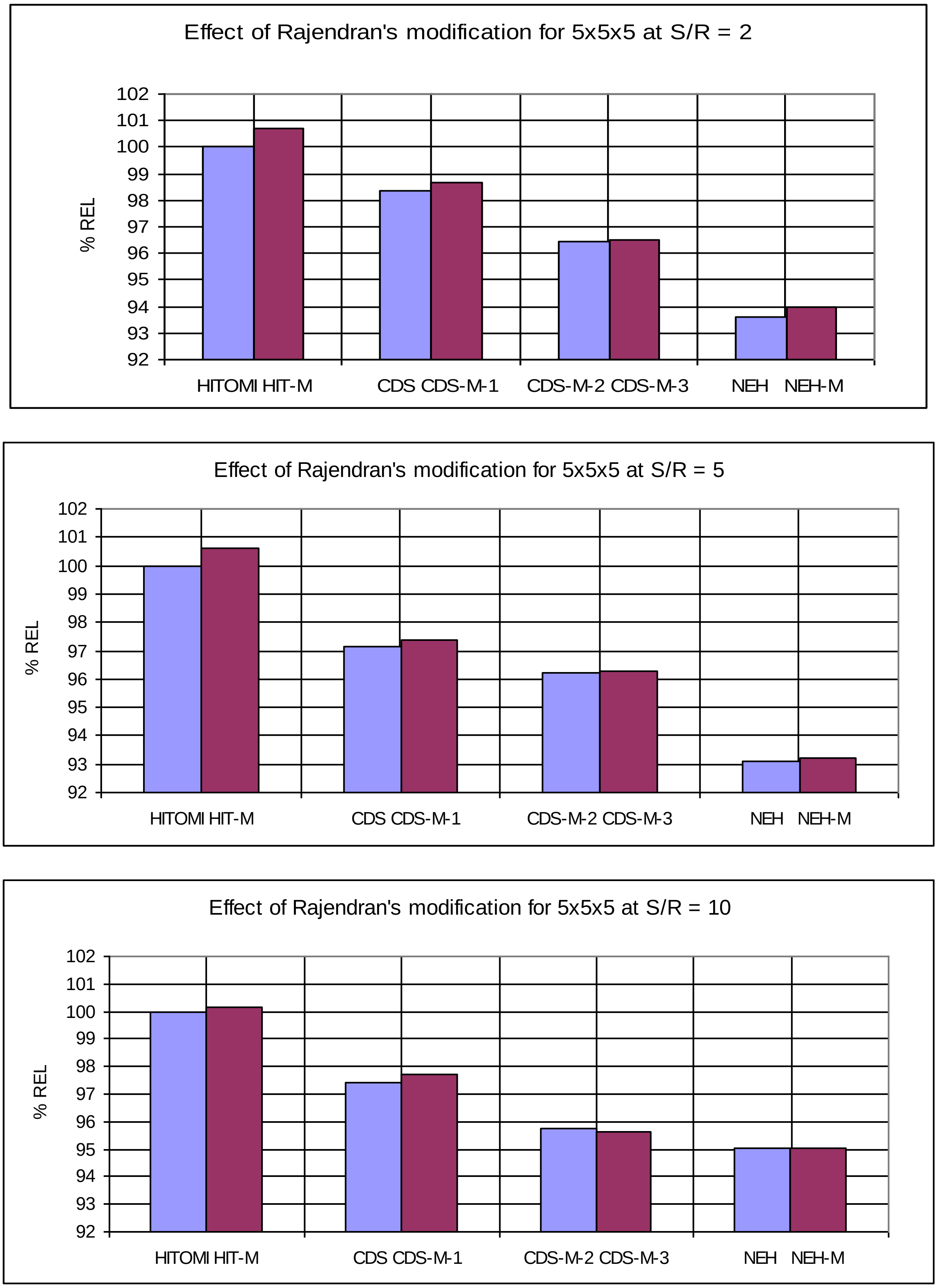


Fig.4.1 The effect of Rajendran's modification on the simple heuristics

🟦 Without Rajendran 🟪 With Rajendran

result indicates the importance of taking phases' interaction in consideration. It is thus logical to consider the development and use of the iterative improvement techniques for the GS problems.

Fig.4.1 also shows that NEH is the best performing among all the simple methods, while Hitomi shows the least performance. This is true for all problem sizes and all S/R values.

Comparing Figs.4.2 and 4.3, it can be observed that NEH is better than CDS-M-2 for all conditions. As problem size increases, the performance of NEH

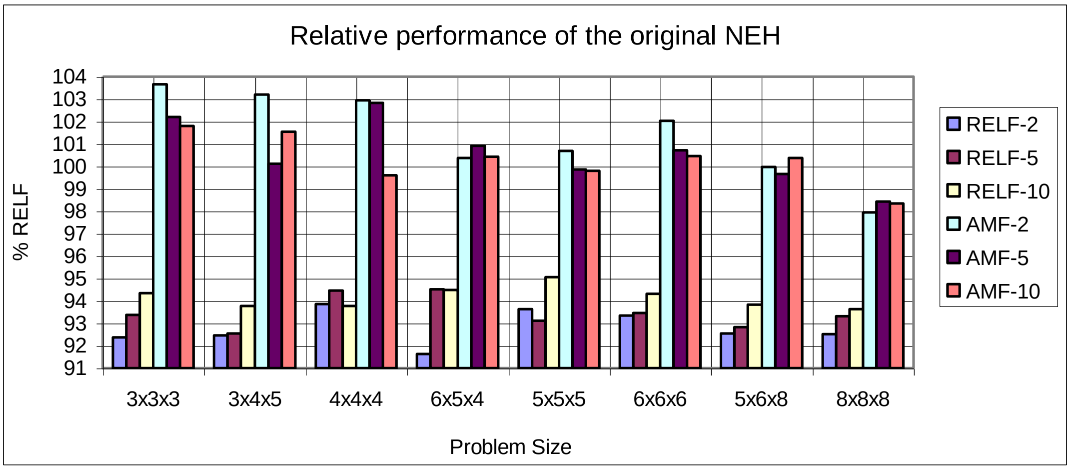


Fig.4.2 Level of performance of the original NEH – Total flow time

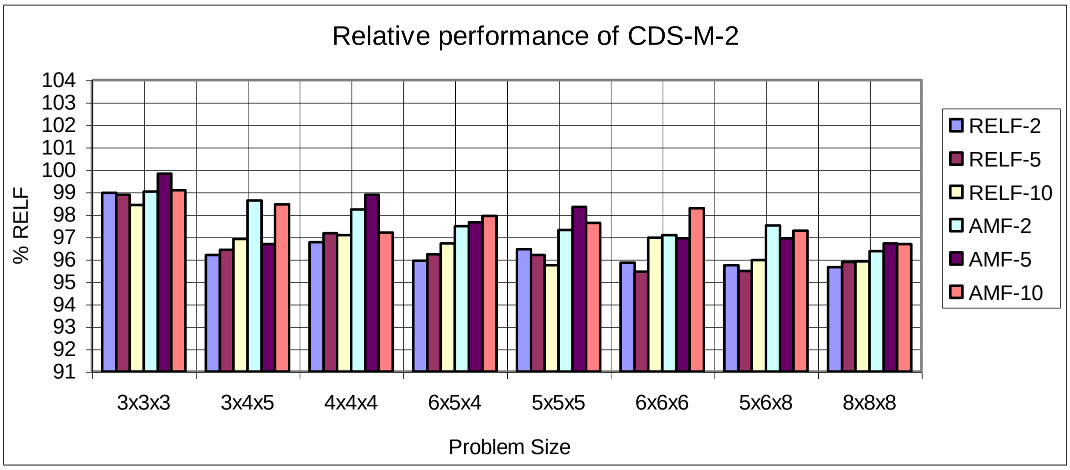


Fig.4.3 Level of performance of CDS-M-2 – Total flow time

fluctuates about a horizontal trend while CDS-M-2 slightly improves. NEH performs better at the smaller S/R values than at the higher S/R. Noting that the scheduling index in NEH is the summation of the processing times for each job on all machines, then at the higher S/R such scheduling index may loose its significance due to the large setup times relative to the processing times. On the other hand there is no clear trend for the effect of varying S/R on CDS-M-2.

The associated makespan with minimizing total flow time (AMF) from NEH is the worst compared with the other methods. This is clear in Appendix A. CDS-M-2 generates relatively improved AMF as a side result to minimizing makespan. AMF from both methods improves as problem size increases.

Regarding the CPU time records in Appendix A, it is observed that CPU time for NEH is longer than CDS-M-2, ranging from 0.04 for the smallest problem up to about 9.45 sec for the largest problem. For CDS-M-2 the CPU time ranges from 0.046 up to 6.84 sec.

4.1.2 The Iterative Improvement Techniques

4.1.2.1 The tabu search heuristics

Studying the results of the comparison of the TS methods; Fig. 4.4, it can be found that the proposed TS-M-1 is superior to the other TS versions. It outperforms them in most of the cases. At S/R = 10 and using a random initial solution, TS-M-1 is the best all the time. At the higher S/R the performance of all versions is relatively better and more robust to increasing the problem size.

Counting the number of times in which a TS version is better than the other versions, Table 4.1 is formulated. From the table, it is observed that using a random initial solution TS-M-1 generates the best results for the largest number of times. The original TS ranks the second and then TS-M-2. S/R does not seem

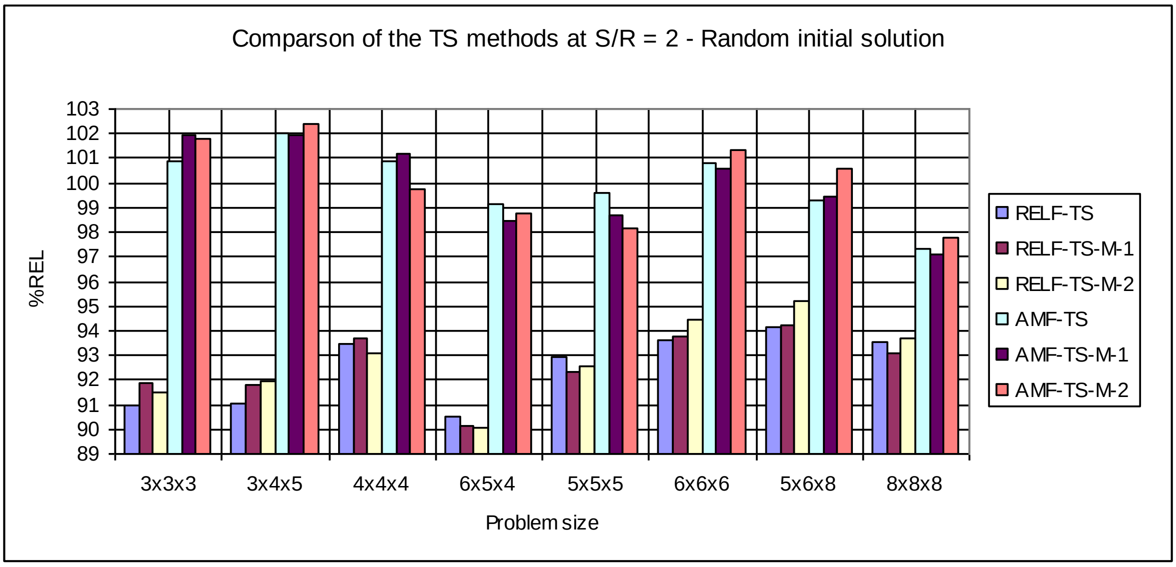
Table 4.1 Statistics of the performance of the TS heuristics – Total flow time

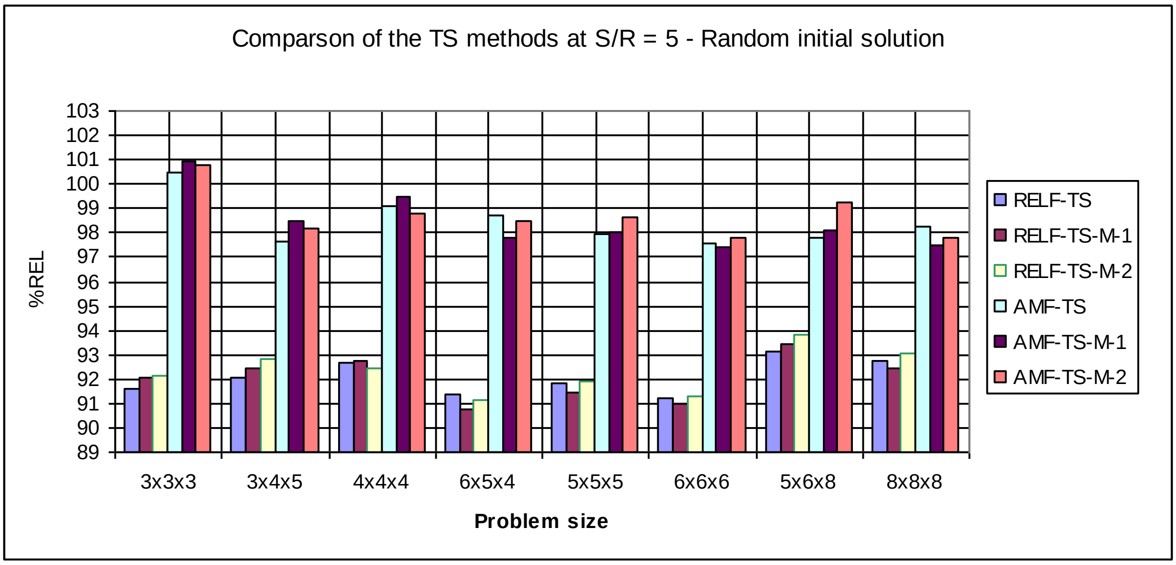
Using a Random Initial Solution									
	TS			TS-M-1			TS-M-2		
S/R	First	Second	Third	First	Second	Third	First	Second	Third
2	3	1	4	2	4	2	3	3	2
5	3	4	1	4	3	1	1	1	6
10	-	4	4	8	-	-	-	4	4
Sum	**6**	**9**	**9**	**14**	**7**	**3**	**3**	**7**	**14**
Using a Hitomi Initial Solution									
	TS			TS-M-1			TS-M-2		
S/R	First	Second	Third	First	Second	Third	First	Second	Third
2	2	-	6	6	1	1	-	7	1
5	2	1	5	6	2	-	-	5	3
10	2	1	5	6	2	-	-	5	3
Sum	**6**	**2**	**16**	**18**	**5**	**1**	**-**	**17**	**7**

to affect their ranks. When using Hitomi as an initial solution, TS-M-1 is still the best, even better than with the random initial solution. TS-M-2 is never in the first position, however it comes the second for the largest number of times (17 out of 24). Observing that TS-M-1 is third for one time only, then TS-M-2 is better than original TS for 66.67% of the cases using Hitomi’s initial solution.

Fig.4.5 shows the effect of using Hitomi as an initial solution on TS. Largest differences are observed for TS-M-2, while the least effect is seen for the original TS. Since TS-M-2 could outperform the original TS when Hitomi is used, then it is the use of the complete LTM in TS-M-2 that made it possible for such a relatively good initial solution to release more potentials from the TS procedure.

It is thus concluded that the LTM should be used completely in the two scheduling phases of GS. However, using a partial LTM in the family phase with a simple straight way to make use of the search efforts in the job phase (TS-M-1)





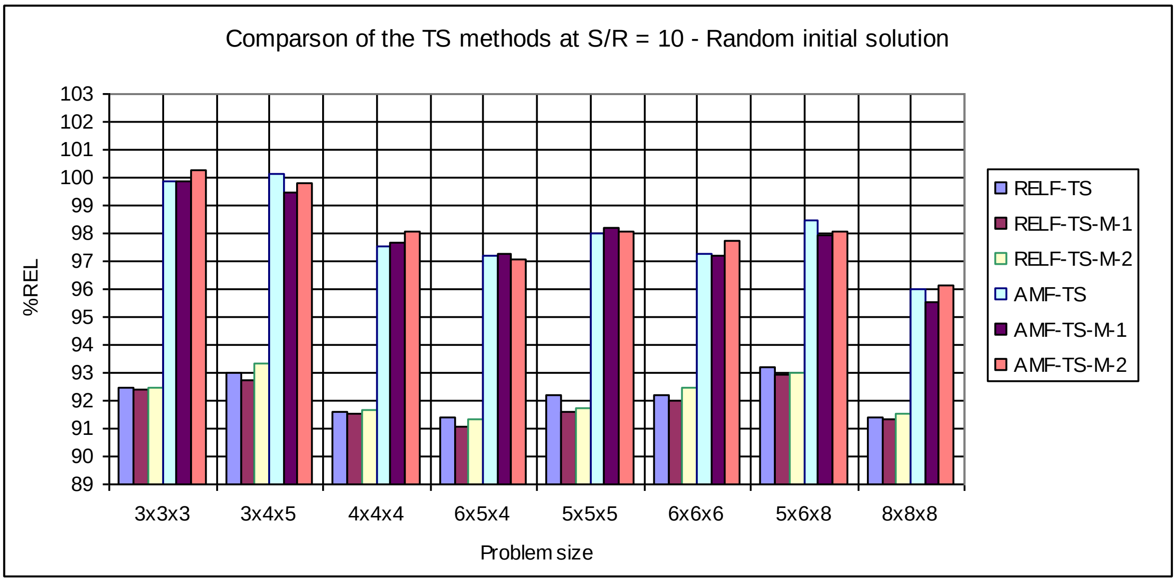
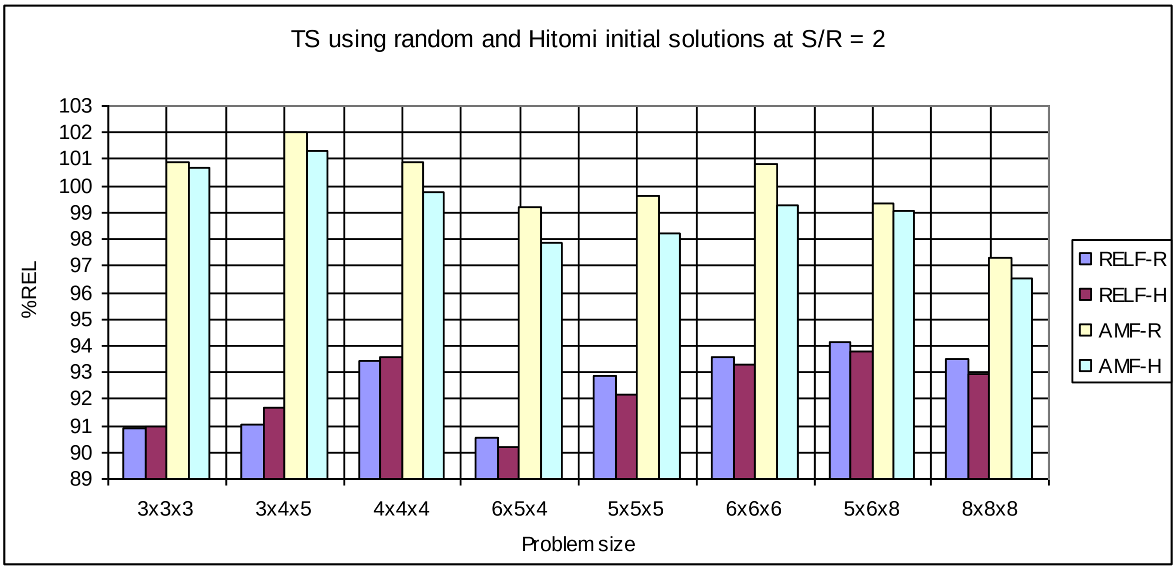
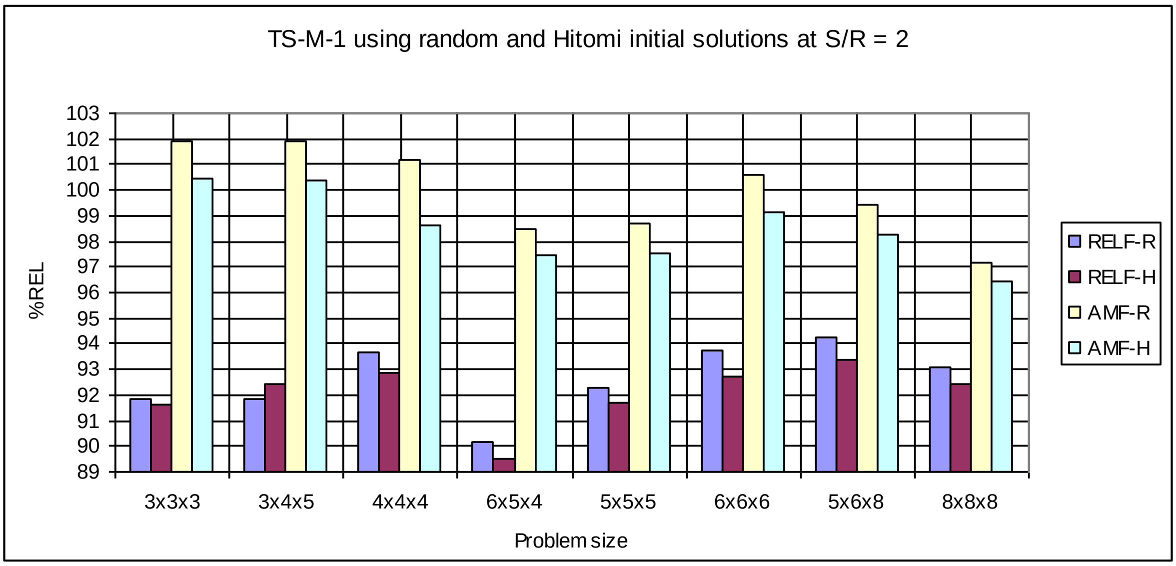


Fig. 4.4 Performance of the TS methods using random initial solution - Total flow time





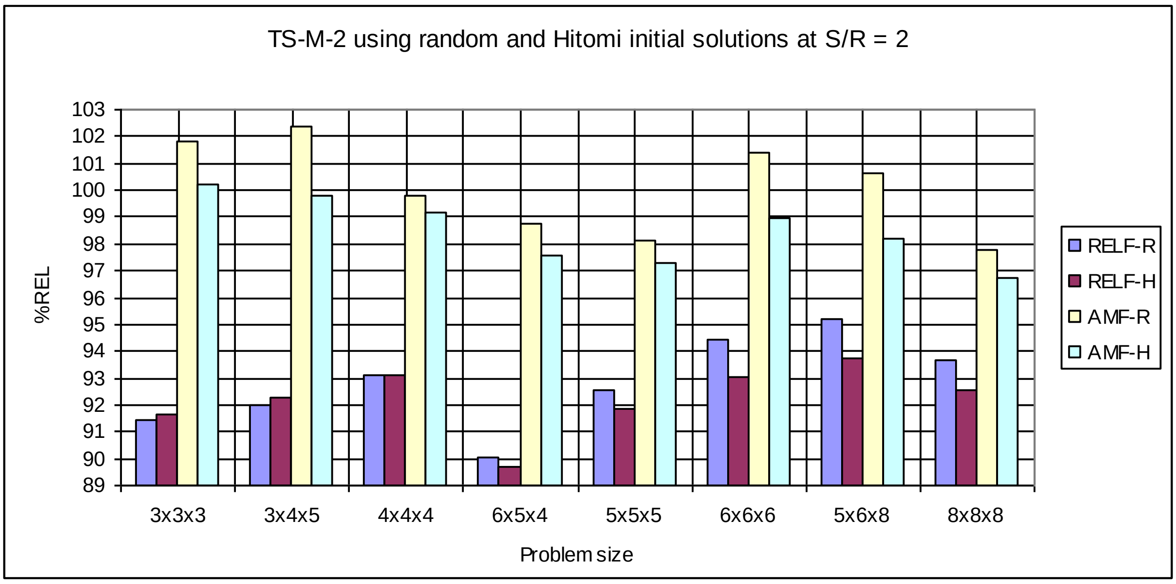


Fig.4.5 The effect of using Hitomi initial solution on TS methods at S/R = 2 - Total flow time

is found better than the complete LTM in TS-M-2. Hence, the complete LTM is needed but LTM is not properly defined for the TS heuristic as proposed in [12]. That is considering only the number of times a family comes in some position in the trial solutions during the iterations is not the enough information to operate LTM. More search-based information may be concerning phases' interaction, deserve to be considered.

It is the nature of the GS problem that the reason why LTM as in [12] does not work as expected. For example, a family (or a job) may come in a position " t " for the largest number of times during the iterations. Then in generating the new initial solution, this family (job) will be placed in position " t ". Let another family (job) come in another position " v " a number of times such that it will be in this position in the new initial solution. It is possible that placing the first family (job) in position " t " and the second one in position " v ", will be a situation that prevent reaching a good complete schedule. Perhaps the number of times that the first family (job) was in position " t " did not coincide with the times in which the other one was in position " v ". In other words getting a good schedule with the first family in position " t " is conditioned by that the other family is not in position " v " or vise versa.

This is similar to the possible situation in using the multi-pass methods when switching from family phase to job phase for example. A family schedule may be a constraint during the job phase that will prevent reaching some better possible schedules. Thus even for the iterative improvement methods phases' interaction has to be considered in the structure of the algorithm.

The effect of increasing S/R is positive in general for the TS methods. From Fig.4.4 it is observed that for the higher S/R, the method is more robust and able to keep its level of performance for the larger problems. The three versions behave similar to each other. An average of about 1.5% improvement in makespan (AMF) is achieved by the TS methods. A maximum of about 3.4% is observed at 8x8x8 problems for TS-M-1. AMF is better for the larger problems as shown in Fig. 4.5 and this is true for all S/R values.

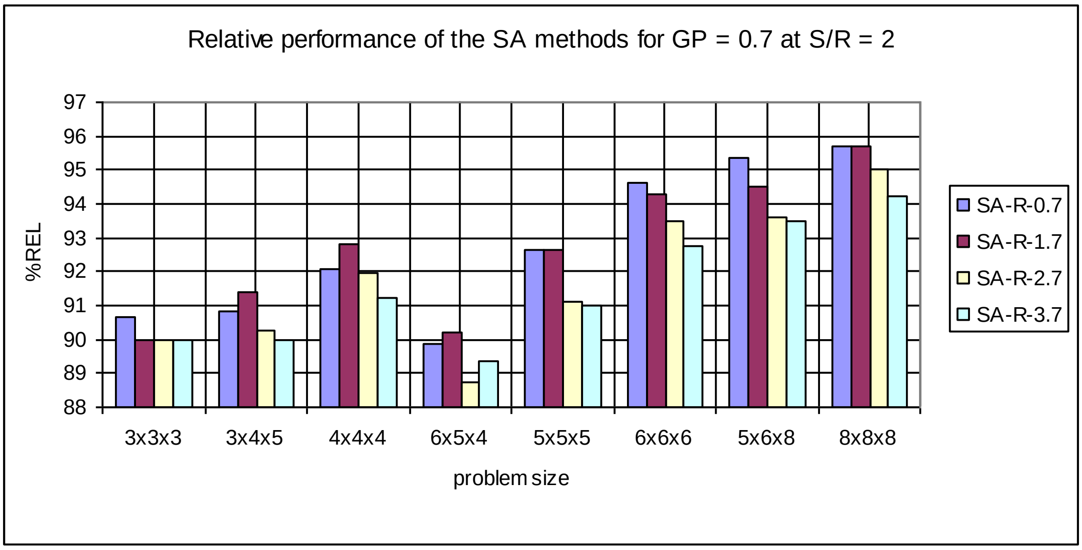
4.1.2.2 The simulated annealing heuristics

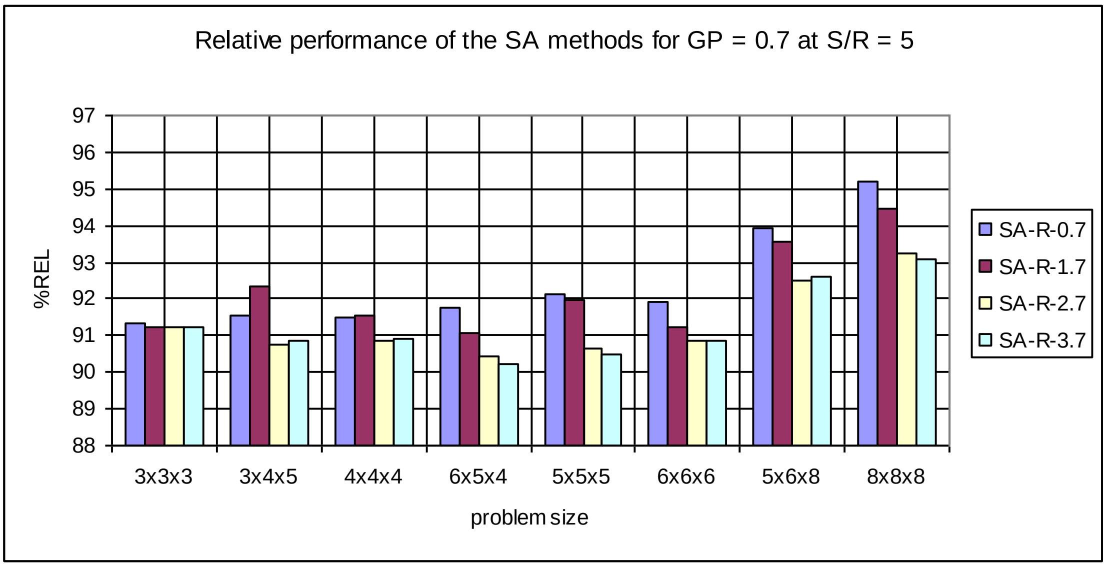
From Fig.4.6 it is observed that SA-M-2 and SA-M-3 are better than the original SA and SA-M-1. This is true for all conditions. This shows that the change-dependent acceptance probability improved the performance of SA. A change dependent acceptance probability can avoid solutions that results in drastic changes in the objective function value. If a non-improving solution is reached during iteration step X with a large value of Δ, then the value of the acceptance probability $AP_x = EXP(- \Delta / X)$ would be low due to the negative sign. Hence the procedure is forced toward asymptotic convergence.

It was found also that SA-M-1 is better than the original SA for about 58.25% of times. Similarly SA-M-3 is better than SA-M-2 for about 61.25% of times. Thus the control on the behaviour of the random numbers in SA-M-1 and SA-M-3 could lead to better performance. Consequently SA-M-3 is preferable to the original SA and the other SA versions.

The performance of SA methods generally tends to deteriorate as the problem size increases. SA is dependent to a large extent on the use of the random numbers and as problem size increases this is a disadvantage leading to the inferior performance. Meanwhile, using Hitomi as an initial solution made no important differences.

Varying the GP value affects as follows. For the original SA and SA-M-1, GP of 0.7 and 0.9 result in the best performance in most cases, while 0.5 gives fairly good results. This is shown in Fig.4.7. For SA-M-2 and SA-M-3, GP of 0.5 and 0.7 are the best most often. This is shown in Fig.4.8. Thus it is stated that spending the majority of the search effort to the family phase is more worthy.


Relative performance of the SA methods for GP = 0.7 at S/R = 2


Relative performance of the SA methods for GP = 0.7 at S/R = 5

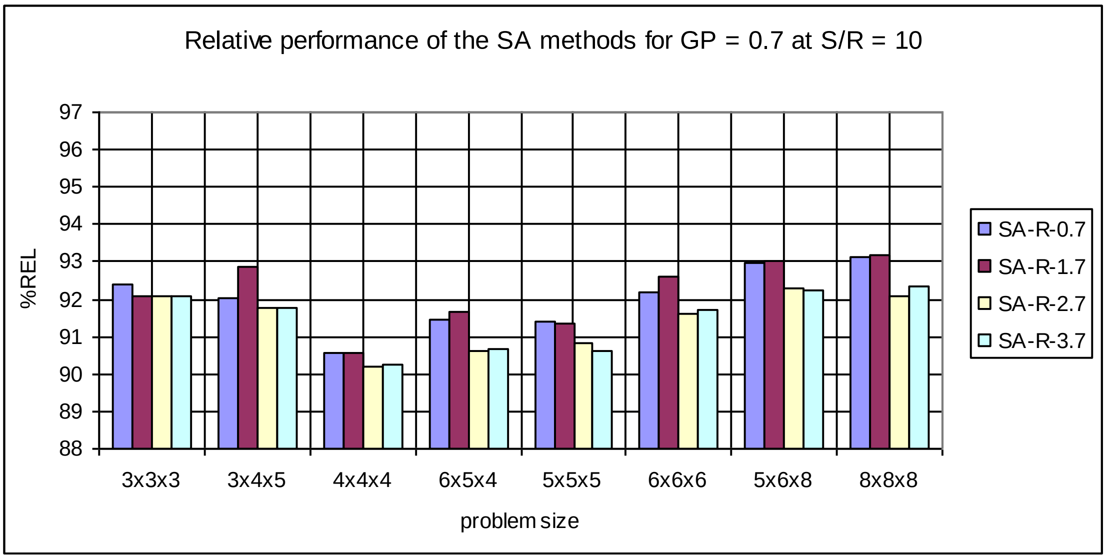
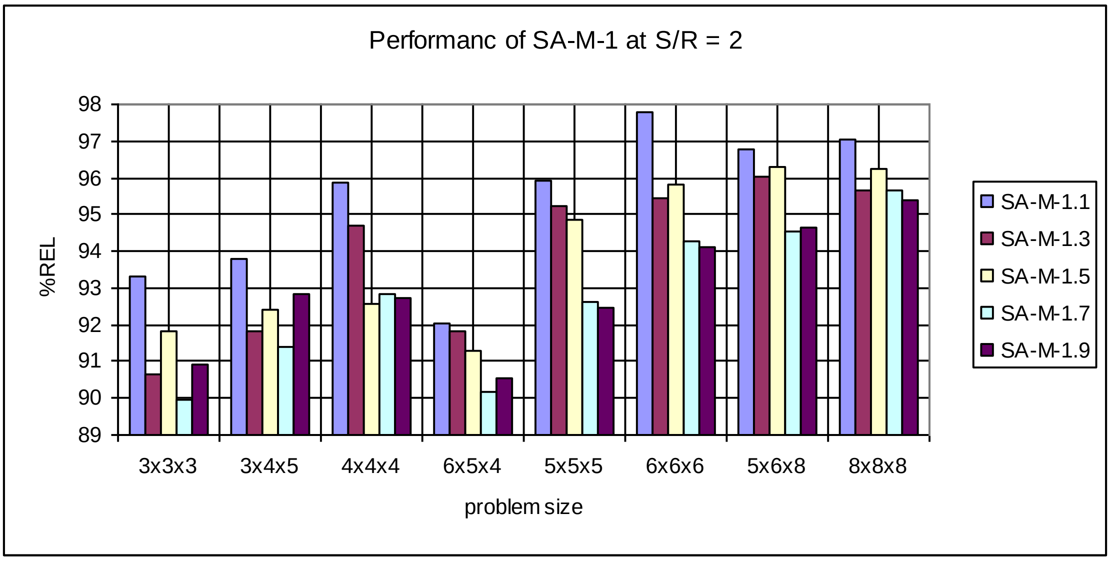
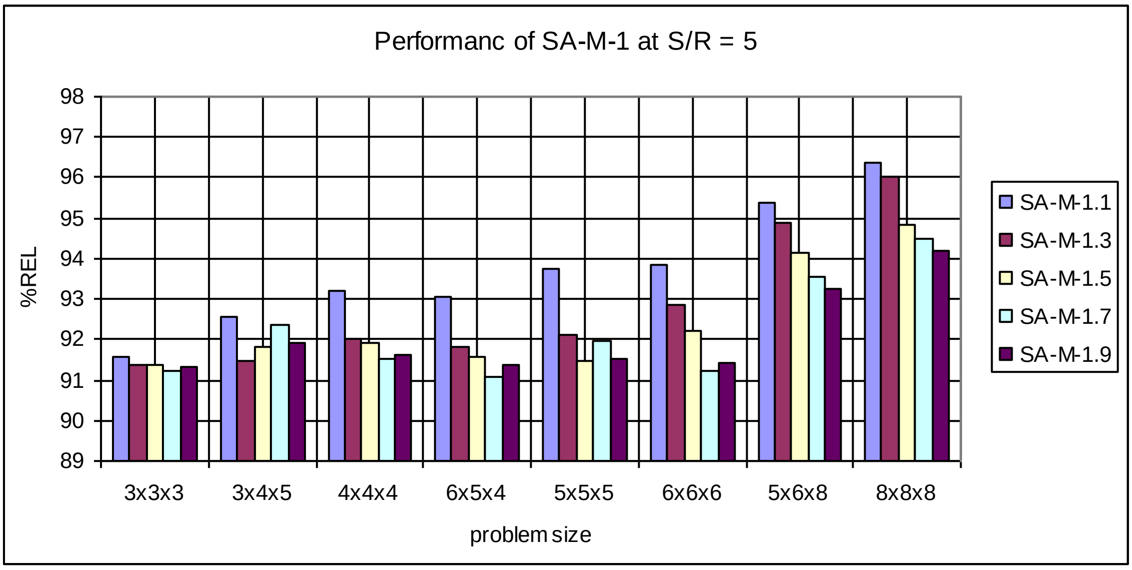

Relative performance of the SA methods for GP = 0.7 at S/R = 10

Fig.4.6 Performance of the SA methods using random initial solution, GP = 0.7- Total flow time





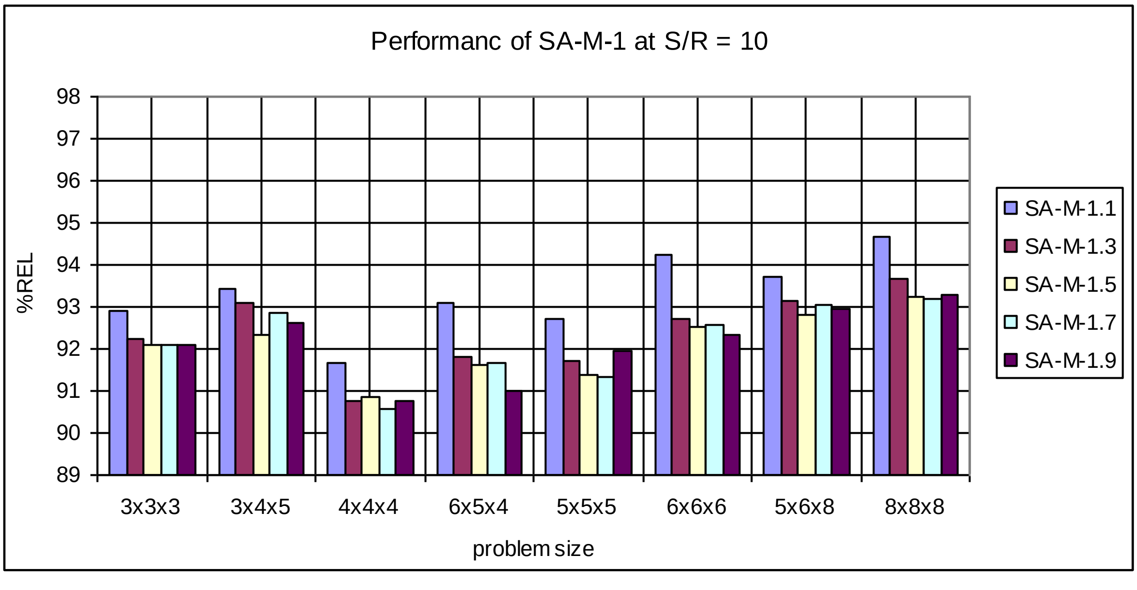
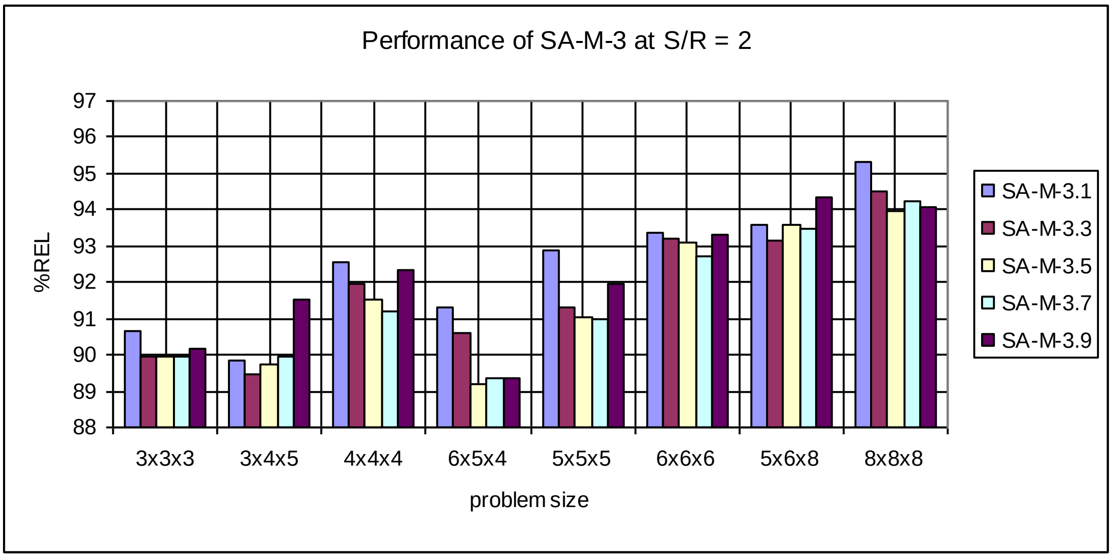
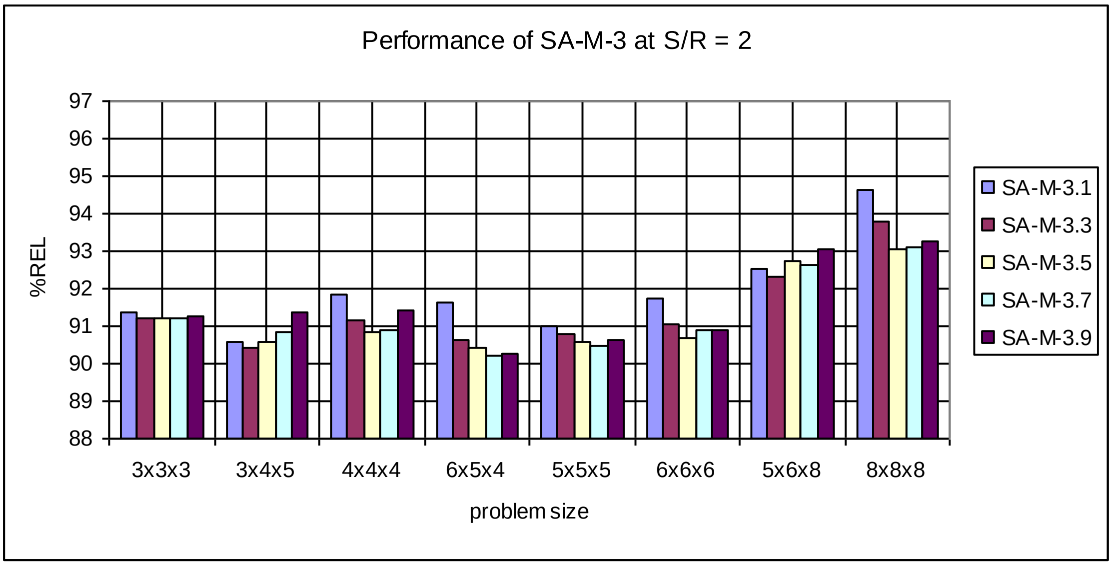


Fig.4.7 Performance of the SA-M-1for different GP values - Total flow time





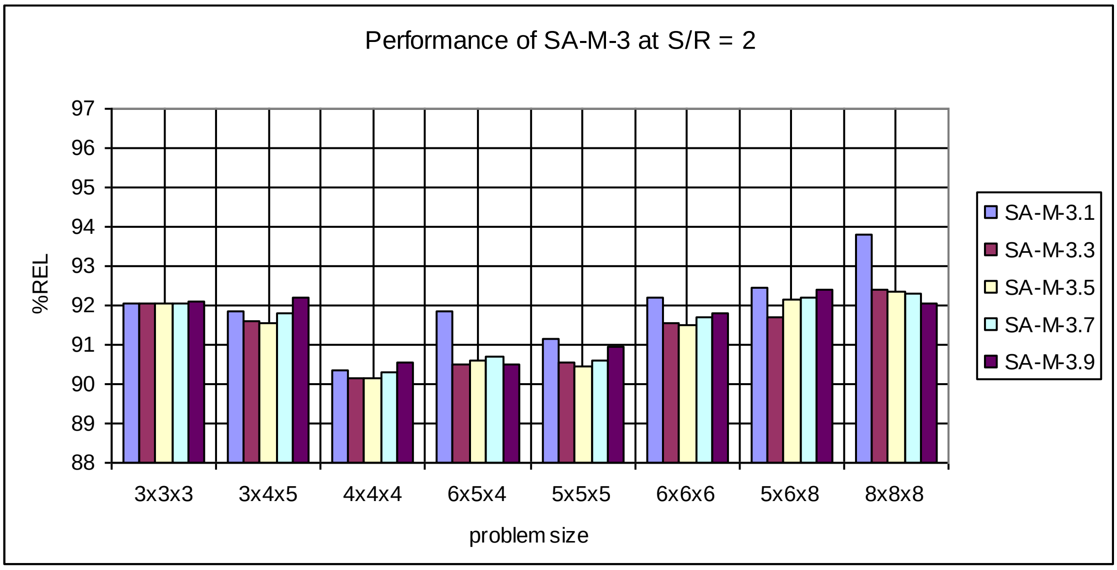


Fig.4.8 Performance of the SA-M-3 for different GP values - Total flow time

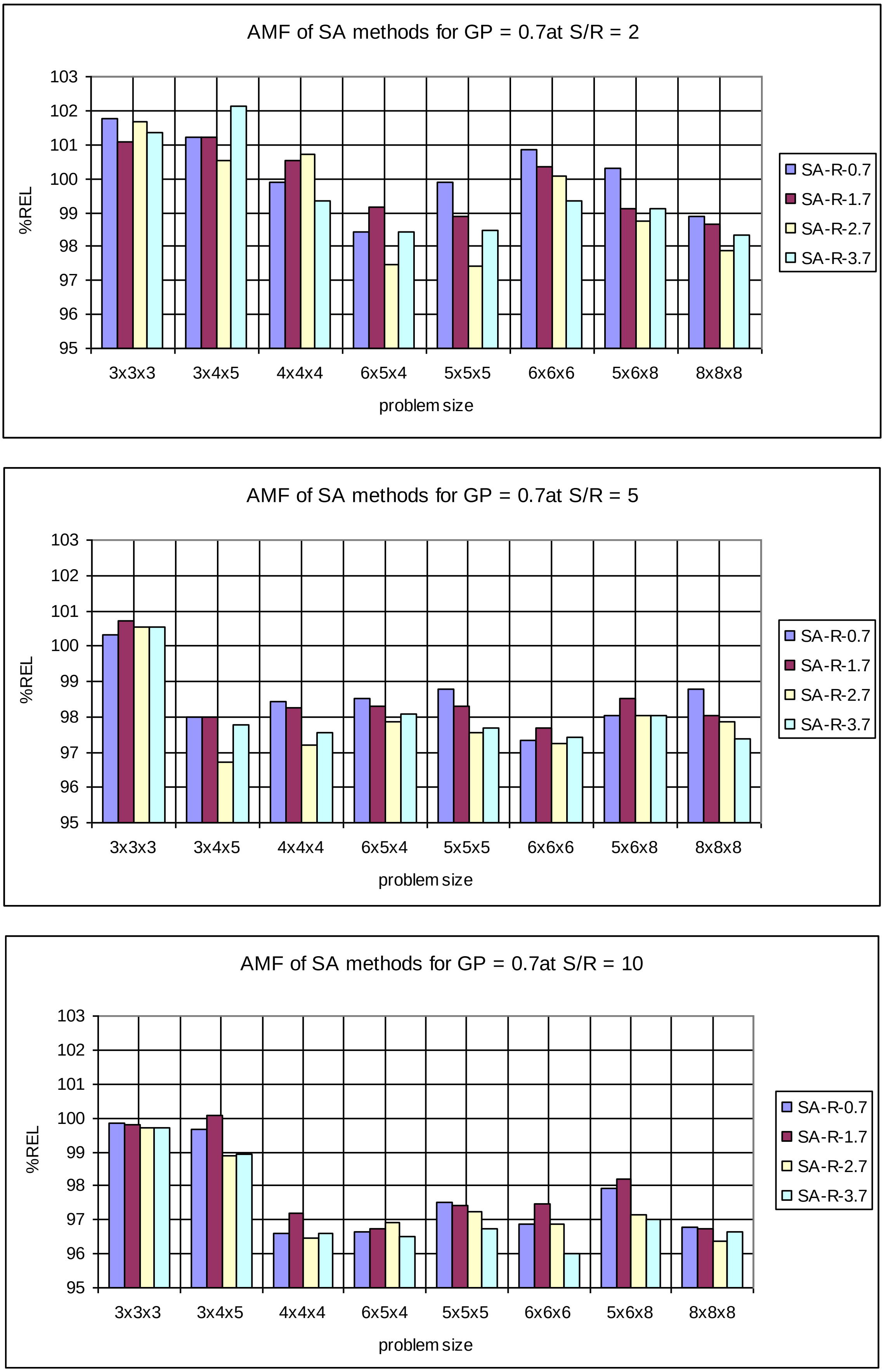


Fig.4.9 AMF from the SA methods for GP of 0.7 - Total flow time

From Figs. 4.7 and 4.8, it is observed that the differences among the various GP values are larger for SA-M-1 (and similarly for original SA) than for SA-M-3 (and SA-M-2). Increasing S/R also reduces the effect of GP. Thus it can be concluded that as the performance of the heuristic improves the effect of GP decreases. It is expected that if the performance is improved more, the two scheduling phases would be equally important. This is reminding with that obtaining the optimal solution by the branch and bound is characterized by the simultaneous determination of the schedules in the two phases.

The SA methods could improve makespan as a side-result for minimizing total flow time (AMF). A maximum improvement of about 3.9% is observed for SA-M-3. The average improvement is about 1.72%. This is shown in Fig.4.9 for SA-M-3. AMF is better from both SA-M-3 and SA-M-2 than from SA-M-1 or SA. In Fig.4.9 it is shown that unlike total flow time, AMF improves at the larger S/R and larger problems and this is true for the other SA versions.

4.1.3 Comparison of Best Heuristics for Total Flow Time

It is expected that the iterative methods are superior to the simple methods. This can be observed in Fig.4.10. In general the SA-M-3 at GP = 0.7 (SA-M-3.7) ranks first, TS-M-1 second, NEH third and CDS-M-2 is last. NEH is comparable in some cases to the two the iterative methods at the smaller S/R values. TS-M-1 outperforms SA-M-3.7 for the largest problem for all S/R. It is noted that SA-M-3.7 tends to deteriorate at larger problems while TS-M-1 is more stable.

In Fig.4.11 it is shown that AMF from NEH is the worst. CDS-M-2 is best for the smaller S/R. At the higher S/R CDS-M-2 is comparable and generally preferable to the iterative methods. SA-M-3 is better than TS-M-1 in most cases, But TS-M-1 is better than SA-M-3.7 at the largest problem.

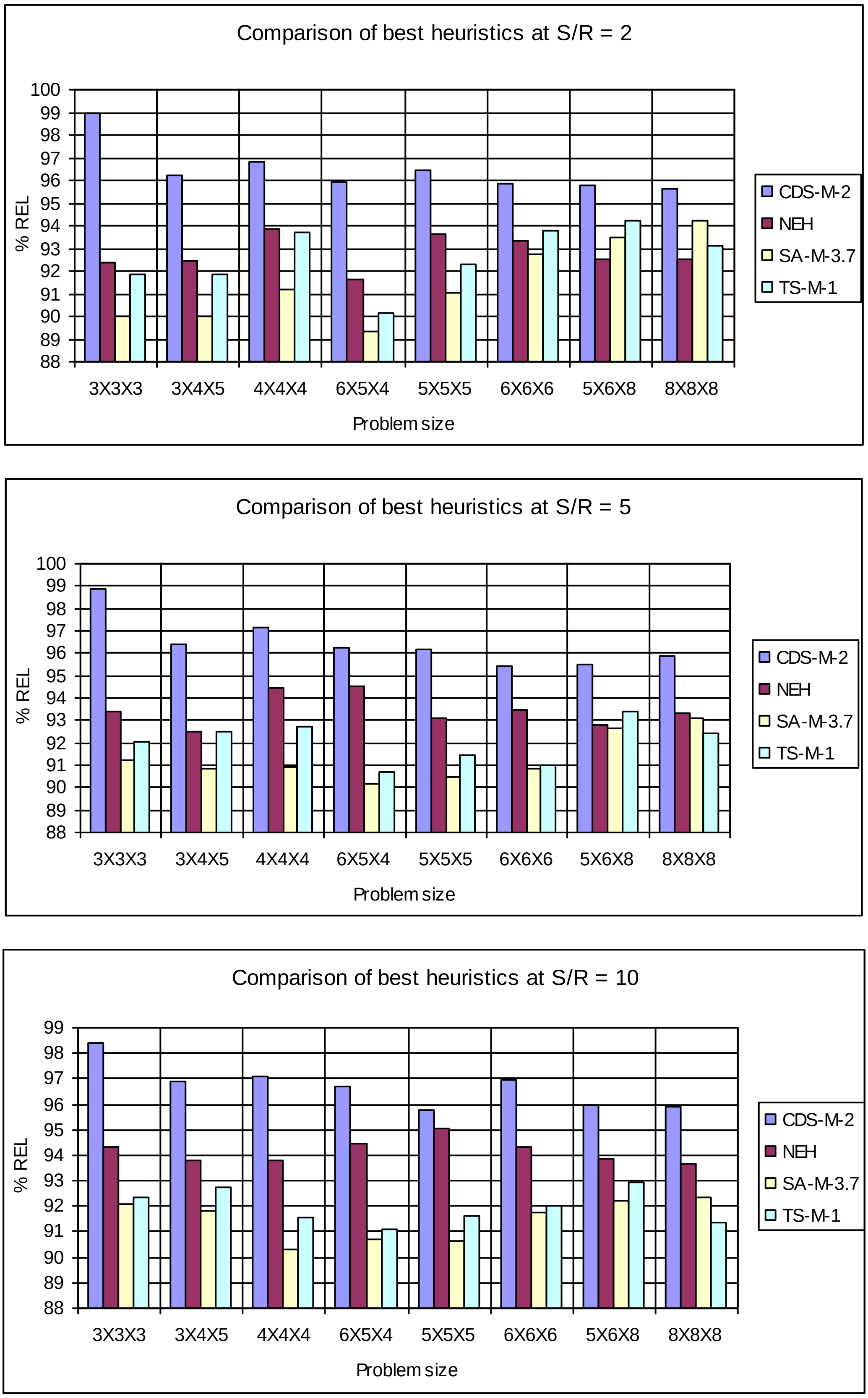
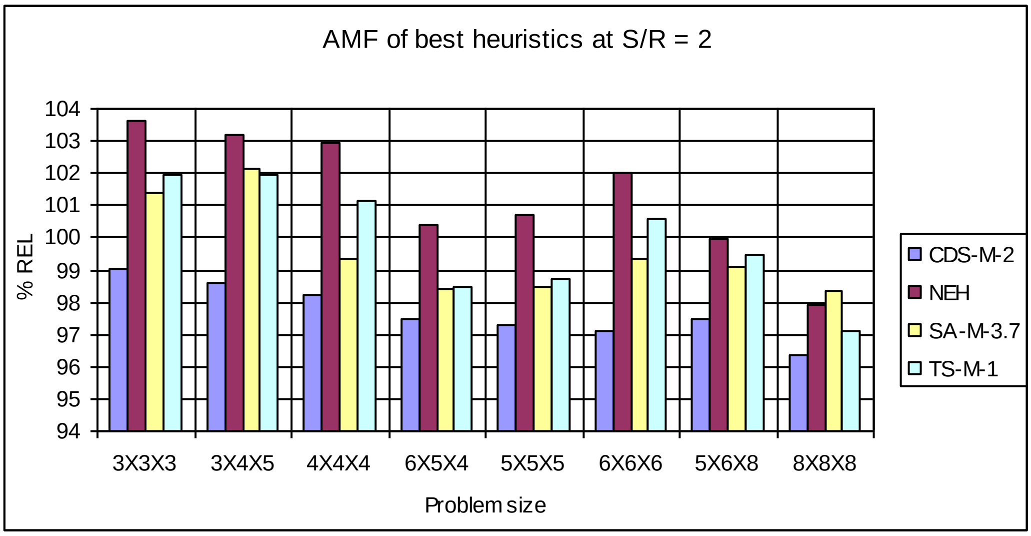
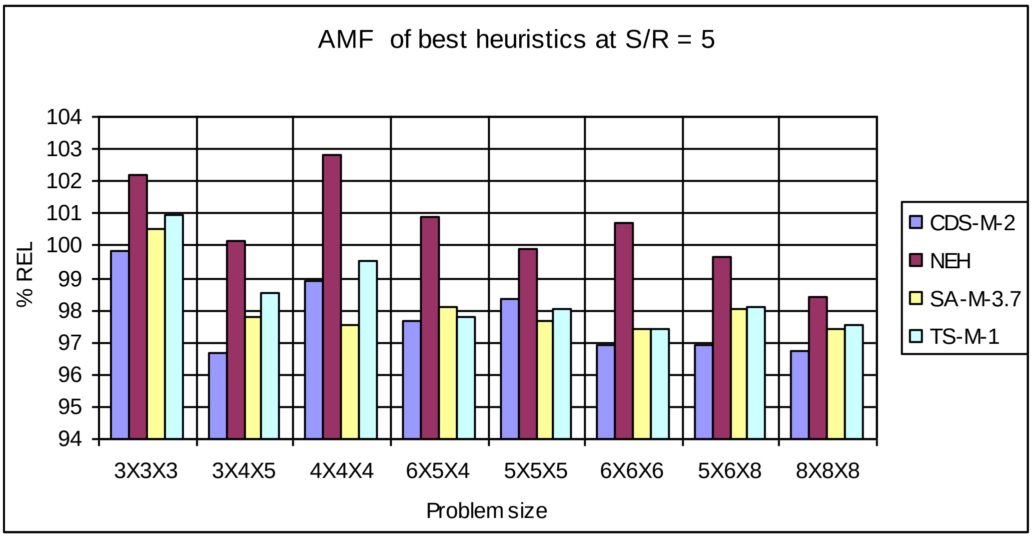


Fig.4.10 Comparison of the best heuristics - Total flow time





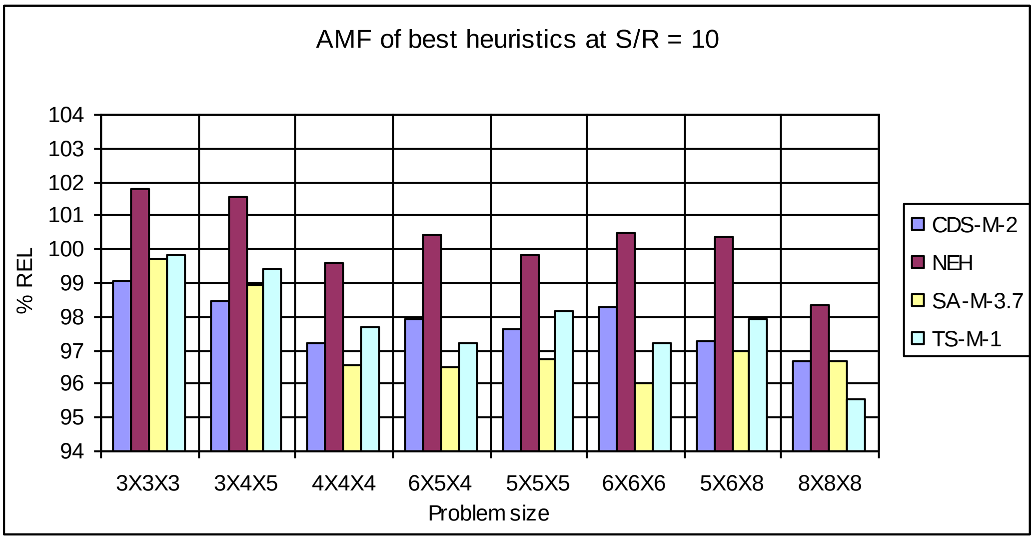


Fig.4.11 AMF from the best heuristics - Total flow time

The CPU time for the TS-M-1 is the longest, ranging from 2.984 sec up to 1773.78 sec, while for SA-M-3.7 it ranges from 4.357 sec up to 75.821sec. Meanwhile the longest CPU time for CDS-M-2 is 6.84 sec and for NEH 9.45 sec.

4.2 RESULTS WITH RESPECT TO MAKESPAN

4.2.1 The Single and Multi-Pass Methods

Regarding the adoption of Rajendarn's modification in the simple methods, it is shown in Fig. 4.12 that Rajendran's modification is generally ineffective for minimizing makespan. This is true for all problem sizes as can be found from the tables of results in Appendix B.

The proposed iterative CDS-M-2 is the best performing CDS version, as shown in Fig. 4.12. This is true for all problem sizes. This is the result of its iterative behaviour that makes it able to handle the constraining effect of the phases' interaction. The superiority of it is limited by the finite number of solutions enumerated by the CDS technique and is at the expense of the CPU time as was the case of minimizing total flow time. This result emphasizes the consideration of the phases' interaction, which in turn gives more significance to developing and using the iterative improvement techniques.

Unlike the case of minimizing total flow time, NEH is not always the best simple method. In fact CDS-M-2 outperforms NEH more frequent. This is shown in Fig. 4.13. In Fig.4.14 it is noticed that NEH shows lower performance at the higher S/R. This was also observed in Sec. 4.1.1 indicating that the scheduling index in NEH is less significant at the higher S/R ratios. Meanwhile, there is no clear trend for the effect of S/R on CDS-M-2 as in Fig. 4.15. From Figs. 4.14 and 4.15, it is observed that both NEH and CDS-M-2 slightly improves as the problem size increases which is clearer for CDS-M-2.

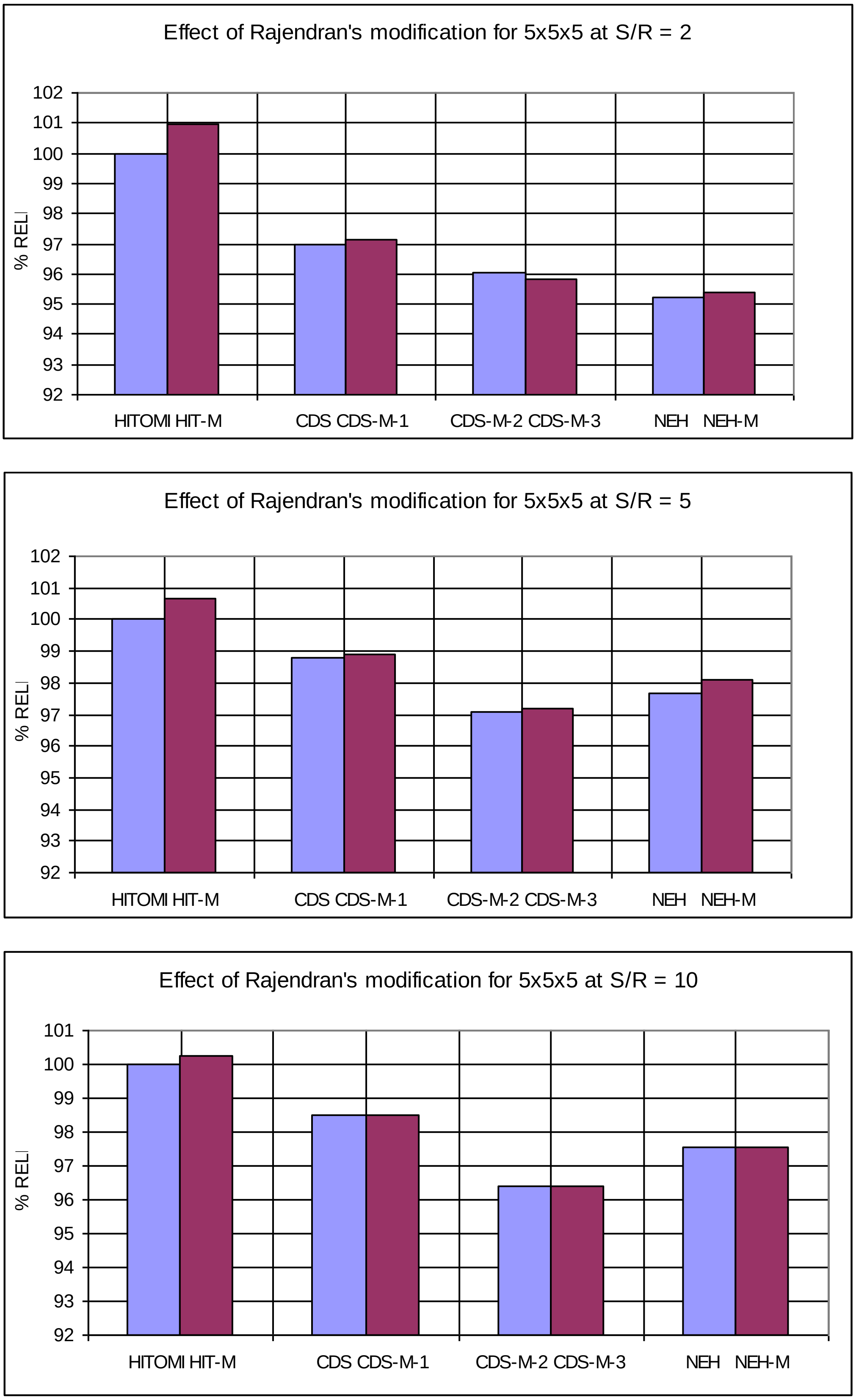
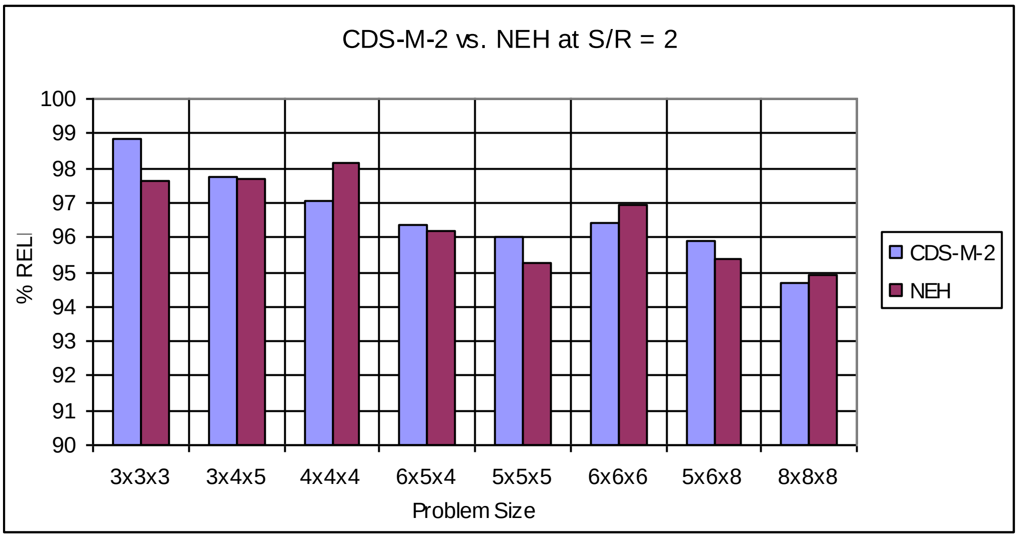
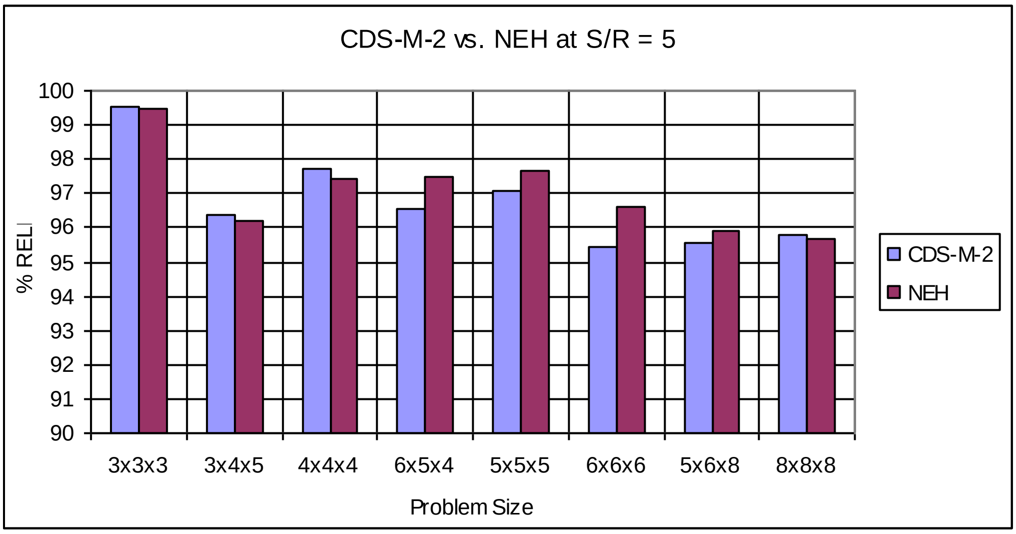


Fig.4.12 The effect of Rajendran's modification on the simple methods - Makespan

🟦 Without Rajendran 🟪 With Rajendran





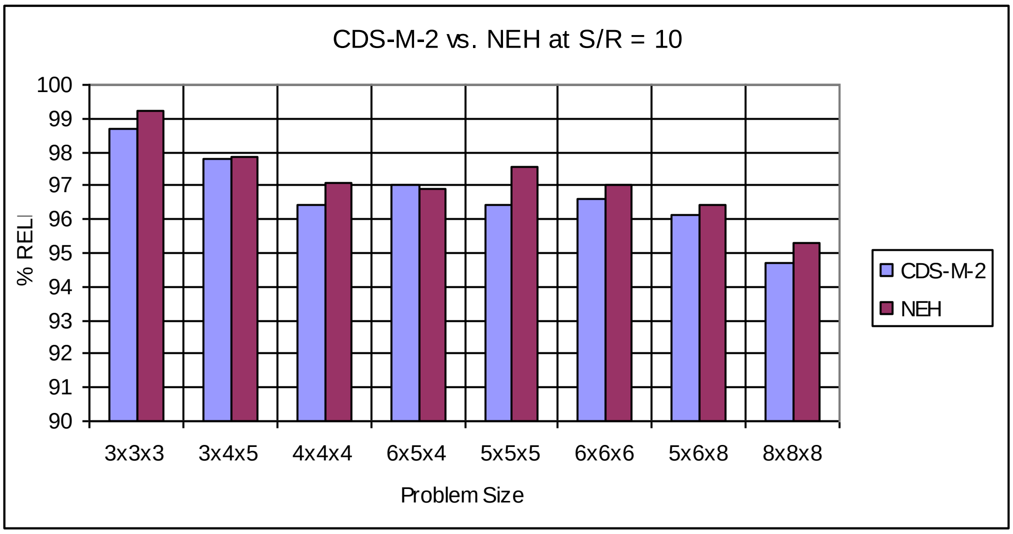


Fig.4.13 Comparison of CDS-M-2 and NEH - Makespan

NEH generates good AFM while minimizing makespan. As in Fig. 4.14 the AFM is improved over the reference value of Hitomi more than the improvement in makespan although makespan is the main objective. This is true for all conditions. For AFM a maximum of 5.73% improvement is achieved at 5x6x8 problems, and a minimum of 3.43% is achieved at 3x3x3 problems. For the main objective of makespan a maximum of 4.72% improvement is achieved at 8x8x8, and minimum of 1.21% at 3x3x3. Overall average improvement in AFM is 4.43%, and for makespan it is 3.1%. Meanwhile CDS-M-2 makes an average of 2% improvement.

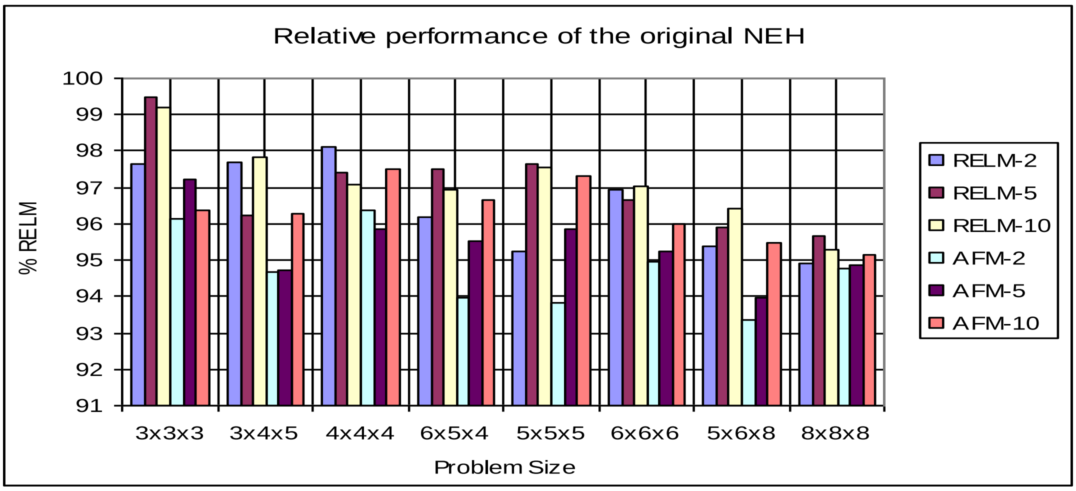


Fig. 4.14 Level of performance of original NEH – Makespan

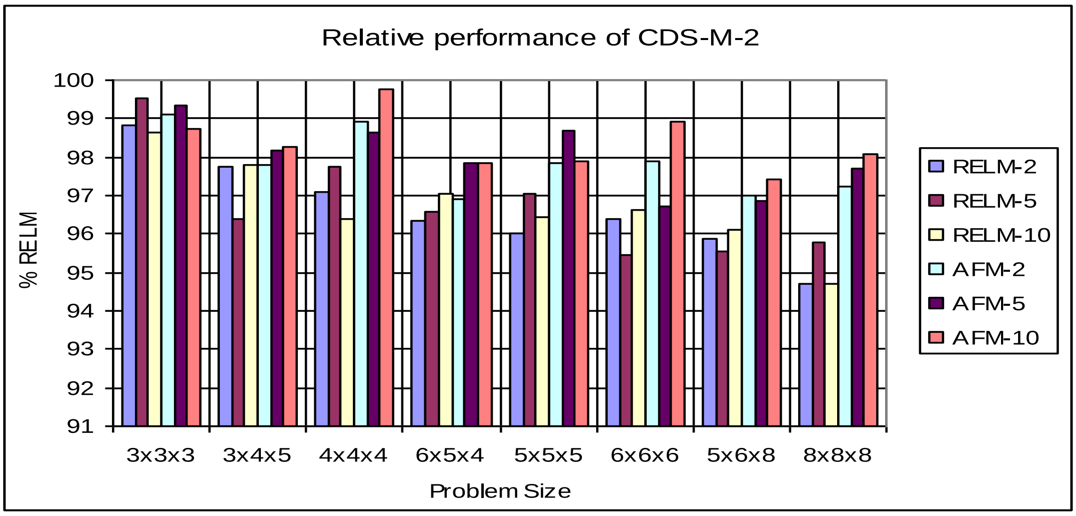


Fig. 4.15 Level of performance of the CDS-M-2 – Makespan

4.2.2 The Iterative Improvement Techniques

4.2.2.1 The tabu search heuristics

Studying the results of the TS methods and Fig. 4.16 it is observed that TS-M-1is best performing in most cases. The performance of the TS methods tends to improve as the problem size increases. Increasing S/R ratio accelerates the improvement rate of the TS heuristics as problem size increases.

Table 4.2 shows the number of times in which a TS version is better than the other versions. It is noticed that TS-M-1 is the best for 66.67% of the cases when using random initial solution. Original TS is the second and TS-M-2 comes last. When using Hitomi’s initial solution, TS-M-1 is still the best, and for 83.33% of times. The original TS could be the best for 17.67% of times, mainly for the smaller problems and with negligible differences between it and the other two versions. However, comparing TS and TS-M-2, it can be found that TS-M-2 outperforms the original TS when using Hitomi’s initial solution for more than 70% of times. Similar to the case of minimizing total flow time, the complete LTM in TS-M-2 is that made it possible to improve its performance by the use of Hitomi’ s initial solution.

In Fig.4.17, it is shown that using Hitomi improved performance of the TS methods. Largest effect is seen for TS-M-2, while the least is seen for the original TS. Hence, the same conclusion derived in Sec.4.1.2.1 about the use of LTM and the information it should contain is applicable for optimizing makespan.

AFM from the TS methods is well improved over the reference value. It is better as problem size increases for both the random and Hitomi’s initial solutions. It can be observed in Fig.4.17 that TS-M-1 is also the best performing for the AFM. A maximum improvement of 5.83% is observed for TS-M-1 at 6x5x4 and a minimum of 1.05% at 3x3x3. Average improvement is 2.41%.

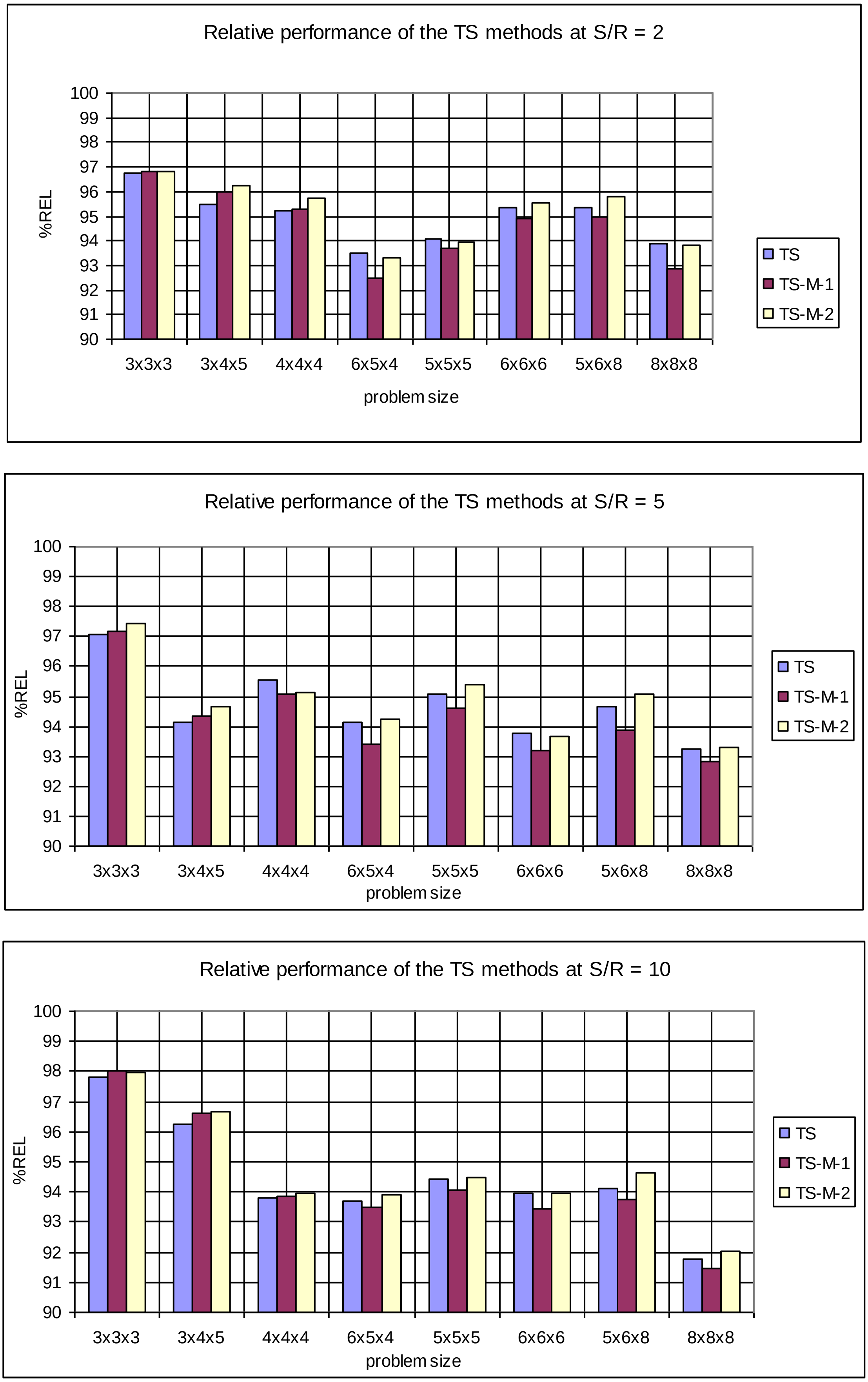
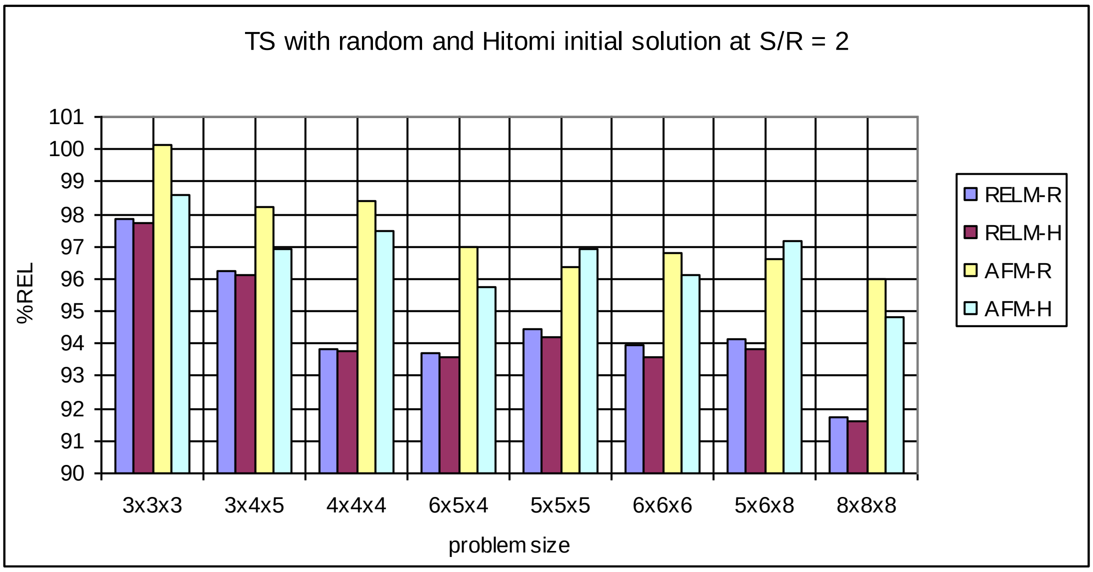
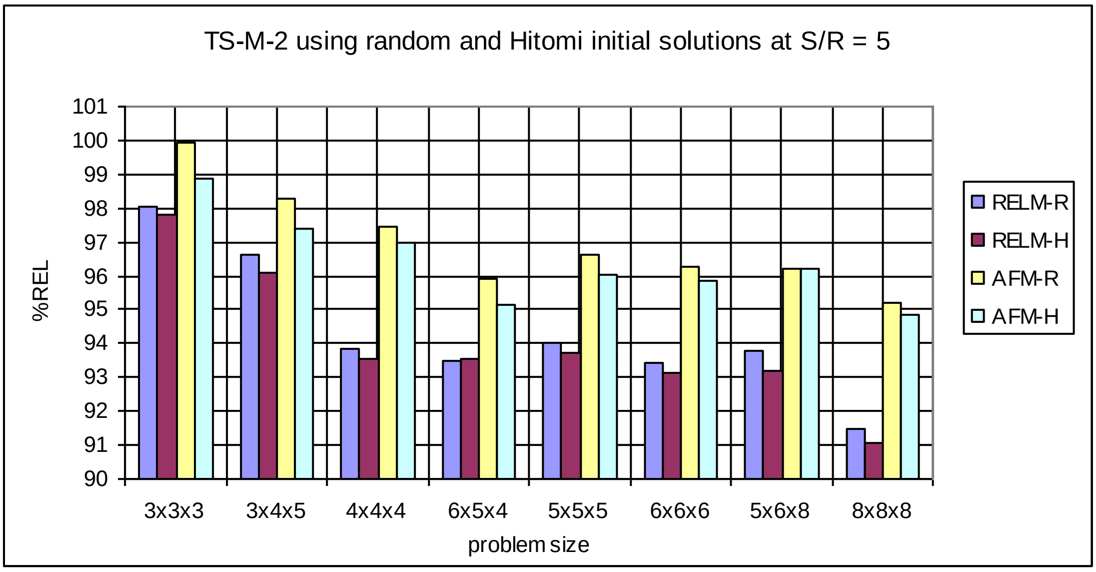


Fig.4.16 Performance of the TS methods using random initial solution - Makespan





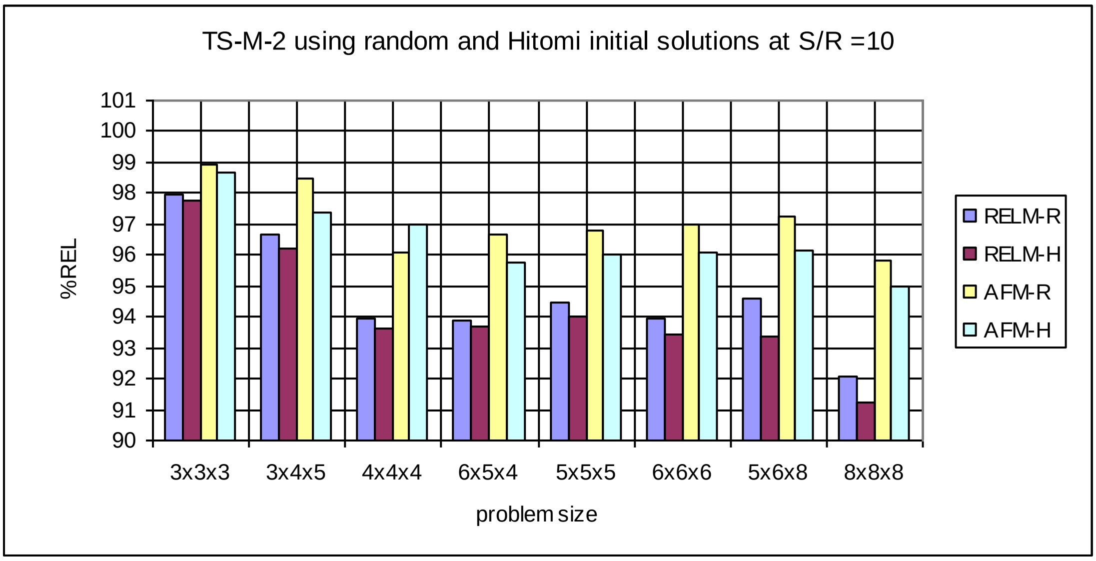


Fig.4.17 Effect of using Hitomi initial solution on TS methods at S/R = 10 - Makespan

Table 4.2 Statistics of the performance of the TS versions - Makespan

Using a Random Initial Solution									
	TS			TS-M-1			TS-M-2		
S/R	First	Second	Third	First	Second	Third	First	Second	Third
2	3	2	3	5	3	-	-	3	5
5	2	4	2	6	2	-	-	2	6
10	3	5	-	6	2	-	-	1	7
Sum	**8**	**11**	**5**	**16**	**7**	**1**	**0**	**6**	**18**
Using a Hitomi initial Solution									
	TS			TS-M-1			TS-M-2		
S/R	First	Second	Third	First	Second	Third	First	Second	Third
2	2	-	6	6	2	-	-	6	2
5	1	1	6	7	1	-	-	6	2
10	1	2	5	7	-	1	-	6	2
Sum	**4**	**3**	**17**	**20**	**3**	**1**	**0**	**18**	**6**

4.2.2.2 The simulated annealing heuristics

Results of the SA methods show that the change-dependent acceptance probability versions (SA-M-2 or SA-M-3) are better than the change independent acceptance probability (original SA and SA-M-1). This is shown in Fig.4.18. In addition, from the tables of results in Appendix B, it is possible to observe that SA-M-1 outperforms the original SA for 75% of cases for both the random and Hitomi initial solutions. Similarly, SA-M-3 outperforms SA-M-2 for about 62% of cases. This indicates as in Sec.4.1.2.2, the necessity to add a form of control on the behaviour of the random numbers in the SA techniques. Consequently SA-M-3 is preferred to the original SA and the other SA versions.

It can be also seen in Fig.4.18 that the performance of the SA methods is generally better at the higher S/R. However, as problem size increases the performance becomes inferior. Using Hitomi as an initial solution made no important difference.

The GP factor has less effect for makespan than for total flow time. In Fig.4.19 and following the values in Appendix B, it can be found that best results are observed for GP of 0.5 to 0.7 most often. Then comes 0.9 and 0.3 with 0.9 slightly preferable. That is giving the majority of the search efforts to the family phase is more worthy in minimizing makespan as well.

It is noted that makespan is relatively simpler to optimize than total flow time, and hence the performance of SA is expected to be better for makespan than for total flow time. And as indicated in Sec.4.1.2.2, the effect of GP is less when the performance of the heuristic is improved. Hence, it becomes logical that GP has less effect here than it had with minimizing total flow time.

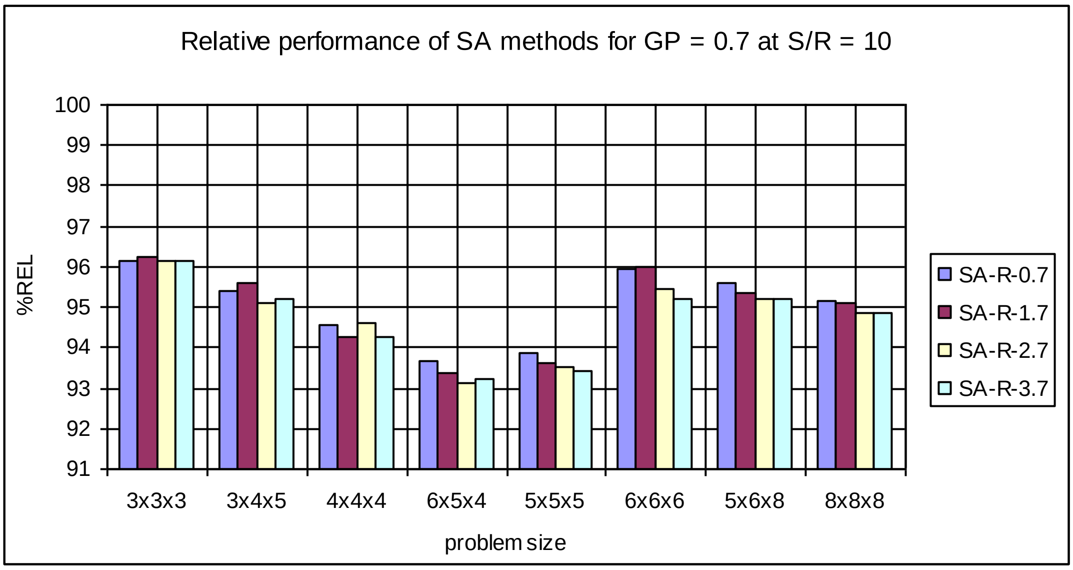
AFM from the SA methods is improved and as for makespan, AFM is better at the lower S/R but it deteriorates as problem size increases. A TS-M-1 is the best for AFM as well. maximum improvement of 4.74% is observed for SA-M-3 at 6x5x4 while a minimum of 1.43 is observed at 3x3x3. The average improvement is about 3.04%.

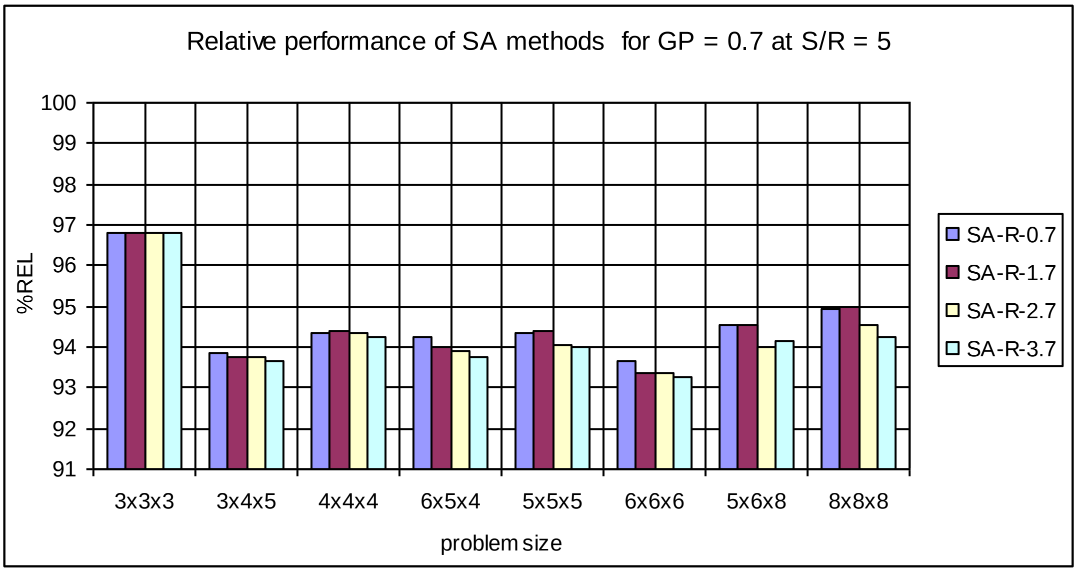
4.2.3 Comparison of Best Heuristics for Makespan

As shown in Fig.4.20, the two best iterative improvement heuristics are superior to the simple methods. In general SA-M-3.7 and TS-M-1 performs equivalently. TS-M-1 is more stable for the larger problem sizes than SA-M-3.7.

Longest CPU times are observed for TS-M-1 ranging from about 4.4 sec up to about 1789 sec, while for SA-M-3.7 the ranges is from about 4.4 sec up to 75.8 sec. Meanwhile the longest CPU time for CDS-M-2 is 5.92 sec and for NEH 9.30 sec.

AFM from the four heuristics in Fig.4.21 are comparable which is clearer at the higher S/R. In general NEH seems preferable to TS-M-1 and SA-M-3.7.





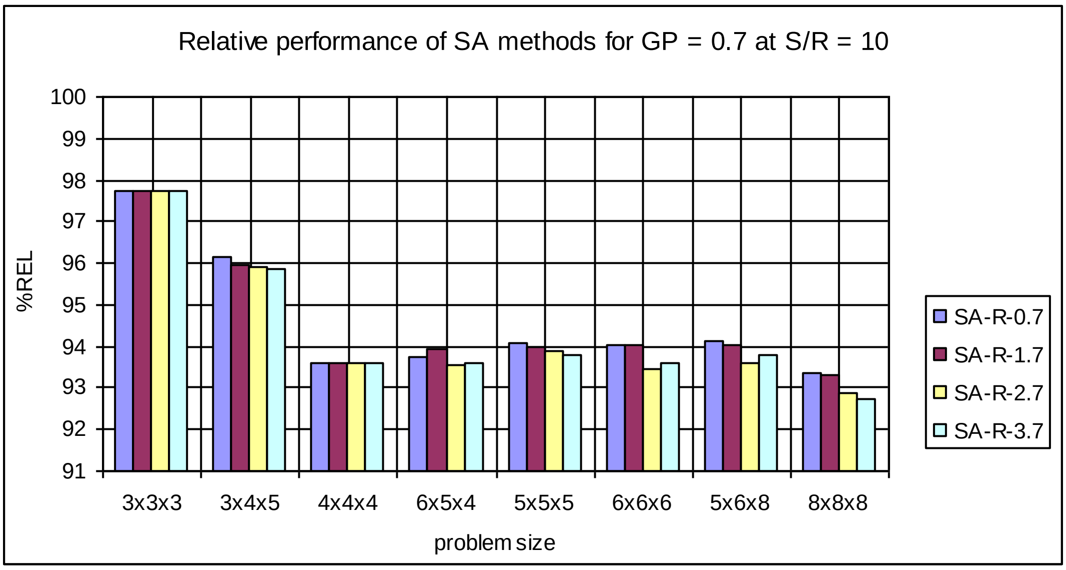


Fig.4.18 Performance of SA methods using random initial solution, GP = 0.7 - Makespan

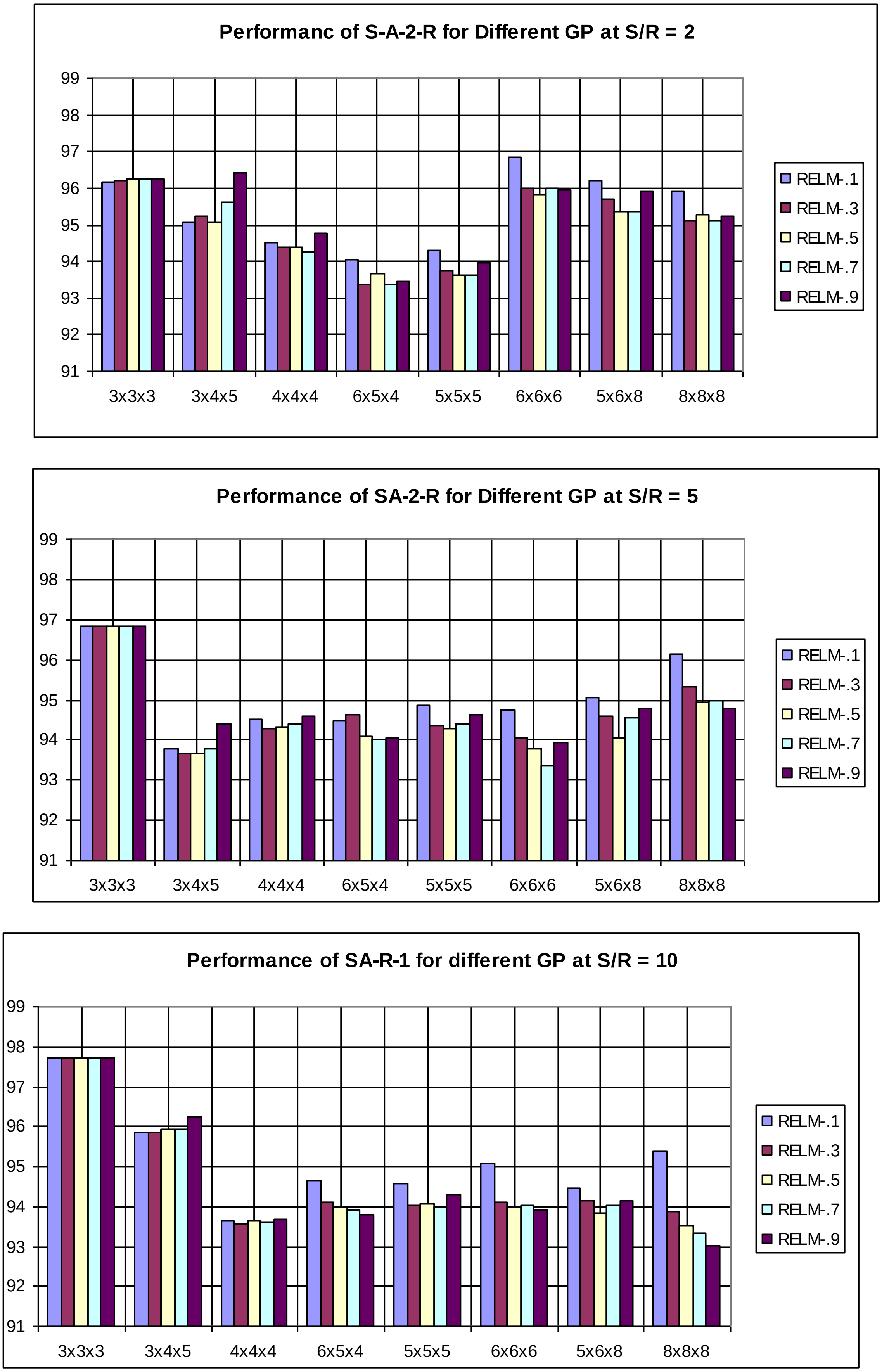


Fig.4.19 Performance of SA-M-1 for Different GP - Makespan

4.3 COMPARISON OF THE BEST HEURISTIC VERSIONS

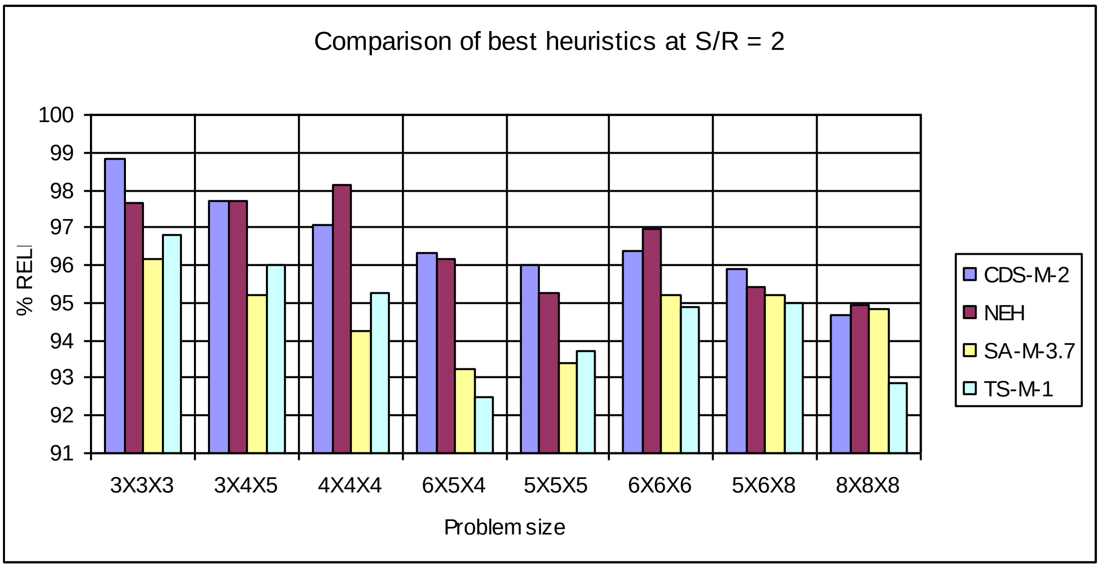
Summarizing the findings in the previous sections, it can be stated that the proposed modifications could improve the performance of the CDS, SA, and TS heuristics. The best versions are CDS-M-2, SA-M-3 (GP = 0.7) and the TS-M-1 respectively. While Rajendarn’s modification was found ineffective and the original NEH and Hitomi were preferred to their modifications. This is true for both objectives of total flow time and makespan.

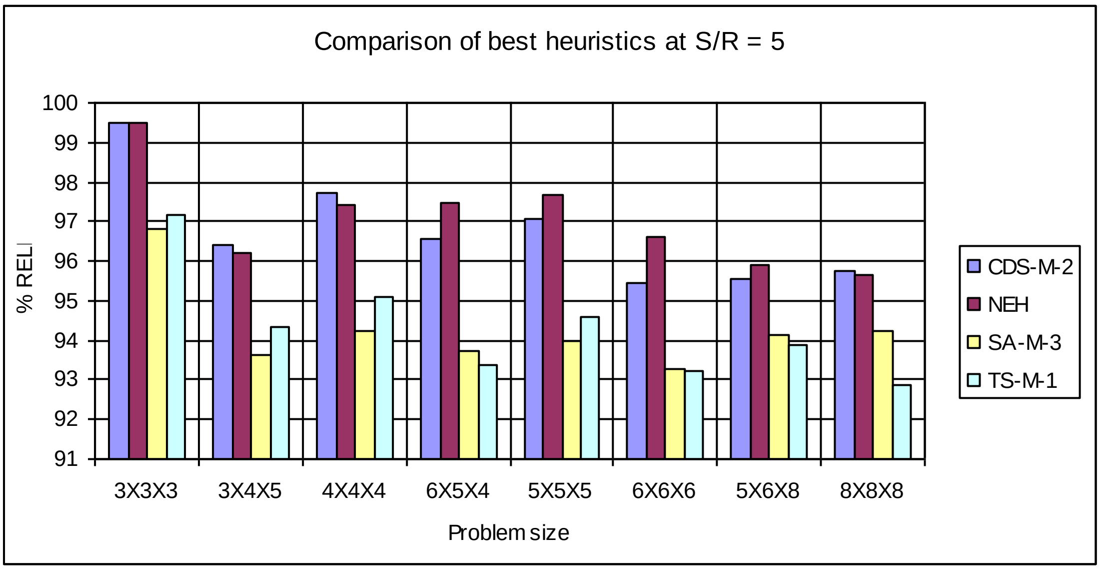
The original NEH is the best simple method for the total flow time, while CDS-M-2 is the best for makespan. AFM of NEH is better than RELM when optimizing makespan. In addition NEH performs well for optimizing total flow time associated with the worst AMF. Thus NEH seems to be more appropriate for the minimization of total flow time. Nevertheless, it is recommended to be used to minimize makespan so as to get good results with respect to the two measures of performance.

SA-M-3 at GP = 0.7 (SA-M-3.7) is better than TS-M-1 for total flow time, except for the largest problems. For makespan the two methods are approximately equivalent but TS-M-1 used to be better at the larger problems. SA-M-3.7 may be preferred for total flow time at the small and medium size problems.

Still TS-M-1 is found more stable than SA-M-3.7 as the problem size increases. It offers the possibility to redefine its components and the information included in them so that to improve its efficiency employing more relevant search based-information. Consequently TS-M-1 is considered preferable in general to SA-M-3.7.

The disadvantage of TS methods is the CPU time compared with SA methods. As can be found in Appendices, CPU time for TS methods increases





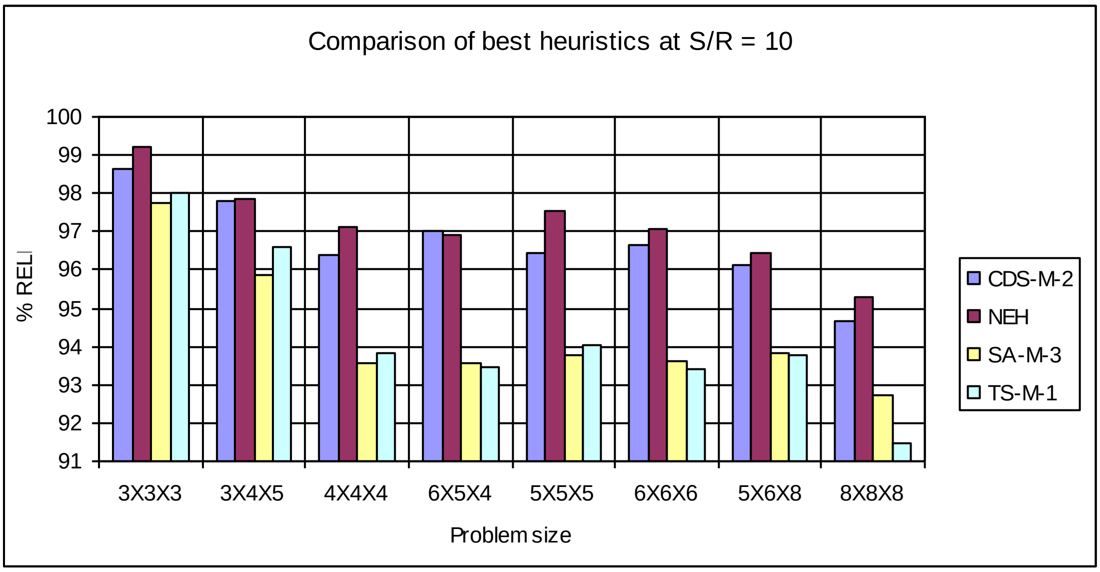
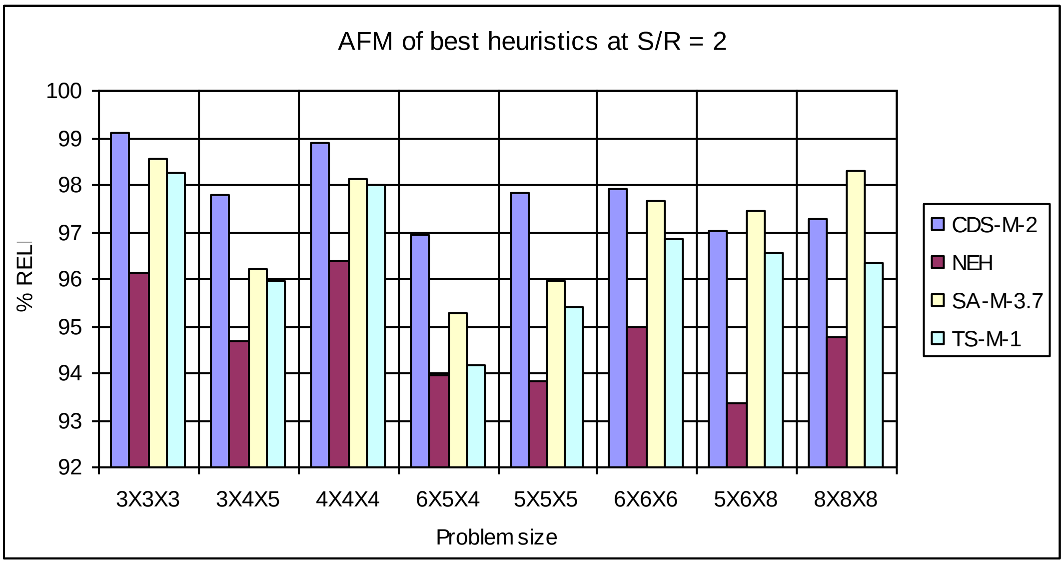
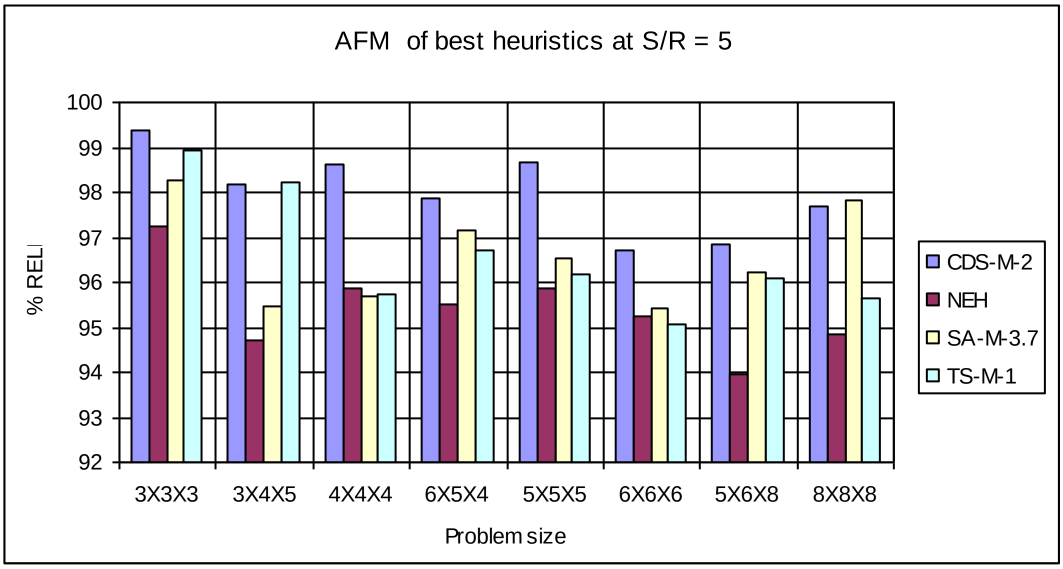


Fig.4.20 Comparison of the best heuristics - Makespan





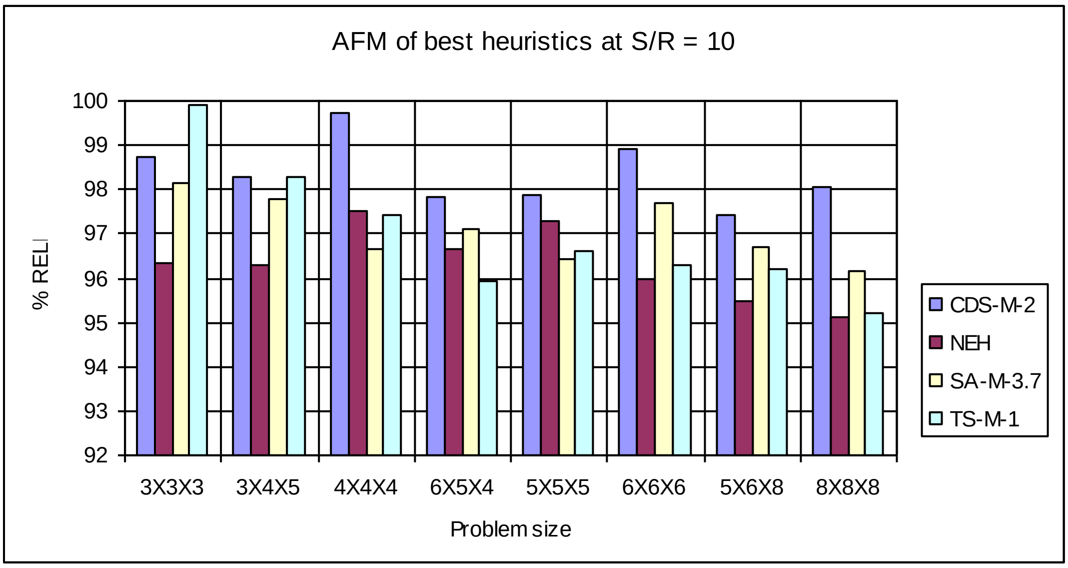


Fig.4.21 AFM with makespan from the best heuristics - Makespan

very rabidly compared to the SA methods. CPU time for TS-M-1 ranges from about 3 sec up to about 1800 sec. CPU time for SA-M-3.7 ranges from about 4.3 sec up to 75.7 sec. This is because TS uses a larger number of matrices and that the frequency of calculating the total flow time and makespan is higher than for the SA methods. For example in a 5x5x5 problem SA used the timetabling calculation 1250 times while TS uses the calculations 5100 times.

CHAPTER 5

CHAPTER 5

CASE STUDY

In order to investigate the applicability of the GS approach, a case study was conducted applying GS to a traditional batch production system that is already existing.

The machining shop (Hunger 6) in El-Nasr for Automobile Manufacturing Company, Helwan-Cairo, was chosen for the study. Hunger 6 produces all the parts (machined parts) used in the manufacture of automobiles, buses, and trucks produced in the company.

The shop consists of a large number of traditional cutting machines classified into about 105 classes. Machines include various types of lathes (center, production, heavy duty, turret, …), milling machines (universal, horizontal, vertical, gear milling, …), drills (radial, multi-spindle, special,…), and other necessary machines including slotters, presses, grinders, finishing machines ,… in variety of types and capabilities.

5.1 EXISTING OPERATING SYSTEM

The planning sector develops the yearly production plan for the company. A typical production plan, shown in Fig.5.1, shows the production year divided into four quarters, 48 weeks. Models of the automobile, bus, or truck are listed vertically. Figures in the table are the number of units required for each model in the indicated week. For example; 50 trucks of Model 1 are due during weeks 41-44. According to the plan each sector in the company develops its internal plan.

Class	Models	1st Quarter			2nd Quarter			3rd Quarter			4th Quarter				
		1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36	37-40	41-44	45-48	Total	Remarks
Trucks	Model 1	-	-	-	-	-	-	-	-	-	50	50	-	100	
	Model 2	-	10	-	6	-	-	-	-	-	40	44	-	100	
	Model 3	-	-	-	-	-	-	-	-	-	10	10	5	25	
	Model 4	-	-	-	-	-	-	-	-	-	25	25	25	75	
	Model 5	-	-	-	-	-	-	-	-	-	35	30	-	65	
	Model 6	-	4	-	-	20	4	19	-	20	6	6	5	84	
	Model 7	-	-	-	-	-	-	-	-	34	50	50	70	204	
	Total	-	14	-	6	20	4	19	-	54	216	215	105	635	
Buses	Model 1	-	-	-	12	-	-	1	-	-	-	-	-	13	
	Model 2	-	-	-	-	-	-	-	-	20	-	-	-	20	
	Model 3	-	-	-	-	-	-	-	-	-	-	-	50	50	
	Model 4	-	-	26	4	-	10	10	10	25	-	-	-	85	
	Model 5	10	-	3	-	11	15	17	12	20	20	10	10	129	
	Model 6	-	-	-	-	-	-	-	1	-	40	-	59	100	
	Model 7	-	-	-	-	-	-	-	-	-	10	5	35	50	
	Model 4	-	-	-	-	10	-	2	10	5	-	3	20	50	
	Model 5	-	-	-	-	-	-	-	-	-	20	20	10	50	
	Model 6	-	-	-	35	96	67	2	40	150	150	-	-	540	
	Model 7	-	-	-	-	-	-	-	-	-	-	100	-	100	
	Total	10	-	29	51	117	92	32	74	220	240	138	184	1187	
	Model 1	-	-	-	28	31	64	-	60	200	200	200	217	1000	
	Model 2	-	45	150	3	2	95	92	8	100	100	-	-	595	
	Total	-	45	150	31	33	159	92	68	300	300	200	217	1595	
Engines	Model 1	-	-	-											
	Model 2														
	Model 3														
	Total														
Spare parts for sale		614	1095	1242	1289	2396	2257	1633	895	895	895	895	895	15000	
Total Value (4 periods)															
Total Value (12 periods)															
Total Value (24 periods)															
Total Value (1 Year)															

In Hunger 6, the process-planning department develops the operation plans and process sheets for each part to be processed. A copy of each operation plan is released to the control room in the hunger. Control room in turn contacts the material department to deliver the required materials. The operation plan is issued for the shop floor level at the control room with a job card for each part. Then, experienced workers set the machine and begin processing to achieve the required level of accuracy.

The job card summarizes the information related to the part and the operation, including part name and number, quantity, customer, operation plan number, machine number, and the estimated times. Besides, it is a time record to follow the execution of the process sheet and to calculate the actual working hours. The process sheet is a brief summary of all the operations needed for the part, while the operation plan (See Fig.5.2) shows the setting and tooling requirements, estimated setup and processing times and the working drawing.

5.1.1 Shop Loading

Actually, jobs are assigned to machines so as to have all machines working if possible. For example if there are *m* machines of the same type and *m* parts to be processed on that type of machine, then each part would be assigned to a machine although one machine can perform all the jobs with some scheduling efforts.

5.2 SUGGESTING THE GS SYSTEM

The suggested GS system in brief, requires classifying the parts into a number of part-families. For each part family there will be a family setup, which is the common setting among the parts contained in the family. It is expected that each part will still have special setting requirements. The times for these special

Fig. 5-2 Example of an Operation Plan

requirements will be included in the part’s processing time. Machines currently used to process the parts will constitute a manufacturing cell, but machines will not be rearranged. However, flow pattern of jobs through machines will be changed to the flow line pattern (unidirectional flow).

Defining the cell and the part families, the time for setting each machine for each family is estimated. In addition, the processing times for jobs after including the special job settings will be estimated. Afterward, the system is ready to be scheduled using the GS approach.

The required information to apply the study and make the required changes are the processing times for jobs, the setup times, the setting requirements (tools, jigs, fixtures,…), and the sequence of processing of each part through the required machines. The information is supposed to be found in the operation plans and processes sheets.

5.2.1 Performing The Suggested Changes

First trial to apply the study was carried out in the gears workshop, which may be considered a separated section in Hunger 6. It was expected that parts would show similarities in terms of processing requirements. After collecting the necessary information available, the following were observed:

1. Extensive uses of heat treatment operations for most of the parts in-between the machining operations. Heat treatments take relatively very long times, besides being performed outside Hunger 6.
2. The workshop size is big and the number of parts in it is larger than what is recommendable in such a study.
3. Applying the production-flow analysis technique [29] to formulate part-families and define the related cells, there were no positive results. This means that the similarities among the parts are not in an encouraging level.

Consequently, conducting the study in the gear workshop was not fruitful, and was abandoned. A second trial was carried out in the heat treatment department. This was not also found to be the proper place for the study due to the nature of the heat treatment work.

Consequently, it was preferable to perform the study to a selected sample of parts that are processed in Hunger 6. A sample of parts was selected based on which the general guidelines of the suggested manufacturing cell and the part families it will be dedicated for would be defined. The front and rear axles of one model of truck were chosen. The axles consist of 42 parts, a list of them is shown in Table 5.1.

Table 5.1 List of part for the front and rear axles selected

No	Part No.	Qt	Part description	No	Part No.	Qt	Part description
1	III 284	4	Bush	22	V 106	5	Nut
2	III 293	4	Locking bolt	23	V 107	4	Differential pinion
3	III 302	4	Bolt	24	V 209	2	Spacer ring
4	IV 318	1	Cable lever left hand	25	V 218	2	Rear stup axle
5	IV 319	1	Cable lever right hand	26	V 219	2	Rear wheel hub
6	IV 534	4	Brake drum	27	V 222	2	Distance ring
7	IV 536	2	Rear axle shaft	28	V 225	2	Driver axle
8	V 80	1	Rear axle drive housing	29	VI 362	1	Front axle
9	V 86	1	Front cover rear axle housing	30	VI 365	2	King pin
10	V 89	1	Bevel pinion	31	VI 509	2	Stub axle
11	V 91	1	Crown wheel	32	VI 512	1	Tie rod arm
12	V 92	1	Collar nut	33	D 1568	1	Steering arm
13	V 94	1	Drive flange	34	M 528	40	Wheel stud
14	V 96	1	Bearing bush	35	M 529	2	Support
15	V 97	1	Internal ring	36	M 530	2	Thrust piece
16	V 99	1	Differential case	37	M 531	2	Hose clip
17	V 100	1	Cover differential case	38	M 532	2	Shoe brake
18	V 101	2	Friction washer	39	M 533	4	Thrust piece
19	V 102	1	Spider	40	M 539	2	Screw
20	V 103	1	Spider	41	M 573	2	Front wheel hub
21	V 104	2	Differential wheel	42	M 615	2	Cable steel

Available related data for the selected parts were collected. Table C.1 in Appendix C shows the selected parts and the machines currently used for their manufacture. Numbers in the table are indicating the technological order of operations for each part. It is observed that jobs are not flowing in the same direction. Backtracking occurs frequently. Besides that a number of successive operations for one part may be performed on the same machine.

To switch to GS it is required to reduce the number of machines needed and to remove and prevent backtracking. More important is to establish a unidirectional flow pattern for all the parts. To achieve these, parts will be reassigned to machines and machines will be exchanged as needed to carry out the required modifications.

Nine parts were excluded due to the need to heat treatments in-between the machining operations. Special operations that occur at the start or the end of the processing of a part are removed and considered as out-of-cell operations. These operations include some heat treatment, galvanizing, priming, phosphating, and sand blasting. Consequently, the list in Table C.1 could be reduced to be as shown in Table 5.2. The number of machines required was reduced to from about 180 to 73 machines through which parts (33 parts) are flowing in a unidirectional flow pattern. Based on the new situation shown in Table 5.2, new operation plans for each part were developed.

Concerning the changes and modifications made to the current job-machine relationships, the following notes are listed:

1. All changes are accepted by the process planning department. in Hunger 6.
2. The coding system of the machines in the company was found to contain errors and conflicting data about machines capabilities and being replaceable to each other. Consequently, the process of reassigning jobs among machines had to be repeated several times within the study.

Table 5.2 See Excel File

Table 5.2 See Excel File

3. Some errors were found in indicating a machine for a certain job in the operation plans. This is corrected at the shop floor level by assigning the job to the suitable machine not necessarily the one indicated in the operation plan. Consequently, timing and settings in the operation plans may not be the actual or the correct information.
4. There are no rules to estimate the setup times in the process planning department. The estimation depends on the personal judgment of the estimator.
5. The size of the cell is still large, however this is accepted since only traditional machines are present. In addition, this is the smallest number of machines based on the available information about machine replacability.
6. Parts listed Table 5-1 are not the only work that the cell will be dedicated for. These parts are a sample used to define the cell and identify the part family membership. Once the cell and part families are defined, other work can be assigned to the cell given its capabilities. New part families can be formulated so that to make full use of the cell.
7. Effect of separating the machines required on the progress of the work in Hunger 6 was neglected given that no unique machines are involved.

5.3 FORMULATING PART FAMILIES

Having the modified operations plans, the next step is to formulate the part families of the parts in consideration. In this step, parts that are processed on the same machines are checked for the existence of common settings among them at each machine. The common setup will be the family setup (major setup), and the family will consist of these parts.

Table 5.2 was studied to find out the similar parts. Parts that are processed on same machines were identified. However, it was found that similarities among parts in terms of setting requirements did not occur as expected.

Table 5.3 Common settings for parts 99V and 100V on machine 120002

	Description	**Code number**
1	Power operated chuck	KL 400
2	Setting gauge	B 9654 – 010 – N002
3	Boring bar	C 9407 – 470 – N001
4	Turning tool	S 218 – 1616 – 60 – HSS
5	Turning tool	S 166 – 1625 – 150 – P30
6	Turning tool	S 459 – 98 – 90 – K20

It was found that some parts may share only some of all the machines necessary for their processing. Further, parts processed on the same machine may not have any common settings on this machine. An example for the parts sharing only some of the necessary machines for their processing are parts 99V and 100V. Nevertheless, the common settings are not found for all the shared machines.

Both parts 99V and 100V are processed on machine 120002. Out of 15 settings for 99V and 11 settings for 100V, common settings between them are shown in Table 5.3. These settings would be the family settings for the family consisting of these two parts on machine 120002. Parts 534 IV and 573M are processed on machine 120002 as well. No settings were found common between the two parts and neither of them has common settings with 99V and 100 V.

Parts 318 IV and 319 IV are processed on the same machine for all the necessary operations. Both share the same settings on all machines. Thus, they can be considered to be a part family. However, no more parts could be appended to this part family.

Such examples of parts similarity were found very rare among the parts listed in Table 5.2. Consequently it was not possible to formulate the part families.

5.4 RESULTS OF THE CASE STUDY

The objective of the case study was to explore the possibility to apply the GS model to an existing traditional discrete parts manufacturer without formulating a cell physically. The cell and part families were supposed to be defined based on already existing operation plans. The following can be concluded from the this study:

1. Group scheduling is applicable in traditional flow shops without formulating cells physically.
2. To achieve this it is necessary to consider the GT principles in the first stages of developing the operation plans and machine loading, such that part families membership are in prospect from the beginning.
3. For the workshop considered in the study, it was found that formulating part families for group scheduling, based on an existing situation leads to unpredictable results.
4. Converting a job shop into a flow shop is possible without the rearrangement of the machines.

CHAPTER 6

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

The group scheduling (GS) model was investigated while studying the relative performance of selected simple and iterative GS heuristics in a flow line manufacturing cell that is dedicated for the processing of a number of part-families. Heuristics were modified in order to improve their performance and explore the characteristics of the GS model. A timetabling procedure that can account for the presence of zero processing times in a multi-family cell is proposed. Besides, a case study was conducted to investigate the applicability of GS to a traditional batch production system. The main conclusions and recommendation for future work are summarized as follows.

6.1 CONCLUSIONS

1. The proposed modifications to the group scheduling heuristics studied were found effective and preferable to the original formulations for the Cambell, Dudek and Smith (CDS), simulated annealing (SA), and tabu search (TS) techniques. The proposed CDS-M-2, SA-M-3 and TS-M-1 are the best performing heuristic versions each in its class.

2. The two-phase nature of group scheduling model should be considered in the group scheduling heuristic methods in order to compensate for the possible interaction between the two phases of scheduling.

3. The iterative improvement techniques are preferable to the single, and multi pass methods not only because of their superior performance but because they can handle the phases’ interaction in group scheduling as well.

4. The tabu search (TS) technique is found preferable in general to the simulated annealing (SA) technique. Tabu search is more robust when increasing problem sizes than simulated annealing. It offers the possibility to redefine its components to include more relevant search-based information thus to increase its efficiency.

5. The change-dependent acceptance probability in the simulated annealing heuristic is more efficient than the change-independent acceptance probability. Moreover, simulated annealing technique needs to incorporate some form of control over the effect of the random numbers in its behaviour.

6. The possibility of the zero processing times should be considered during timetabling calculations. It does not seem effective to consider the presence of the zero processing times in the structure of the scheduling heuristic. Rajendran's modification [18,30] adopted in Hitomi and NEH was found ineffective.

7. The proposed timetabling procedure for the multi-family manufacturing cells was shown to be effective in compensating for the consequences of the presence of the zero processing times and providing more realistic timetabling information.

8. Due to the presence of zero processing times, it may not be always correct to define makespan as the time span from the start of the first job on the first machine to the finish of the last job on the last machine. Instead it is defined as the largest completion time. Makespan is not necessarily associated with the last job in schedule, or the last machine.

9. The group scheduling approach is applicable in traditional flow shops without formulating the manufacturing cells physically.

6.2 RECOMMENDATION FOR FUTURE WORK

1. Studying the group scheduling heuristic performance for different cell parameters other than the setup to run time ratio (S/R), and for other practical problem formulations.

2. Applying the group scheduling technique adopted in the research to practical manufacturing cells and flow shops.

REFERENCES

REFERENCES

1. Dorlet J., ”Scheduling factories of the future”, 2nd CIRP conf. on new manufacturing tech., TN, USA, Jun 1-2, 1989.

2. Brown, Harhen, Shirnan, "Production Management: CIM Prospective", 1988.

3. Hitomi K., ”Manufacturing Systems Engineering”, 2nd Ed., Taylor & Francis, 1996, UK.

4. Cusumano M. A., "Manufacturing Innovation: Lessons from the Japanese Auto Industry”, Sloan Management Review, Vol. 30, No. 1, Fall 1988.

5. Sadowski, Harmonosky, “Production and Project Scheduling”, Chap. 3.9, Production HB. Editor: White J. A., 4th Ed., John Weliy and Sons 1987.

6. Monden Y., "How Toyota Shortened Supply Lot production, waiting time and conveyance time", IE, Sept. 1981.

7. Blazewicz, Ecker, Schmidt, "Scheduling in Computers and Manufacturing Systems", Springer Verlag, Germany 1993.

8. Morton, Pentico, "Heuristic Scheduling Systems", John Wiley and Sons, Canada, 1993.

9. Darrow, Jatindr, Gupta, "Integrating GT and MRP Systems Through Lot Sizing and Scheduling", Computers and Industrial Engineering, Vol. 16, No. 2, 1989.

10. Rajiv, Gupta, Tompkins, ”An examination of the dynamic behaviour of part families in GT”, Int. J. of Prod. Res., Vol. 20, No. 1, 1982.

11. Hitomi K., Ham, Yoshida T., “Group Technology”, Kluwer-Nijhoff, Hingham, MA, 1985.

12. Kapov, Vakharia A. J., “Scheduling a flow-line manufacturing cell : a tabu search approach”, Int. J. of Prod. Res., Vol. 31, No. 7, 1993.

13. Hitomi K., “Group scheduling with interactive computer graphics”, Memories of the Faculty of Eng., Kyoto Univ., Vol. 50, No. 3, 1988.

14. Mahmoodi, Dooley K. J., “A Comparison of exhaustive and non-exhaustive group scheduling heuristics in a manufacturing cell”, Int. J. of Prod. Res., Vol. 29, No. 9, 1991.

15. Vakharia A. J., Chang Y., “A Simulated Annealing Approach to Scheduling a Manufacturing Cell“, Naval Research Logistics, Vol. 37, 1990, pp. 559-577.

16. Warren, Boe, Cheng, “A close neighbor algorithm for designing cellular manufacturing systems”, Int. J. of Prod. Res., Vol. 29, No. 10,1991.

17. Sridhar J., Rajendran C., ”Scheduling in a cellular manufacturing system: a simulated annealing approach”, Int. J. of Prod. Res., Vol. 31, No. 3, 1993.

18. Sridhar J., Rajendran C., “A Genetic Algorithm for Family and Job Scheduling in a Flowline-Based Manufacturing Cell“, Computers & Industrial Engineering, Vol. 27, No. 1-4, 1994, pp. 469-472.

19. Allison J. D., “Combining Petrov’s Heuristic and The CDS Heuristic in a Group Scheduling Problems“, Computers & Industrial Engineering, Vol. 23, 1990.

20. Cleland D. I., Bidanda B., “The Automated Factory HB”, TAB Books, 1st Ed. 1990.

21. Grasso, Masanta, Piacentini, "Heuristic procedures for group scheduling"; 4th Cairo Univ. MDP Conf., Dec. 1988, pp. 629-636.

22. Ruben R. A., Mosier C. T., Mahmoodi F.,”A comprehensive analysis of group scheduling heuristics in a job shop cell”, Int. J. of Prod. Res., Vol. 31, No. 6, 1993.

23. Wemmerlov U., Vakharia A. J., “Job and Family Scheduling of a Flow-Line Manufacturing Cell: A Simulation Study”, IIE Transactions, Vol. 23, No. 4, 1991.

24. Etman M. I., ”Computerized Production Control ”, Unpublished Master thesis, Alexandria Univ., 1976.

25. Inst. of Ind. Engineers, “Industrial Engineering Terminology”, According to ANSI standard Z94.10, Ch.10; Production Planning and Control, 1983.

26. Baker, K. R., “Introduction to sequencing and scheduling”, John Wiely & Sons, New York, 1974.

27. MacCarthy B. L., Liu J., “Addressing the gap in scheduling research: a review of optimization and heuristic methods in production control” , Int. J. of Prod. Res., Vol. 31, No. 1, 1993.

28. Abd El-Aal R. S., "A quick heuristic approach to solve n job, m machine flow shop problem", 4th Cairo Univ. MDP Conf., Dec. 1988, pp. 621-628.

29. Gallagher, Knight, "Group technology production methods in manufacture", Ellis Horwood Ltd., England, 1986.

30. Rajendran C., "A heuristic for scheduling in flow shop and flowline-based manufacturing cell with multi criteria", Int. J. of Prod. Res., Vol. 32, No. 11, 1994.

31. Cambell, Dudek, Smith, "A heuristic for the n Job, m machine sequencing problem", Management Science, Vol. 16, No. 10, June 1970.

32. Park, Pegden, Enscore, "A survey and evaluation of static flow shop scheduling heuristics", Int. J. of Prod. Res., Vol. 22, No. 1, 1984.

33. Schuur P. C., “Classification of acceptance criteria for the simulated annealing algorithm”, Mathematics of Operations Research, Vol. 22, No. 2, May 1997.

34. Kirkpatrick S., Gelatt C. D., Vecchi M. P., “Optimization by simulated annealing”, Science, Vol. 220, No. 4598, 13 May 1983.

35. Glover F., “Tabu search - Part I”, ORSA Journal of Computing, Vol. 1, No. 3, 1989, pp. 190-206.

36. Glover F., Hubscher R., “Applying tabu search with influential diversification to multiprocessor scheduling”, Computer and Operation Research, Vol. 21, No. 9, 1994, pp. 877-884.

37. Glover, F., “Tabu search - Part II”, ORSA Journal of Computing, Vol. 2, No. 1, 1990, pp. 4-32.

APPENDICES

APPENDIX A

RESULTS WITH RESPECT TO TOTAL FLOW TIME

Tables A.1 through A.8 shows the results of solving the experimental group scheduling problems using the heuristics under study, with respect to the total flow time, a table for each problem size. The table is divided into three parts vertically one for each S/R ratio. For each heuristic at each S/R, the relative total flow time (RELF) is listed in the first column. Relative makespan for the solution (AMF) is given in the second column and the third column exhibits the computational times (CPU) in seconds.

The iterative improvement methods in the tables of results are named as in the following table for all tables in both appendices A and B, and in Chapter 4 as well. X and R are for using a random initial solution, H for using a relatively goo initial solution generated by Hitomi. GP takes values of 0.1, 0.3, 0.5, 0.7, and 0.9 in SA methods in the study of the effect of the GP factor.

Heuristic	Name	Heuristic	Name
Original Hitomi	HITOMI	Original Tabu	TABU-X
Hitomi-Mod	HIT-M	Tabu-Mod-1	T-X-M-1
		Tabu-Mod-2	T-X-M-2
Original CDS	CDS		
CDS-Mod-1	CDS-M-1	Original SA	SA-X-GP
CDS-Mod-2	CDS-M-2	SA-Mod-1	SA-X-1GP
CDS-Mod-3	CDS-M-3	SA-Mod-2	SA-X-2GP
		SA-Mod-3	SA-X-3GP
Original NEH	NEH		
NEH-Mod-1	NEH-M-1		

Table A.1 Results for 3x3x3-problem size with total flow time as the performance criterion

3 x 3 x 3 Problems									
	S/R = 2			**S/R = 5**			**S/R = 10**		
Heuristic	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	.009	100.00	100.00	.011	100.00	100.00	.009
HIT-M	100.09	100.29	.007	100.77	100.44	.009	100.32	100.28	.009
CDS	98.997	99.024	.038	98.970	100.00	.033	98.523	99.049	.029
CDS-M-1	98.661	99.060	.033	98.672	99.794	.035	98.340	99.049	.031
CDS-M-2	98.966	99.024	.046	98.890	99.840	.042	98.427	99.076	.046
CDS-M-3	98.991	98.951	.024	98.791	99.634	.049	98.229	98.897	.046
NEH	92.378	103.65	.040	93.385	102.20	.046	94.341	101.79	.042
NEH-M-1	92.155	103.40	.046	93.449	101.92	.040	94.316	101.81	.037
TABU-R	90.930	100.90	3.018	91.576	100.50	3.050	92.489	99.890	2.978
T-R-M-1	91.857	101.92	2.948	92.041	100.94	2.953	92.378	99.835	2.890
T-R-M-2	91.464	101.81	2.964	92.101	100.78	2.957	92.465	100.30	2.948
TABU-H	90.943	100.69	2.988	91.902	100.69	2.890	92.253	99.917	2.987
T-H-M-1	91.603	100.43	2.883	91.763	100.57	2.887	92.473	99.848	2.894
T-H-M-2	91.667	100.18	2.876	91.794	100.92	2.909	92.480	99.835	2.944
SA-R-.1	91.292	102.86	4.354	93.202	100.18	4.300	92.159	99.573	4.354
SA-R-.3	91.051	102.42	4.353	91.544	100.55	4.278	92.115	99.766	4.344
SA-R-.5	90.067	101.19	4.333	91.464	100.94	4.270	92.164	99.628	4.325
SA-R-.7	90.645	101.77	4.315	91.333	100.30	4.262	92.391	99.848	4.312
SA-R-.9	90.245	101.59	4.297	91.289	100.69	4.252	92.130	99.890	4.299
SA-R-1.1	93.317	103.22	4.400	91.588	100.73	4.331	92.917	99.642	4.381
SA-R-1.3	90.651	102.86	4.387	91.393	100.89	4.324	92.253	100.10	4.372
SA-R-1.5	91.806	101.92	4.369	91.373	100.64	4.315	92.106	99.683	4.363
SA-R-1.7	89.985	101.09	4.355	91.245	100.71	4.293	92.072	99.807	4.357
SA-R-1.9	90.937	101.95	4.338	91.321	100.55	4.288	92.111	99.986	4.341
SA-R-2.1	90.099	101.37	4.343	91.237	100.55	4.288	92.226	99.614	4.337
SA-R-2.3	90.321	101.74	4.344	91.226	100.55	4.277	92.067	99.862	4.326
SA-R-2.5	89.972	101.30	4.328	91.226	100.57	4.266	92.067	99.711	4.321
SA-R-2.7	89.978	101.66	4.321	91.241	100.55	4.253	92.067	99.711	4.305
SA-R-2.9	90.042	101.77	4.307	91.345	100.76	4.238	92.111	99.711	4.278
SA-R-3.1	90.664	101.66	4.383	91.345	100.57	4.318	92.067	99.766	4.376
SA-R-3.3	89.972	101.45	4.382	91.226	100.53	4.315	92.067	99.766	4.366
SA-R-3.5	89.972	101.30	4.363	91.226	100.55	4.294	92.067	99.724	4.351
SA-R-3.7	89.978	101.37	4.357	91.233	100.55	4.302	92.067	99.724	4.343
SA-R-3.9	90.188	101.66	4.340	91.257	100.48	4.277	92.101	99.793	4.332
SA-H-.1	92.162	101.88	4.385	91.719	100.64	4.318	92.977	100.15	4.370
SA-H-.3	90.353	101.95	4.341	91.468	100.57	4.308	92.067	99.711	4.351
SA-H-.5	90.467	101.30	4.342	91.548	100.99	4.275	92.072	99.779	4.330
SA-H-.7	90.143	101.19	4.325	91.345	100.85	4.258	92.077	99.793	4.309
SA-H-.9	90.569	102.46	4.305	91.568	100.44	4.243	92.089	99.766	4.293
SA-H-1.1	91.711	101.99	4.427	91.918	100.89	4.257	92.292	99.917	4.420
SA-H-1.3	90.226	102.13	4.410	91.397	100.71	4.347	92.243	99.835	4.412
SA-H-1.5	90.283	101.30	4.399	91.691	100.55	4.335	92.072	99.710	4.382
SA-H-1.7	90.042	101.09	4.382	91.229	100.53	4.312	92.772	102.42	4.373
SA-H-1.9	90.772	102.42	4.372	91.301	100.96	4.310	92.098	99.848	4.357
SA-H-2.1	89.991	101.37	4.384	91.226	100.71	4.328	92.103	99.683	4.375
SA-H-2.3	89.972	101.37	4.368	91.230	100.41	4.300	92.067	99.779	4.356
SA-H-2.5	89.972	101.66	4.358	91.233	100.62	4.289	92.067	99.779	4.346
SA-H-2.7	89.978	101.66	4.334	91.226	100.57	4.277	92.072	99.779	4.323
SA-H-2.9	90.036	101.74	4.326	91.301	100.80	4.258	92.106	99.766	4.309
SA-H-3.1	90.112	101.16	4.409	91.225	100.39	4.359	92.067	99.793	4.402
SA-H-3.3	89.972	101.45	4.407	91.225	100.39	4.338	92.067	99.710	4.389
SA-H-3.5	89.972	101.30	4.381	91.397	100.76	4.338	92.067	99.793	4.389
SA-H-3.7	89.972	101.16	4.378	91.229	100.53	4.313	92.072	99.724	4.361
SA-H-3.9	90.029	101.48	4.371	91.289	100.73	4.302	92.101	99.848	4.358

Table A.2 Results for 3x4x5-problem size with total flow time as the performance criterion

3 x 4 x 5 Problems									
	S/R = 2			S/R = 5			S/R = 10		
Heuristic	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	.021	100.00	100.00	.020	100.00	100.00	.021
HIT-M	100.53	101.51	.018	100.62	100.64	.018	100.03	100.42	.016
CDS	97.613	99.643	.101	97.082	97.636	.100	97.307	98.131	.103
CDS-M-1	97.340	99.770	.102	96.905	98.027	.103	97.348	98.210	.103
CDS-M-2	96.196	98.621	.167	96.430	96.676	.157	96.920	98.457	.141
CDS-M-3	96.132	98.672	.165	96.939	97.387	.150	97.159	98.637	.152
NEH	92.462	103.19	.217	92.531	100.12	.216	93.784	101.53	.209
NEH-M-1	92.470	103.58	.215	92.386	99.893	.207	93.829	101.61	.214
TABU-R	91.072	102.04	17.509	92.042	97.653	16.439	92.989	100.15	16.317
T-R-M-1	91.819	101.94	16.074	92.474	98.507	14.985	92.725	99.257	16.038
T-R-M-2	91.980	102.40	16.507	92.828	98.187	14.506	93.335	99.786	15.840
TABU-H	91.664	101.30	16.691	92.166	98.169	15.570	92.550	99.426	15.741
T-H-M-1	92.425	100.38	15.253	92.181	97.547	15.015	92.714	98.547	15.146
T-H-M-2	92.304	99.821	15.418	92.317	98.133	14.949	92.751	98.626	15.226
SA-R-.1	93.319	102.43	9.136	93.031	99.218	9.107	93.907	99.572	9.059
SA-R-.3	93.603	102.91	9.117	91.063	97.404	9.083	92.215	99.258	9.048
SA-R-.5	91.125	100.97	9.101	91.751	98.471	9.083	92.081	99.662	9.030
SA-R-.7	90.847	101.20	9.071	91.559	97.991	9.061	92.050	99.640	9.014
SA-R-.9	91.712	102.43	9.066	92.149	98.009	9.047	92.369	99.122	8.984
SA-R-1.1	93.807	103.35	9.218	92.580	98.240	9.193	93.429	99.324	9.155
SA-R-1.3	91.830	101.30	9.196	91.456	97.653	9.170	93.105	99.809	9.131
SA-R-1.5	92.427	103.01	9.187	91.805	97.173	9.152	92.326	99.054	9.116
SA-R-1.7	91.417	101.23	9.151	92.341	97.973	9.125	92.864	100.06	9.079
SA-R-1.9	92.829	102.25	9.124	91.901	98.667	9.098	92.622	99.651	9.065
SA-R-2.1	89.992	101.20	9.089	90.580	97.404	9.068	91.634	99.043	9.029
SA-R-2.3	89.740	100.54	9.089	90.447	97.173	9.063	91.528	98.930	9.013
SA-R-2.5	89.944	100.46	9.080	90.458	97.102	9.050	91.683	99.099	9.012
SA-R-2.7	90.279	100.54	9.087	90.746	96.711	9.054	91.792	98.874	9.006
SA-R-2.9	91.693	101.92	9.063	91.366	96.996	9.041	92.273	99.065	9.001
SA-R-3.1	89.861	99.770	9.138	90.571	96.996	9.114	91.838	99.009	9.079
SA-R-3.3	89.478	101.10	9.138	90.415	97.262	9.123	91.593	98.806	9.071
SA-R-3.5	89.722	100.87	9.138	90.560	97.476	9.108	91.547	99.268	9.060
SA-R-3.7	89.971	102.12	9.140	90.838	97.760	9.111	91.790	98.919	9.064
SA-R-3.9	91.533	100.82	9.136	91.383	97.884	9.102	92.223	98.896	9.070
SA-H-.1	94.026	101.58	9.198	93.376	98.347	9.187	93.595	100.65	9.131
SA-H-.3	92.323	101.74	9.166	91.298	97.636	9.147	92.328	99.200	9.111
SA-H-.5	91.366	100.20	9.151	91.390	97.813	9.116	92.314	99.899	9.072
SA-H-.7	91.629	101.40	9.119	91.381	97.404	9.089	92.336	99.291	9.052
SA-H-.9	91.964	101.18	9.080	91.976	98.169	9.059	92.660	99.764	9.017
SA-H-1.1	93.376	101.51	9.303	93.064	98.382	9.270	93.523	98.975	9.222
SA-H-1.3	91.348	101.66	9.270	91.512	97.387	9.238	92.600	98.885	9.198
SA-H-1.5	91.147	101.51	9.245	91.176	97.404	9.226	92.145	99.077	9.171
SA-H-1.7	91.508	101.58	9.220	91.601	97.547	9.196	92.332	99.752	9.152
SA-H-1.9	91.841	101.81	9.189	91.834	97.476	9.168	92.451	99.279	9.131
SA-H-2.1	90.531	101.51	9.123	90.793	97.209	9.111	91.733	98.998	9.063
SA-H-2.3	89.949	100.31	9.123	90.659	96.853	9.089	91.577	98.446	9.050
SA-H-2.5	89.786	100.74	9.107	90.483	97.724	9.083	91.573	99.099	9.043
SA-H-2.7	90.065	101.12	9.092	90.601	96.658	9.064	91.618	99.122	9.032
SA-H-2.9	91.468	101.18	9.086	91.680	97.511	9.048	92.157	99.133	9.012
SA-H-3.1	91.101	101.18	9.207	90.823	96.942	9.176	91.645	98.930	9.136
SA-H-3.3	89.719	100.08	9.195	90.573	97.707	9.170	91.596	98.480	9.127
SA-H-3.5	89.478	100.26	9.192	90.377	97.084	9.184	91.548	99.133	9.122
SA-H-3.7	89.765	101.05	9.187	90.802	97.102	9.167	91.924	99.178	9.119
SA-H-3.9	90.957	101.05	9.177	91.294	96.978	9.162	92.218	98.716	9.115

Table A.3 Results for 4x4x4-problem size with total flow time as the performance criterion

4 x 4 x 4 Problems									
	S/R = 2			**S/R = 5**			**S/R = 10**		
Heuristic	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	.022	100.00	100.00	.020	100.00	100.00	.018
HIT-M	100.65	100.84	.020	100.52	100.59	.020	100.07	100.20	.018
CDS	98.510	99.509	.118	98.376	100.00	.119	97.751	98.536	.110
CDS-M-1	99.291	100.47	.121	98.180	99.985	.119	97.718	98.619	.122
CDS-M-2	96.784	98.224	.198	97.171	98.886	.188	97.099	97.210	.191
CDS-M-3	97.094	98.411	.197	97.097	98.827	.200	97.155	97.201	.182
NEH	93.851	102.95	.192	94.464	102.82	.188	93.766	99.602	.192
NEH-M-1	93.695	102.67	.191	94.338	102.48	.195	93.785	99.602	.194
TABU-R	93.452	100.91	17.932	92.651	99.076	17.492	91.581	97.535	16.943
T-R-M-1	93.683	101.15	18.045	92.748	99.516	17.514	91.541	97.692	16.042
T-R-M-2	93.115	99.766	18.750	92.407	98.812	17.422	91.699	98.082	16.261
TABU-H	93.586	99.790	16.466	92.306	98.739	17.508	91.348	96.738	16.148
T-H-M-1	92.834	98.598	16.250	92.060	98.079	17.208	91.220	97.127	16.187
T-H-M-2	93.094	99.159	16.768	92.501	97.932	17.027	91.511	97.739	16.492
SA-R-.1	95.820	101.89	9.896	93.487	98.974	9.974	91.280	96.589	9.950
SA-R-.3	92.841	100.30	9.873	93.077	99.487	9.965	90.926	96.738	9.923
SA-R-.5	92.747	99.953	9.849	91.943	98.666	9.937	91.117	97.349	9.910
SA-R-.7	92.053	99.883	9.828	91.476	98.446	9.912	90.567	96.589	9.890
SA-R-.9	92.445	100.09	9.811	92.122	98.402	9.894	90.903	96.608	9.857
SA-R-1.1	95.885	102.10	9.982	93.191	99.281	10.07	91.645	97.461	10.04
SA-R-1.3	94.702	101.96	9.958	92.006	99.296	10.06	90.746	96.145	10.02
SA-R-1.5	92.555	99.813	9.937	91.936	98.930	10.02	90.868	97.127	10.00
SA-R-1.7	92.808	100.54	9.918	91.527	98.270	10.01	90.586	97.192	9.975
SA-R-1.9	92.731	100.02	9.901	91.608	98.504	9.985	90.741	96.636	9.958
SA-R-2.1	93.195	100.98	9.833	91.579	98.138	9.915	90.358	97.044	9.893
SA-R-2.3	91.459	99.252	9.826	91.226	98.108	9.918	90.183	96.210	9.889
SA-R-2.5	91.411	100.16	9.807	90.949	98.108	9.897	90.155	96.534	9.872
SA-R-2.7	91.945	100.70	9.792	90.882	97.214	9.884	90.218	96.478	9.855
SA-R-2.9	92.019	99.813	9.799	91.378	97.874	9.876	90.623	96.868	9.848
SA-R-3.1	92.570	101.43	9.894	91.817	98.812	9.977	90.341	96.691	9.961
SA-R-3.3	91.957	100.33	9.881	91.151	98.211	9.967	90.130	96.849	9.950
SA-R-3.5	91.521	100.35	9.880	90.816	97.859	9.966	90.144	96.571	9.949
SA-R-3.7	91.211	99.346	9.869	90.895	97.536	9.952	90.276	96.589	9.925
SA-R-3.9	92.353	98.995	9.855	91.419	98.358	9.947	90.573	96.747	9.917
SA-H-.1	96.139	100.94	9.938	93.803	99.736	10.03	91.558	97.442	10.00
SA-H-.3	94.471	100.73	9.910	92.824	99.091	10.00	91.043	97.377	9.976
SA-H-.5	93.781	101.47	9.879	92.729	98.812	9.972	90.726	96.358	9.940
SA-H-.7	92.041	100.80	9.837	91.778	98.578	9.931	90.731	96.552	9.901
SA-H-.9	92.526	100.35	9.816	91.772	98.167	9.895	90.747	96.923	9.870
SA-H-1.1	95.421	101.87	10.03	94.103	99.839	10.12	91.571	97.924	10.10
SA-H-1.3	94.040	101.36	10.01	92.731	99.223	10.11	91.178	96.784	10.08
SA-H-1.5	93.007	100.02	10.00	91.532	98.284	10.09	90.657	96.664	10.05
SA-H-1.7	92.716	100.21	9.978	91.545	97.903	10.08	90.509	96.96	10.05
SA-H-1.9	92.817	99.720	9.961	91.958	98.402	10.06	90.744	96.654	10.03
SA-H-2.1	91.904	100.35	9.837	91.516	98.754	9.929	90.491	96.506	9.900
SA-H-2.3	92.298	99.509	9.817	91.068	97.903	9.910	90.208	96.552	9.880
SA-H-2.5	91.793	100.47	9.808	91.500	98.798	9.892	90.200	96.905	9.874
SA-H-2.7	91.452	99.462	9.793	90.889	97.786	9.872	90.281	96.552	9.863
SA-H-2.9	91.957	100.16	9.766	91.517	97.903	9.863	90.493	96.608	9.829
SA-H-3.1	92.296	100.40	9.969	91.160	97.786	10.05	90.497	96.219	10.04
SA-H-3.3	92.017	100.19	9.944	90.963	98.182	10.04	90.209	96.673	10.02
SA-H-3.5	91.469	99.696	9.960	90.816	98.314	10.05	90.114	96.821	10.02
SA-H-3.7	91.716	99.603	9.946	91.083	98.138	10.05	90.309	97.034	10.02
SA-H-3.9	91.844	99.790	9.952	91.206	97.507	10.04	90.430	97.053	9.999

Table A.4 Results for 6x5x4-problem size with total flow time as the performance criterion

6 x 5 x 4 Problems									
Heuristic	S/R = 2			S/R = 5			S/R = 10		
	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	.035	100.00	100.00	.035	100.00	100.00	.033
HIT-M	100.44	100.82	.036	100.52	100.78	.037	100.00	100.07	.035
CDS	96.944	98.726	.339	97.881	99.213	.334	98.467	99.688	.332
CDS-M-1	96.865	99.071	.341	97.663	99.374	.345	98.365	99.748	.343
CDS-M-2	95.954	97.497	.584	96.215	97.648	.661	96.726	97.949	.591
CDS-M-3	95.452	97.497	.648	96.173	97.820	.624	96.573	97.991	.622
NEH	91.621	100.38	.493	94.508	100.90	.498	94.484	100.44	.497
NEH-M-1	91.517	100.54	.496	94.358	100.80	.497	94.485	100.37	.502
TABU-R	90.542	99.176	78.168	91.356	98.678	76.910	91.403	97.175	76.397
T-R-M-1	90.169	98.472	76.581	90.732	97.809	75.335	91.070	97.235	77.175
T-R-M-2	90.041	98.771	82.929	91.141	98.456	75.474	91.342	97.038	73.839
TABU-H	90.199	97.857	76.235	91.248	98.688	75.677	91.661	96.936	75.644
T-H-M-1	89.511	97.467	75.477	90.796	97.870	74.118	91.059	96.258	75.036
T-H-M-2	89.681	97.557	75.155	91.025	97.698	74.808	91.189	96.072	76.857
SA-R-.1	93.236	100.06	18.390	94.122	100.71	18.434	92.932	98.603	18.421
SA-R-.3	91.754	99.266	18.346	93.008	99.970	18.402	92.457	98.171	18.396
SA-R-.5	90.807	99.371	18.310	91.606	98.274	18.377	91.600	97.247	18.265
SA-R-.7	89.896	98.412	18.276	91.761	98.526	18.321	91.433	96.630	18.335
SA-R-.9	90.386	98.951	18.236	90.967	98.112	18.288	91.463	97.121	18.288
SA-R-1.1	92.055	98.576	18.521	93.078	99.859	18.587	93.076	98.291	18.584
SA-R-1.3	91.800	99.191	18.487	91.801	99.071	18.541	91.818	96.840	18.548
SA-R-1.5	91.313	99.146	18.462	91.592	98.021	18.510	91.603	96.852	18.515
SA-R-1.7	90.194	99.176	18.429	91.193	98.314	18.492	91.657	96.726	18.482
SA-R-1.9	90.546	99.116	18.380	91.351	98.385	18.456	91.018	96.882	18.434
SA-R-2.1	91.388	100.18	18.243	91.525	98.355	18.298	91.501	96.114	18.293
SA-R-2.3	89.421	98.292	18.227	91.393	98.294	18.288	90.976	97.385	18.270
SA-R-2.5	89.877	98.681	18.204	90.411	98.516	18.253	90.366	96.396	18.265
SA-R-2.7	88.723	97.467	18.190	90.425	97.840	18.241	90.599	96.900	18.235
SA-R-2.9	89.412	97.363	18.154	90.457	97.840	18.213	90.526	96.474	18.209
SA-R-3.1	91.293	100.12	18.279	91.615	98.355	18.350	91.856	96.978	18.350
SA-R-3.3	90.626	98.591	18.285	90.656	97.749	18.326	90.524	96.708	18.334
SA-R-3.5	89.191	97.767	18.273	90.422	98.072	18.320	90.622	96.348	18.325
SA-R-3.7	89.369	98.412	18.262	90.208	97.628	18.321	90.691	96.492	18.322
SA-R-3.9	89.373	98.217	18.236	90.286	98.092	18.313	90.523	96.354	18.292
SA-H-.1	93.021	99.521	18.405	94.180	99.960	18.459	93.548	97.583	18.453
SA-H-.3	91.624	99.685	18.360	92.620	99.647	18.399	91.703	97.080	18.413
SA-H-.5	92.154	100.64	18.315	92.618	99.273	18.376	91.569	97.403	18.373
SA-H-.7	91.126	99.595	18.271	92.012	97.961	18.337	91.578	96.894	18.327
SA-H-.9	90.188	98.546	18.233	91.336	98.274	18.285	91.279	96.846	18.292
SA-H-1.1	93.407	101.21	16.623	93.039	100.29	18.680	93.172	98.111	18.673
SA-H-1.3	92.320	98.531	18.603	92.110	99.243	18.651	91.852	96.924	18.651
SA-H-1.5	91.243	99.640	18.576	91.379	98.960	18.636	91.594	96.714	18.636
SA-H-1.7	90.077	97.962	18.556	91.727	98.456	18.611	91.359	97.247	18.606
SA-H-1.9	90.400	98.307	18.532	91.183	98.627	18.593	91.212	96.750	18.598
SA-H-2.1	90.666	98.382	18.313	91.391	98.910	18.365	92.203	97.859	18.372
SA-H-2.3	90.227	98.996	18.277	90.642	98.233	18.342	91.060	96.888	18.336
SA-H-2.5	90.370	98.307	18.277	90.067	97.820	18.315	90.620	96.534	18.320
SA-H-2.7	88.784	97.422	18.248	90.171	98.425	18.312	90.432	96.018	18.298
SA-H-2.9	89.495	98.472	18.225	90.279	98.183	18.283	90.453	96.288	18.279
SA-H-3.1	90.507	98.472	18.491	92.050	98.617	18.548	91.939	96.948	18.543
SA-H-3.3	89.518	98.142	18.470	90.361	97.416	18.536	90.738	96.840	18.538
SA-H-3.5	89.307	97.363	18.473	90.039	97.759	18.523	90.853	97.002	18.508
SA-H-3.7	89.148	98.606	18.457	90.421	97.890	18.522	90.334	95.856	18.504
SA-H-3.9	89.719	98.262	18.448	90.352	98.355	18.504	90.500	95.460	18.498

Table A.5 Results for 5x5x5-problem size with total flow time as the performance criterion

5 x 5 x 5 Problems									
	S/R = 2			**S/R = 5**			**S/R = 10**		
Heuristic	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	.036	100.00	100.00	.042	100.00	100.00	.035
HIT-M	100.70	100.95	.038	100.62	100.68	.037	100.18	100.27	.036
CDS	98.384	99.669	.315	97.125	98.819	.319	97.445	98.628	.326
CDS-M-1	98.675	100.46	.325	97.359	99.356	.328	97.706	98.885	.329
CDS-M-2	96.444	97.317	.602	96.193	98.336	.565	95.754	97.631	.577
CDS-M-3	96.479	97.459	.659	96.305	98.594	.587	95.658	97.423	.621
NEH	93.625	100.68	.650	93.102	99.871	.659	95.049	99.813	.658
NEH-M-1	93.975	101.07	.650	93.193	99.818	.657	95.044	99.855	.660
TABU-R	92.903	99.621	79.601	91.834	97.950	78.089	92.217	98.033	72.328
T-R-M-1	92.308	98.706	75.291	91.483	98.035	77.501	91.611	98.171	73.197
T-R-M-2	92.576	98.153	77.447	91.906	98.615	75.234	91.746	98.046	73.788
TABU-H	92.158	98.185	75.111	92.014	98.476	75.835	92.048	98.095	77.674
T-H-M-1	91.695	97.522	74.595	91.432	97.842	75.004	91.746	97.839	77.951
T-H-M-2	91.886	97.317	73.709	91.804	97.778	73.892	91.862	98.060	77.849
SA-R-.1	95.827	99.684	18.865	94.819	100.40	18.962	92.549	98.857	18.898
SA-R-.3	93.396	99.353	18.827	91.988	99.012	18.930	92.188	98.386	18.881
SA-R-.5	92.957	99.416	18.800	91.748	98.529	18.894	91.754	97.742	18.850
SA-R-.7	92.627	99.905	18.773	92.113	98.798	18.859	91.418	97.499	18.808
SA-R-.9	93.002	99.006	18.716	91.541	98.197	18.817	91.923	97.998	18.765
SA-R-1.1	95.936	100.87	19.024	93.757	99.345	19.119	92.714	98.296	19.070
SA-R-1.3	95.214	100.22	18.988	92.110	98.583	19.081	91.710	97.797	19.037
SA-R-1.5	94.845	100.06	18.936	91.470	97.756	19.025	91.397	97.638	18.992
SA-R-1.7	92.617	98.895	18.906	91.975	98.304	18.994	91.325	97.430	18.942
SA-R-1.9	92.468	98.075	18.851	91.509	97.402	18.937	91.931	98.310	18.890
SA-R-2.1	93.471	99.037	18.676	92.113	98.400	18.780	91.086	97.284	18.724
SA-R-2.3	90.825	98.264	18.671	90.589	97.713	18.756	90.452	97.534	18.723
SA-R-2.5	91.443	98.359	18.657	90.027	97.917	18.752	90.401	96.973	18.707
SA-R-2.7	91.097	97.443	18.648	90.637	97.552	18.744	90.807	97.222	18.696
SA-R-2.9	91.534	98.264	18.644	90.934	97.928	18.739	90.962	97.264	18.686
SA-R-3.1	92.860	98.627	18.747	90.976	98.057	18.854	91.153	97.846	18.789
SA-R-3.3	91.287	98.343	18.745	90.807	97.821	18.849	90.571	97.264	18.789
SA-R-3.5	91.035	97.238	18.749	90.578	97.585	18.837	90.431	97.769	18.798
SA-R-3.7	91.002	98.453	18.752	90.497	97.681	18.835	90.606	96.751	18.797
SA-R-3.9	91.961	99.006	18.744	90.610	97.649	18.829	90.958	97.271	18.779
SA-H-.1	94.671	99.432	18.978	94.214	99.850	19.083	92.838	98.815	19.023
SA-H-.3	94.130	100.00	18.993	92.692	99.528	19.079	92.166	98.123	19.031
SA-H-.5	92.684	98.169	18.917	91.589	98.540	18.999	91.456	97.492	18.960
SA-H-.7	94.195	99.258	18.879	91.518	98.121	18.966	91.659	97.998	18.921
SA-H-.9	92.086	98.848	18.836	91.660	99.227	18.934	91.433	97.451	18.876
SA-H-1.1	97.144	100.95	19.103	94.172	100.74	19.205	93.143	98.635	19.150
SA-H-1.3	94.279	99.637	19.129	92.059	98.884	19.222	91.974	97.721	19.167
SA-H-1.5	93.969	99.779	19.021	91.779	98.508	19.114	91.539	97.243	19.068
SA-H-1.7	92.392	98.185	19.036	91.620	98.229	19.144	91.479	98.116	19.092
SA-H-1.9	92.463	98.690	18.938	91.474	97.864	19.026	91.544	97.478	18.987
SA-H-2.1	91.860	98.895	18.784	91.402	98.068	18.881	91.285	97.423	18.840
SA-H-2.3	90.957	97.506	18.771	90.338	97.488	18.870	90.476	97.444	18.819
SA-H-2.5	91.482	97.522	18.751	90.123	97.488	18.853	90.497	97.250	18.807
SA-H-2.7	91.239	97.917	18.747	90.258	97.778	18.835	90.714	97.312	18.782
SA-H-2.9	91.248	97.333	18.730	90.878	97.799	18.825	91.006	97.187	18.779
SA-H-3.1	92.978	98.374	18.944	90.846	98.046	19.023	91.148	98.040	19.000
SA-H-3.3	90.901	96.623	18.922	90.551	97.628	19.039	90.552	97.049	18.977
SA-H-3.5	90.931	97.854	18.920	90.297	97.230	19.017	90.756	97.104	18.969
SA-H-3.7	91.083	96.638	18.922	90.344	97.424	19.015	90.525	96.959	18.983
SA-H-3.9	91.164	97.901	18.923	90.863	97.542	19.012	90.852	97.402	18.963

Table A.6 Results for 6x6x6-problem size with total flow time as the performance criterion

6 x 6 x 6 Problems									
	S/R = 2			S/R = 5			S/R = 10		
Heuristic	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	0.064	100.00	100.00	0.063	100.00	100.00	0.066
HIT-M	101.31	101.21	0.065	100.30	100.42	0.066	100.37	100.38	0.066
CDS	96.991	98.234	0.747	96.912	98.521	0.761	98.412	99.593	0.751
CDS-M-1	98.215	99.241	0.762	97.130	99.236	0.763	98.655	99.868	0.764
CDS-M-2	95.861	97.079	1.445	95.454	96.937	1.554	96.969	98.277	1.588
CDS-M-3	95.895	97.317	1.410	95.398	96.969	1.657	97.006	98.230	1.521
NEH	93.334	102.03	1.830	93.449	100.72	1.832	94.325	100.47	1.824
NEH-M-1	93.316	101.74	1.818	93.497	100.68	1.822	94.371	100.43	1.819
TABU-R	93.597	100.84	286.70	91.249	97.572	273.32	92.221	97.289	268.34
T-R-M-1	93.761	100.58	276.66	91.025	97.427	260.17	92.029	97.205	251.47
T-R-M-2	94.442	101.37	281.69	91.310	97.781	263.41	92.498	97.744	250.17
TABU-H	93.296	99.242	262.78	91.574	96.446	264.01	92.197	96.127	245.97
T-H-M-1	92.753	99.162	264.64	91.042	96.044	262.76	91.712	96.016	249.38
T-H-M-2	93.015	98.959	268.38	91.276	95.843	270.56	92.068	95.736	252.90
SA-R-.1	96.158	101.43	32.265	94.034	98.673	32.399	93.928	97.717	32.139
SA-R-.3	96.873	101.55	32.221	93.137	98.143	32.363	93.309	97.416	32.103
SA-R-.5	95.359	101.14	32.183	92.126	97.998	32.315	92.808	96.946	32.055
SA-R-.7	94.648	100.87	32.139	91.926	97.347	32.264	92.162	96.856	32.013
SA-R-.9	94.328	100.63	32.088	91.854	97.869	32.230	92.566	96.565	31.966
SA-R-1.1	97.780	102.09	32.441	93.860	98.858	32.564	94.243	98.056	32.319
SA-R-1.3	95.420	101.22	32.405	92.869	98.416	32.535	92.730	96.735	32.287
SA-R-1.5	95.818	100.99	32.372	92.197	98.360	32.501	92.536	97.295	32.247
SA-R-1.7	94.276	100.37	32.340	91.223	97.676	32.481	92.595	97.448	32.226
SA-R-1.9	94.092	100.07	32.312	91.419	96.768	32.451	92.331	97.046	32.190
SA-R-2.1	94.653	100.33	31.937	91.556	97.282	32.067	92.890	96.851	31.810
SA-R-2.3	93.038	99.796	31.923	90.895	97.057	32.047	91.924	96.291	31.787
SA-R-2.5	93.044	100.27	31.917	91.028	96.470	32.044	91.569	97.104	31.783
SA-R-2.7	93.470	100.09	31.897	90.875	97.250	32.015	91.593	96.872	31.781
SA-R-2.9	94.067	100.86	31.878	90.978	96.519	32.012	91.948	96.803	31.751
SA-R-3.1	93.389	101.72	32.306	91.750	96.864	32.436	92.204	96.222	32.173
SA-R-3.3	93.200	100.51	32.290	91.056	96.703	32.439	91.572	96.222	32.170
SA-R-3.5	93.098	100.06	32.284	90.703	96.446	32.419	91.480	96.211	32.160
SA-R-3.7	92.739	99.343	32.285	90.869	97.427	32.422	91.724	96.016	32.167
SA-R-3.9	93.327	100.56	32.283	90.870	96.470	32.413	91.781	97.416	32.152
SA-H-.1	96.841	100.77	32.302	94.521	98.601	32.432	94.406	97.606	32.176
SA-H-.3	96.219	101.17	32.246	92.509	98.046	32.384	93.053	96.930	32.122
SA-H-.5	95.723	99.751	32.180	92.938	97.636	32.324	92.598	97.247	32.071
SA-H-.7	94.748	101.01	32.120	91.826	97.387	32.270	93.020	97.490	31.999
SA-H-.9	94.084	99.570	32.056	91.915	97.250	32.198	92.449	97.115	31.957
SA-H-1.1	97.385	101.80	32.832	93.575	98.046	32.962	94.418	97.976	32.701
SA-H-1.3	95.175	100.18	32.749	93.154	98.199	32.877	92.883	97.274	32.620
SA-H-1.5	95.306	101.71	32.674	92.072	97.178	32.802	92.403	97.443	32.553
SA-H-1.7	94.548	100.07	32.597	91.975	97.604	32.733	92.438	96.856	32.469
SA-H-1.9	94.397	100.56	32.523	92.045	97.379	32.652	92.311	97.089	32.399
SA-H-2.1	93.929	99.389	32.216	90.831	97.154	32.353	92.149	96.856	32.093
SA-H-2.3	92.556	99.140	32.199	90.755	96.824	32.360	91.892	96.777	32.083
SA-H-2.5	93.380	99.117	32.190	90.470	96.261	32.340	91.539	96.349	32.090
SA-H-2.7	92.639	99.717	32.197	90.715	97.009	32.321	91.257	96.238	32.069
SA-H-2.9	93.103	98.653	32.180	90.840	96.671	32.326	91.813	96.523	32.059
SA-H-3.1	94.407	100.68	32.391	91.955	97.483	32.536	92.707	96.708	32.287
SA-H-3.3	93.123	99.728	32.399	91.217	96.744	32.534	91.651	96.576	32.261
SA-H-3.5	92.592	99.853	32.376	90.484	96.607	32.520	91.596	96.539	32.275
SA-H-3.7	92.887	100.06	32.369	90.874	96.502	32.525	91.515	96.407	32.262

SA-H-3.9	92.793	98.721	32.384	90.698	96.720	32.518	91.692	96.217	32.242

Table A.7 Results for 5x6x8-problem size with total flow time as the performance criterion

5 x 6 x 8 Problems									
Heuristic	**S/R = 2**			**S/R = 5**			**S/R = 10**		
	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	0.095	100.00	100.00	0.079	100.00	100.00	0.097
HIT-M	100.96	100.88	0.084	100.85	100.90	0.082	100.70	100.73	0.084
CDS	96.821	98.698	0.837	96.638	98.326	0.839	97.781	98.915	0.835
CDS-M-1	97.296	99.223	0.860	96.841	98.533	0.838	98.011	99.196	0.850
CDS-M-2	95.741	97.505	1.414	95.489	96.942	1.618	95.978	97.290	1.689
CDS-M-3	96.100	97.812	1.423	95.899	97.580	1.515	96.060	97.600	1.711
NEH	92.549	99.978	2.940	92.816	99.644	2.944	93.836	100.37	2.956
NEH-M-1	92.334	99.727	2.930	92.923	99.577	2.939	93.807	100.35	2.934
TABU-R	94.136	99.311	398.81	93.137	97.829	383.68	93.229	98.437	324.41
T-R-M-1	94.259	99.442	383.40	93.416	98.086	344.80	92.957	97.914	341.87
T-R-M-2	95.199	100.60	349.36	93.838	99.287	351.74	93.014	98.100	340.23
TABU-H	93.799	99.026	340.18	93.031	96.544	346.29	92.991	97.352	330.45
T-H-M-1	93.402	98.293	334.13	93.072	96.685	353.95	92.487	97.493	326.11
T-H-M-2	93.728	98.227	333.76	93.296	96.983	346.56	92.695	97.021	327.93
SA-R-.1	96.856	101.26	35.495	95.926	100.15	35.475	94.433	98.533	35.458
SA-R-.3	95.930	99.803	35.432	93.711	97.961	35.411	93.718	98.595	35.412
SA-R-.5	95.733	100.27	35.395	94.869	99.892	35.356	93.121	98.179	35.347
SA-R-.7	95.354	100.31	35.319	93.950	98.036	35.294	92..966	97.920	35.300
SA-R-.9	94.940	100.10	35.255	93.338	98.077	35.226	93.021	97.358	35.222
SA-R-1.1	96.790	100.56	35.607	95.360	99.884	35.586	93.702	98.190	35.577
SA-R-1.3	96.030	100.14	35.580	94.897	100.08	35.554	93.155	97.763	35.539
SA-R-1.5	96.299	100.56	35.557	94.140	98.608	35.512	92.810	97.363	35.514
SA-R-1.7	94.526	99.136	35.509	93.536	98.500	35.494	93.039	98.195	35.470
SA-R-1.9	94.661	99.103	35.473	93.272	98.376	35.442	92.952	98.010	35.431
SA-R-2.1	94.062	99.070	35.058	93.074	98.492	35.031	92.845	97.622	35.030
SA-R-2.3	93.413	98.063	35.048	92.341	97.505	35.011	92.091	97.240	35.023
SA-R-2.5	93.645	99.004	35.029	92.408	96.983	35.002	91.955	97.510	34.997
SA-R-2.7	93.590	98.764	35.029	92.487	98.028	35.001	92.282	97.155	34.988
SA-R-2.9	94.111	99.519	35.016	93.197	97.787	34.970	92.556	97.217	34.978
SA-R-3.1	93.584	98.227	35.390	92.506	97.132	35.353	92.426	96.947	35.343
SA-R-3.3	93.177	98.906	35.388	92.302	98.351	35.365	91.712	97.015	35.343
SA-R-3.5	93.575	99.333	35.378	92.739	97.671	35.344	92.134	96.992	35.346
SA-R-3.7	93.485	99.114	35.394	92.614	98.052	35.352	92.221	96.998	35.358
SA-R-3.9	94.320	98.961	35.377	93.074	98.276	35.369	92.382	96.947	35.352
SA-H-.1	95.921	100.29	35.522	95.536	98.682	35.491	94.110	97.841	35.484
SA-H-.3	95.474	99.934	35.470	94.511	99.644	35.442	92.771	97.751	35.432
SA-H-.5	95.260	100.12	35.419	93.614	97.481	35.388	93.107	97.791	35.376
SA-H-.7	95.768	100.29	35.359	93.688	98.011	35.328	92.779	97.526	35.309
SA-H-.9	94.754	99.147	35.300	93.251	98.036	35.259	92.771	97.729	35.255
SA-H-1.1	97.027	100.02	35.977	95.496	99.809	35.943	93.892	98.235	35.929
SA-H-1.3	96.085	100.26	35.934	94.381	98.475	35.906	92.991	97.352	35.901
SA-H-1.5	95.496	99.945	35.901	93.465	98.657	35.871	93.017	97.071	35.869
SA-H-1.7	94.526	99.453	35.847	93.269	97.704	35.824	92.741	97.217	35.817
SA-H-1.9	94.919	99.727	35.819	93.623	97.961	35.791	92.953	97.195	35.792
SA-H-2.1	93.912	99.190	35.342	92.426	97.265	35.316	92.288	96.908	35.305
SA-H-2.3	92.506	98.764	35.322	92.094	97.878	35.292	92.377	97.122	35.288
SA-H-2.5	92.577	97.658	35.303	92.260	97.423	35.279	91.770	97.088	35.278
SA-H-2.7	93.119	98.742	35.277	92.073	97.464	35.244	92.139	96.475	35.240
SA-H-2.9	93.609	98.687	35.251	92.837	97.812	35.244	92.333	97.094	35.220
SA-H-3.1	93.500	98.315	35.801	92.721	97.398	35.768	92.274	97.279	35.770
SA-H-3.3	92.643	97.812	35.833	91.870	97.141	35.809	91.512	96.790	35.800
SA-H-3.5	92.941	98.895	35.855	92.400	97.870	35.843	91.917	96.728	35.826
SA-H-3.7	92.724	97.637	35.910	92.500	97.232	35.871	92.104	96.959	35.863

SA-H-3.9	93.694	99.004	35.931	92.870	97.920	35.901	92.367	97.257	35.919

Table A.8 Results for 8x8x8-problem size with total flow time as the performance criterion

8 X 8 X 8 Problems									
Heuristic	**S/R = 2**			**S/R = 5**			**S/R = 10**		
	RELF	AMF	CPU	RELF	AMF	CPU	RELF	AMF	CPU
HITOMI	100.00	100.00	0.156	100.00	100.00	0.160	100.00	100.00	0.156
HIT-M	101.28	100.82	0.159	100.67	100.62	0.161	100.60	100.56	0.162
CDS	96.708	97.500	2.938	97.250	98.138	2.948	98.252	97.835	2.952
CDS-M-1	97.021	98.100	2.965	97.256	98.369	2.968	98.743	98.009	2.972
CDS-M-2	95.654	96.376	5.426	95.898	96.722	5.895	95.906	96.675	6.842
CDS-M-3	96.054	96.771	5.301	95.931	97.096	5.615	95.899	95.897	6.876
NEH	92.519	97.936	9.430	93.320	98.415	9.446	93.636	98.341	9.457
NEH-M-1	92.785	98.324	9.446	93.368	98.538	9.468	93.500	98.239	9.475
TABU-R	93.517	97.296	1859.056	92.736	98.220	1936.089	91.422	96.031	1838.963
T-R-M-1	93.109	97.132	1857.241	92.431	97.512	1859.559	91.366	95.529	1773.783
T-R-M-2	93.699	97.745	1849.782	93.032	97.825	1801.088	91.546	96.134	1729.010
TABU-H	92.941	96.512	1724.829	92.177	97.286	1801.022	91.352	96.141	1750.284
T-H-M-1	92.409	96.424	1740.744	91.679	96.814	1741.302	90.609	95.257	1769.955
T-H-M-2	92.552	96.744	1777.433	91.806	96.701	1739.967	90.931	95.363	1739.421
SA-R-.1	97.146	99.312	75.864	96.591	99.528	75.931	95.639	98.101	75.936
SA-R-.3	96.455	99.571	75.791	95.841	99.179	75.873	94.366	97.092	75.880
SA-R-.5	95.913	98.406	75.730	94.950	98.702	75.815	93.556	97.492	75.808
SA-R-.7	95.675	98.876	75.672	95.193	98.779	75.736	93.150	96.781	75.745
SA-R-.9	95.385	98.433	75.583	94.723	98.107	75.650	93.328	97.135	75.664
SA-R-1.1	97.064	99.796	75.438	96.383	99.564	75.514	94.643	97.899	75.510
SA-R-1.3	95.954	98.488	75.374	96.041	99.359	75.443	93.679	96.824	75.450
SA-R-1.5	96.231	99.441	75.313	94.828	98.805	75.383	93.226	96.976	75.385
SA-R-1.7	95.674	98.665	75.237	94.484	98.040	75.311	93.186	96.749	75.317
SA-R-1.9	95.389	99.060	75.154	94.179	98.599	75.217	93.302	96.926	75.227
SA-R-2.1	95.528	98.583	74.707	94.798	99.118	74.800	94.219	97.036	74.826
SA-R-2.3	94.547	98.195	74.689	93.741	97.753	74.779	93.014	96.898	74.783
SA-R-2.5	94.448	98.304	74.678	93.379	98.148	74.720	92.822	97.110	74.740
SA-R-2.7	95.029	97.868	74.612	93.216	97.866	74.696	92.082	96.389	74.708
SA-R-2.9	94.384	98.045	74.598	93.652	98.497	74.676	92.474	96.824	74.664
SA-R-3.1	95.301	99.728	75.802	94.638	98.646	75.880	93.821	96.926	75.888
SA-R-3.3	94.514	97.725	75.785	93.787	97.312	75.851	92.416	96.265	75.855
SA-R-3.5	93.949	97.159	75.768	93.044	97.266	75.840	92.354	96.626	75.852
SA-R-3.7	94.230	98.331	75.765	93.090	97.389	75.835	92.323	96.664	75.821
SA-R-3.9	94.097	97.956	75.748	93.286	98.240	75.784	92.027	96.473	75.793
SA-H-.1	98.014	99.326	75.626	96.369	98.564	75.701	95.394	97.704	75.700
SA-H-.3	96.084	99.571	75.582	95.582	98.476	75.652	93.936	97.025	75.651
SA-H-.5	96.391	98.890	75.528	94.884	98.641	75.610	93.566	96.502	75.615
SA-H-.7	95.547	98.556	75.484	94.343	97.984	75.549	93.460	97.245	75.561
SA-H-.9	95.192	98.542	75.432	94.296	98.122	75.499	93.352	96.707	75.498
SA-H-1.1	97.965	99.373	75.702	96.502	99.066	75.788	95.323	98.037	75.770
SA-H-1.3	97.205	99.428	75.651	95.128	98.553	75.704	93.995	97.390	75.720
SA-H-1.5	95.305	98.065	75.588	94.735	98.805	75.648	93.644	97.322	75.651
SA-H-1.7	95.322	98.392	75.508	94.368	98.107	75.571	93.282	97.195	75.591
SA-H-1.9	95.264	98.426	75.449	94.129	98.179	75.468	93.203	96.707	75.528
SA-H-2.1	94.847	98.154	74.551	94.224	97.030	74.619	93.474	96.516	74.632
SA-H-2.3	94.372	97.343	74.544	93.492	97.584	74.604	93.536	97.170	74.609
SA-H-2.5	94.088	97.902	74.504	93.415	97.389	74.576	92.507	96.183	74.584
SA-H-2.7	93.926	97.793	74.506	92.836	97.199	74.567	92.543	96.611	74.579
SA-H-2.9	93.938	97.868	74.495	93.137	97.840	74.576	92.726	96.350	74.441
SA-H-3.1	95.040	97.793	75.901	94.431	98.676	75.969	93.772	96.657	75.992
SA-H-3.3	93.748	97.112	75.866	93.671	97.938	75.928	92.743	96.894	75.942
SA-H-3.5	93.925	98.290	75.828	93.050	97.994	75.911	92.335	95.695	75.918
SA-H-3.7	93.537	97.779	75.811	93.098	97.553	75.909	92.247	95.554	75.874

SA-H-3.9	93.611	97.146	75.815	92.844	97.363	75.853	92.040	95.515	75.873

APPENDIX B

RESULTS WITH RESPECT TO MAKESPAN

Tables B.1 through B.8 shows the results of solving the experimental group scheduling problems using the heuristics under study, with respect to makespan, a table for each problem size. The table is divided into three parts vertically one for each S/R ratio. For each heuristic at each S/R, the makespan (RELM) is listed in the first column. Relative total flow time for the solution (AFM) is given in the second column and the third column exhibits the computational times (CPU) in seconds.

Table B.1 Results for 3x3x3-problem size with makespan as the performance criterion

3 x 3 x 3 Problems									
Heuristic	S/R = 2			S/R = 5			S/R = 10		
	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	.009	100.00	100.00	.011	100.00	100.00	.009
HIT-M	100.29	100.09	.007	100.44	100.77	.009	100.28	100.32	.009
CDS	99.024	99.238	.038	99.908	99.459	.031	99.090	98.986	.031
CDS-M-1	99.060	99.175	.031	99.679	99.395	.031	99.035	99.068	.029
CDS-M-2	98.843	99.086	.022	99.519	99.364	.026	98.663	98.733	.048
CDS-M-3	98.915	98.832	.024	99.336	99.276	.059	98.635	98.805	.042
NEH	97.649	96.129	.040	99.473	97.228	.042	99.200	96.358	.037
NEH-M-1	97.794	95.875	.038	99.473	97.295	.042	99.380	96.568	.038
TABU-R	96.781	99.219	4.625	97.092	97.442	4.432	97.822	100.15	4.303
T-R-M-1	96.817	98.248	4.489	97.161	98.954	4.383	98.028	99.908	4.410
T-R-M-2	96.817	99.422	4.486	97.458	99.018	4.351	97.987	98.945	4.435
TABU-H	96.275	98.477	4.388	96.863	98.524	4.143	97.753	98.612	4.214
T-H-M-1	96.347	98.356	4.311	96.932	98.910	4.143	97.835	98.866	4.223
T-H-M-2	96.347	98.344	4.335	96.932	98.866	4.174	97.794	98.641	4.249
SA-R-.1	96.239	98.445	4.198	96.817	96.373	4.126	97.725	100.05	4.182
SA-R-.3	96.166	97.303	4.185	96.817	98.791	4.125	97.725	99.295	4.171
SA-R-.5	96.166	98.312	4.177	96.817	97.745	4.106	97.725	99.940	4.151
SA-R-.7	96.166	98.140	4.159	96.817	97.948	4.093	97.725	99.602	4.132
SA-R-.9	96.166	98.305	4.140	96.863	98.306	4.076	97.725	99.392	4.125
SA-R-1.1	96.166	97.607	4.417	96.817	98.147	4.354	97.725	100.82	4.404
SA-R-1.3	96.203	97.874	4.410	96.817	97.979	4.337	97.725	98.875	4.391
SA-R-1.5	96.239	97.550	4.398	96.817	98.035	4.326	97.725	99.583	4.382
SA-R-1.7	96.239	97.918	4.370	96.817	97.868	4.307	97.725	100.02	4.361
SA-R-1.9	96.239	97.284	4.365	96.817	96.957	4.298	97.725	98.933	4.337
SA-R-2.1	96.166	98.020	4.481	96.817	98.270	4.425	97.725	99.360	4.479
SA-R-2.3	96.166	97.944	4.453	96.817	98.047	4.387	97.725	101.18	4.452
SA-R-2.5	96.166	99.181	4.420	96.817	97.279	4.344	97.725	98.410	4.416
SA-R-2.7	96.166	98.337	4.391	96.817	97.836	4.327	97.725	98.366	4.376
SA-R-2.9	96.239	97.227	4.357	96.817	99.109	4.287	97.725	97.746	4.347
SA-R-3.1	96.203	97.899	4.535	96.817	98.258	4.472	97.725	99.686	4.535
SA-R-3.3	96.203	99.283	4.507	96.817	97.693	4.441	97.725	99.821	4.500
SA-R-3.5	96.166	98.451	4.482	96.817	98.107	4.404	97.725	97.980	4.472
SA-R-3.7	96.166	98.572	4.446	96.817	98.254	4.381	97.725	98.139	4.440
SA-R-3.9	96.383	98.198	4.416	96.840	98.508	4.336	97.739	98.984	4.400
SA-H-.1	96.166	97.531	4.197	96.817	98.226	4.128	97.725	99.151	4.184
SA-H-.3	96.166	98.470	4.190	96.817	97.960	4.119	97.725	99.160	4.167
SA-H-.5	96.166	97.982	4.175	96.817	97.383	4.114	97.725	99.030	4.162
SA-H-.7	96.166	98.255	4.164	96.817	98.015	4.099	97.725	99.208	4.152
SA-H-.9	96.166	98.325	4.153	96.817	98.015	4.081	97.725	99.225	4.128
SA-H-1.1	96.166	97.861	4.381	96.817	97.661	4.310	97.725	98.670	4.365
SA-H-1.3	96.166	97.779	4.365	96.817	97.932	4.293	97.725	99.078	4.354
SA-H-1.5	96.166	98.388	4.363	96.817	98.143	4.281	97.725	99.126	4.343
SA-H-1.7	96.166	97.880	4.347	96.817	97.244	4.273	97.725	98.880	4.323
SA-H-1.9	96.275	98.432	4.322	96.840	98.011	4.253	97.725	98.861	4.312
SA-H-2.1	96.166	98.445	4.453	96.817	97.283	4.411	97.725	99.013	4.466
SA-H-2.3	96.203	97.696	4.429	96.817	97.944	4.374	97.725	98.231	4.435
SA-H-2.5	96.203	97.645	4.410	96.817	98.333	4.347	97.725	98.192	4.403
SA-H-2.7	96.275	97.956	4.375	96.817	97.486	4.315	97.725	98.678	4.369
SA-H-2.9	96.275	97.633	4.347	96.817	97.876	4.274	97.725	98.366	4.345
SA-H-3.1	96.166	98.001	4.477	96.817	97.928	4.419	97.725	99.749	4.485
SA-H-3.3	96.166	98.490	4.460	96.817	98.019	4.385	97.725	99.377	4.453
SA-H-3.5	96.166	98.210	4.422	96.817	98.536	4.350	97.725	98.226	4.416
SA-H-3.7	96.166	98.179	4.390	96.817	98.011	4.325	97.725	98.533	4.390
SA-H-3.9	96.203	97.271	4.361	96.840	99.089	4.292	97.725	99.616	4.341

Table B.2 Results for 3x4x5-problem size with makespan as the performance criterion

3 x 4 x 5 Problems									
	S/R = 2			S/R = 5			S/R = 10		
Heuristic	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	.021	100.00	100.00	.020	100.00	100.00	.021
HIT-M	101.51	100.53	.018	100.64	100.62	.018	100.42	100.03	.016
CDS	98.774	98.310	.105	97.636	98.508	.093	98.784	99.159	.102
CDS-M-1	98.851	98.345	.095	97.458	98.393	.096	98.761	99.127	.101
CDS-M-2	97.727	97.803	.152	96.391	98.177	.150	97.804	98.279	.147
CDS-M-3	97.753	98.334	.161	96.338	97.775	.154	97.759	98.109	.150
NEH	97.702	94.672	.207	96.213	94.716	.208	97.838	96.275	.209
NEH-M-1	97.676	94.969	.207	96.516	95.291	.208	97.827	96.475	.209
TABU-R	95.480	96.769	19.549	94.133	96.554	18.668	96.228	98.214	17.949
T-R-M-1	95.991	95.977	18.751	94.329	98.222	17.911	96.610	98.294	17.249
T-R-M-2	96.246	95.894	19.067	94.649	96.251	17.599	96.689	98.493	17.119
TABU-H	95.148	96.151	18.067	94.044	96.932	17.197	96.126	96.937	17.005
T-H-M-1	95.301	96.469	17.146	93.973	96.484	16.979	96.092	97.379	16.430
T-H-M-2	95.531	96.378	17.375	94.098	96.843	17.081	96.182	97.388	16.363
SA-R-.1	95.378	96.365	8.833	93.849	96.781	8.803	96.047	97.579	8.749
SA-R-.3	95.046	96.199	8.806	93.671	95.246	8.771	95.800	97.741	8.740
SA-R-.5	95.148	96.175	8.775	93.920	96.127	8.749	95.946	98.339	8.708
SA-R-.7	95.378	96.678	8.751	93.867	95.812	8.724	96.126	98.184	8.670
SA-R-.9	96.476	97.056	8.720	94.329	95.705	8.694	96.205	98.263	8.646
SA-R-1.1	95.072	96.488	9.265	93.796	95.729	9.240	95.867	96.742	9.195
SA-R-1.3	95.225	96.638	9.244	93.653	96.105	9.223	95.856	96.899	9.174
SA-R-1.5	95.072	95.819	9.230	93.671	96.065	9.202	95.946	97.270	9.155
SA-R-1.7	95.608	96.180	9.200	93.778	96.526	9.171	95.946	98.055	9.128
SA-R-1.9	96.399	97.029	9.185	94.400	96.685	9.147	96.239	97.361	9.110
SA-R-2.1	95.174	95.674	9.231	93.724	96.233	9.204	95.935	96.986	9.165
SA-R-2.3	95.429	96.879	9.190	93.813	96.022	9.166	95.867	97.487	9.131
SA-R-2.5	95.020	95.963	9.151	93.689	95.250	9.116	95.935	97.969	9.074
SA-R-2.7	95.123	96.895	9.117	93.760	95.877	9.073	95.901	98.061	9.038
SA-R-2.9	95.965	96.724	9.074	94.329	96.704	9.041	96.081	98.834	9.006
SA-R-3.1	95.250	95.802	9.338	93.724	96.317	9.314	95.901	97.601	9.271
SA-R-3.3	95.174	96.124	9.281	93.796	95.492	9.261	95.878	97.403	9.214
SA-R-3.5	95.072	95.532	9.241	93.564	95.951	9.215	95.811	97.797	9.175
SA-R-3.7	95.199	96.220	9.200	93.636	95.453	9.168	95.845	97.801	9.130
SA-R-3.9	95.914	96.874	9.171	94.222	96.383	9.132	96.205	97.729	9.087
SA-H-.1	95.072	96.520	8.733	93.849	96.176	8.715	95.935	97.307	8.670
SA-H-.3	95.046	95.371	8.725	93.564	95.564	8.696	95.901	97.351	8.655
SA-H-.5	95.378	96.352	8.705	93.653	95.509	8.685	96.014	97.458	8.637
SA-H-.7	95.046	96.252	8.683	93.796	95.406	8.665	96.104	97.742	8.619
SA-H-.9	95.404	96.228	8.668	94.098	95.889	8.665	96.284	97.850	8.609
SA-H-1.1	95.097	96.453	9.146	93.689	96.234	9.115	95.878	97.157	9.064
SA-H-1.3	94.944	96.432	9.121	93.529	95.611	9.099	95.935	97.455	9.049
SA-H-1.5	94.791	95.415	9.097	93.689	96.481	9.079	95.856	97.816	9.034
SA-H-1.7	95.174	96.266	9.081	93.600	95.894	9.057	95.912	97.160	9.015
SA-H-1.9	95.480	97.067	9.072	94.133	96.225	9.053	96.160	97.019	9.001
SA-H-2.1	95.123	97.278	9.319	93.778	95.883	9.304	95.890	97.419	9.257
SA-H-2.3	94.842	95.481	9.275	93.742	95.673	9.244	95.980	97.335	9.207
SA-H-2.5	95.046	96.081	9.229	93.796	95.639	9.193	95.878	97.680	9.161
SA-H-2.7	95.020	96.563	9.186	93.689	95.321	9.146	95.890	96.554	9.114
SA-H-2.9	95.480	95.768	9.137	94.116	96.080	9.104	96.137	97.928	9.063
SA-H-3.1	95.250	96.183	9.266	93.653	96.045	9.248	95.935	98.119	9.215
SA-H-3.3	94.893	95.489	9.223	93.707	95.947	9.200	95.890	97.334	9.166
SA-H-3.5	94.995	96.269	9.182	93.529	95.676	9.159	95.867	97.051	9.123
SA-H-3.7	95.174	96.394	9.147	93.600	95.945	9.112	95.901	97.818	9.070
SA-H-3.9	95.352	96.670	9.094	94.276	96.090	9.079	96.137	98.910	9.039

Table B.3 Results for 4x4x4-problem size with makespan as the performance criterion

4 x 4 x 4 Problems									
Heuristic	**S/R = 2**			**S/R = 5**			**S/R = 10**		
	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	.022	100.00	100.00	.020	100.00	100.00	.018
HIT-M	100.84	100.65	.020	100.59	100.52	.020	100.20	100.07	.018
CDS	98.691	99.183	.104	98.930	99.312	.110	98.082	100.04	.109
CDS-M-1	99.042	100.07	.110	98.959	99.274	.110	98.045	99.947	.110
CDS-M-2	97.078	98.904	.187	97.727	98.639	.187	96.404	99.744	.185
CDS-M-3	97.218	99.505	.188	97.756	98.597	.204	96.432	99.463	.181
NEH	98.130	96.385	.185	97.419	95.874	.194	97.099	97.495	.191
NEH-M-1	97.779	96.305	.192	97.507	95.951	.192	97.238	97.700	.191
TABU-R	95.208	98.038	20.363	95.557	97.779	19.356	93.809	98.397	19.406
T-R-M-1	95.255	97.988	18.748	95.087	95.726	18.929	93.846	97.436	19.143
T-R-M-2	95.722	99.546	19.418	95.117	95.638	18.935	93.976	96.093	19.021
TABU-H	94.437	96.930	18.887	94.955	96.331	18.429	93.744	97.470	19.475
T-H-M-1	93.922	96.868	19.228	94.545	95.465	18.292	93.577	96.983	19.154
T-H-M-2	94.296	97.517	19.645	94.706	95.758	18.507	93.596	97.019	19.290
SA-R-.1	94.951	98.476	9.467	94.427	96.064	9.562	93.698	97.040	9.522
SA-R-.3	94.367	97.606	9.454	94.398	95.928	9.539	93.596	96.625	9.498
SA-R-.5	94.437	97.774	9.438	94.383	96.242	9.506	93.633	97.446	9.484
SA-R-.7	94.554	97.966	9.416	94.354	95.843	9.495	93.577	97.463	9.465
SA-R-.9	94.764	97.534	9.386	94.853	96.528	9.486	93.707	97.592	9.446
SA-R-1.1	94.507	97.567	9.973	94.515	95.440	10.07	93.642	97.087	10.03
SA-R-1.3	94.390	97.745	9.939	94.281	95.252	10.04	93.577	96.805	10.01
SA-R-1.5	94.367	96.870	9.916	94.325	95.294	10.01	93.633	96.459	9.977
SA-R-1.7	94.250	97.421	9.897	94.383	96.409	9.994	93.615	97.025	9.959
SA-R-1.9	94.764	97.995	9.879	94.603	96.719	9.959	93.679	98.773	9.927
SA-R-2.1	94.460	97.457	10.034	94.413	95.865	10.119	93.615	97.208	10.110
SA-R-2.3	94.296	98.291	9.989	94.354	96.236	10.075	93.577	97.638	10.045
SA-R-2.5	94.507	98.147	9.942	94.369	95.426	10.025	93.550	97.029	10.000
SA-R-2.7	94.624	98.942	9.909	94.339	96.040	9.985	93.577	97.599	9.951
SA-R-2.9	94.857	98.228	9.856	94.589	96.175	9.937	93.652	96.877	9.903
SA-R-3.1	94.437	98.918	10.114	94.427	95.560	10.201	93.587	97.620	10.197
SA-R-3.3	94.367	98.183	10.068	94.222	96.134	10.142	93.596	97.115	10.128
SA-R-3.5	94.156	96.962	10.014	94.237	95.869	10.099	93.568	96.628	10.075
SA-R-3.7	94.250	98.123	9.972	94.222	95.707	10.048	93.587	96.642	10.028
SA-R-3.9	94.694	98.978	9.931	94.530	95.396	10.012	93.754	97.911	9.977
SA-H-.1	94.811	98.224	9.533	94.559	96.102	9.616	93.661	97.072	9.591
SA-H-.3	94.530	98.284	9.518	94.413	95.181	9.599	93.661	96.793	9.572
SA-H-.5	94.461	97.457	9.483	94.354	95.904	9.580	93.587	97.174	9.548
SA-H-.7	94.437	97.377	9.464	94.457	95.623	9.554	93.615	97.279	9.522
SA-H-.9	94.577	97.712	9.452	94.574	96.039	9.527	93.624	97.171	9.502
SA-H-1.1	94.390	97.312	9.855	94.339	95.849	9.949	93.670	96.708	9.914
SA-H-1.3	94.413	97.269	9.833	94.310	95.628	9.924	93.587	97.110	9.883
SA-H-1.5	94.156	96.791	9.815	94.295	95.941	9.892	93.587	96.887	9.866
SA-H-1.7	94.437	98.000	9.793	94.266	95.587	9.885	93.643	97.377	9.854
SA-H-1.9	94.507	97.887	9.764	94.574	95.764	9.855	93.716	97.062	9.829
SA-H-2.1	94.624	98.673	10.051	94.339	96.010	10.135	93.605	97.031	10.133
SA-H-2.3	94.647	98.671	10.000	94.413	95.745	10.084	93.605	96.044	10.062
SA-H-2.5	94.390	98.096	9.954	94.281	95.467	10.033	93.568	97.747	10.000
SA-H-2.7	94.063	97.933	9.904	94.281	96.239	9.981	93.540	96.383	9.944
SA-H-2.9	94.530	98.005	9.855	94.530	95.680	9.925	93.670	97.069	9.899
SA-H-3.1	94.717	98.327	10.004	94.369	96.073	10.085	93.615	97.752	10.080
SA-H-3.3	94.296	97.555	9.950	94.251	96.280	10.024	93.540	97.509	10.001
SA-H-3.5	94.156	97.889	9.897	94.207	95.992	9.973	93.559	97.564	9.956
SA-H-3.7	94.413	98.707	9.840	94.295	95.660	9.927	93.605	97.028	9.882
SA-H-3.9	94.507	97.286	9.790	94.515	95.610	9.859	93.642	96.641	9.836

Table B.4 Results for 6x5x4-problem size with makespan as the performance criterion

6 x 5 x 4 Problems									
	S/R = 2			S/R = 5			S/R = 10		
Heuristic	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	.035	100.00	100.00	.035	100.00	100.00	.033
HIT-M	100.82	100.44	.036	100.78	100.52	.037	100.07	100.00	.035
CDS	97.647	97.049	.304	98.779	98.919	.297	98.489	98.918	.316
CDS-M-1	97.737	96.957	.306	98.839	98.912	.309	98.543	98.843	.309
CDS-M-2	96.343	96.923	.599	96.568	97.866	.619	97.032	97.850	.548
CDS-M-3	96.164	96.541	.576	96.467	97.797	.624	96.882	97.709	.575
NEH	96.179	93.956	.491	97.497	96.513	.489	96.930	96.661	.486
NEH-M-1	95.834	93.939	.492	97.628	96.543	.488	97.038	96.870	.488
TABU-R	93.496	96.199	83.041	94.115	96.603	81.072	93.703	96.981	81.454
T-R-M-1	92.492	94.174	85.907	93.398	96.719	81.052	93.475	95.920	79.883
T-R-M-2	93.301	95.836	82.987	94.216	96.263	82.137	93.889	96.661	81.283
TABU-H	92.687	95.249	82.753	93.822	97.078	81.589	93.601	95.773	81.828
T-H-M-1	92.207	94.641	80.618	93.519	96.313	81.215	93.547	95.122	81.747
T-H-M-2	92.597	95.026	81.632	93.681	96.375	79.650	93.679	95.784	81.751
SA-R-.1	94.455	96.459	17.590	94.801	97.391	17.629	94.909	97.048	17.634
SA-R-.3	93.871	96.409	17.564	94.690	97.267	17.619	94.279	97.142	17.625
SA-R-.5	93.526	95.192	17.527	94.317	96.881	17.582	93.955	96.286	17.581
SA-R-.7	93.661	95.971	17.500	94.266	97.031	17.549	93.751	96.397	17.549
SA-R-.9	93.736	95.428	17.463	94.155	97.092	17.512	93.751	96.260	17.522
SA-R-1.1	94.051	94.773	18.528	94.468	97.553	18.566	94.633	97.817	18.566
SA-R-1.3	93.376	94.736	18.486	94.650	98.141	18.537	94.111	97.491	18.537
SA-R-1.5	93.646	96.048	18.446	94.095	96.653	18.509	93.991	96.584	18.495
SA-R-1.7	93.361	95.105	18.405	94.014	96.951	18.466	93.925	96.412	18.470
SA-R-1.9	93.466	95.850	18.381	94.064	96.267	18.425	93.781	95.995	18.431
SA-R-2.1	93.676	95.581	18.444	94.842	97.597	18.480	94.477	96.832	18.498
SA-R-2.3	93.271	95.750	18.390	93.751	95.553	18.416	93.649	96.150	18.424
SA-R-2.5	93.182	95.469	18.323	94.064	96.274	18.353	93.661	96.595	18.351
SA-R-2.7	93.122	95.536	18.266	93.903	96.301	18.308	93.541	95.929	18.290
SA-R-2.9	93.271	95.625	18.228	94.226	96.982	18.251	93.553	95.982	18.237
SA-R-3.1	93.991	96.102	18.539	94.710	96.756	18.581	94.579	97.234	18.616
SA-R-3.3	93.197	95.351	18.460	94.165	96.842	18.508	93.583	96.689	18.506
SA-R-3.5	92.807	94.736	18.402	93.883	96.601	18.433	93.385	96.921	18.437
SA-R-3.7	93.226	95.257	18.351	93.741	97.161	18.382	93.577	97.128	18.387
SA-R-3.9	93.376	95.103	18.293	93.953	96.886	18.321	93.547	96.138	18.309
SA-H-.1	93.826	95.701	17.505	94.650	97.591	17.544	94.441	96.529	17.540
SA-H-.3	93.646	95.756	17.482	94.438	96.239	17.527	93.871	96.784	17.534
SA-H-.5	93.691	95.814	17.459	94.246	96.476	17.510	93.601	96.213	17.506
SA-H-.7	93.212	94.865	17.432	94.135	97.150	17.478	93.739	95.631	17.483
SA-H-.9	93.706	95.647	17.411	94.044	96.864	17.458	94.027	96.450	17.460
SA-H-1.1	93.631	95.063	18.311	94.347	96.927	18.368	94.207	95.482	18.361
SA-H-1.3	93.361	96.118	18.269	94.246	96.694	18.315	94.099	96.907	18.318
SA-H-1.5	93.481	96.181	18.219	94.105	96.781	18.257	93.913	95.985	18.274
SA-H-1.7	93.451	95.105	18.175	94.115	96.047	18.230	93.733	96.595	18.216
SA-H-1.9	93.226	95.388	18.117	94.195	97.783	18.163	93.673	96.656	18.167
SA-H-2.1	94.111	95.494	18.445	94.508	96.939	18.482	94.303	96.668	18.488
SA-H-2.3	93.107	95.484	18.360	93.943	97.272	18.407	93.409	96.386	18.412
SA-H-2.5	93.017	94.714	18.291	93.701	96.294	18.337	93.523	97.210	18.336
SA-H-2.7	92.942	95.202	18.230	93.782	97.260	18.262	93.553	95.947	18.250
SA-H-2.9	93.226	95.712	18.167	93.994	96.901	18.202	93.625	95.803	18.180
SA-H-3.1	93.391	95.084	18.438	94.498	96.791	18.478	94.423	97.225	18.494
SA-H-3.3	93.137	95.250	18.396	93.912	96.573	18.393	93.553	95.237	18.412
SA-H-3.5	93.316	94.961	18.297	93.651	96.471	18.333	93.529	95.698	18.337
SA-H-3.7	93.212	95.078	18.241	92.812	96.318	18.266	93.427	95.435	18.261
SA-H-3.9	93.301	95.843	18.184	93.923	96.726	18.208	93.679	96.487	18.192

Table B.5 Results for 5x5x5-problem size with makespan as the performance criterion

5 x 5 x 5 Problems									
	S/R = 2			**S/R = 5**			**S/R = 10**		
Heuristic	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	.036	100.00	100.00	.042	100.00	100.00	.035
HIT-M	100.95	100.70	.038	100.68	100.62	.037	100.27	100.18	.036
CDS	96.986	98.248	.299	98.776	99.036	.299	98.497	98.487	.294
CDS-M-1	97.128	98.327	.296	98.884	99.307	.300	98.511	98.597	.294
CDS-M-2	96.023	97.841	.474	97.069	98.684	.580	96.425	97.876	.572
CDS-M-3	95.818	97.577	.531	97.166	98.563	.571	96.398	98.218	.575
NEH	95.249	93.829	.642	97.660	95.872	.642	97.555	97.307	.647
NEH-M-1	95.376	94.107	.638	98.100	96.244	.641	97.534	97.474	.647
TABU-R	94.097	96.460	83.944	95.094	97.906	77.365	94.437	96.383	76.502
T-R-M-1	93.718	95.425	82.356	94.600	96.193	78.857	94.049	96.604	76.135
T-R-M-2	93.924	95.474	81.985	95.416	97.065	79.559	94.493	96.803	76.450
TABU-H	93.939	96.279	79.900	94.418	97.191	77.158	94.195	96.944	76.328
T-H-M-1	93.040	95.063	78.204	94.063	96.269	76.877	93.703	96.005	77.050
T-H-M-2	93.324	95.265	79.247	94.375	96.495	78.182	94.015	96.009	78.077
SA-R-.1	94.460	97.024	18.009	95.137	97.070	18.097	94.929	96.846	18.040
SA-R-.3	93.939	96.291	17.995	94.836	96.262	18.071	94.105	96.845	18.022
SA-R-.5	93.939	95.828	17.972	94.600	96.071	18.052	94.139	96.150	18.012
SA-R-.7	93.892	95.782	17.941	94.332	96.576	18.041	94.077	97.240	17.978
SA-R-.9	93.876	96.536	17.918	94.643	98.086	18.017	94.091	97.183	17.959
SA-R-1.1	94.287	96.199	18.896	94.869	97.653	18.982	94.590	97.760	18.938
SA-R-1.3	93.734	96.263	18.858	94.353	97.271	18.961	94.015	96.754	18.917
SA-R-1.5	93.624	96.565	18.833	94.289	96.819	18.933	94.077	96.666	18.874
SA-R-1.7	93.640	96.935	18.790	94.418	97.041	18.881	94.008	96.428	18.835
SA-R-1.9	93.955	96.100	18.746	94.632	97.141	18.829	94.306	96.871	18.787
SA-R-2.1	94.271	96.289	18.927	94.708	96.439	19.036	94.077	97.112	18.977
SA-R-2.3	93.829	95.811	18.853	94.375	96.802	18.946	93.890	96.504	18.886
SA-R-2.5	93.813	96.114	18.781	94.128	96.168	18.874	93.682	96.264	18.813
SA-R-2.7	93.513	95.696	18.718	94.063	96.615	18.799	93.897	96.225	18.739
SA-R-2.9	93.876	96.126	18.662	94.482	97.360	18.754	93.959	96.858	19.676
SA-R-3.1	94.066	96.222	18.035	94.804	96.859	19.127	94.437	97.492	19.075
SA-R-3.3	93.845	95.968	18.970	93.988	96.238	19.056	93.758	96.466	18.999
SA-R-3.5	93.466	95.797	18.902	94.181	96.338	18.981	93.696	96.748	18.920
SA-R-3.7	93.419	95.965	18.834	93.999	96.540	18.920	93.786	96.422	18.847
SA-R-3.9	93.797	96.725	18.778	94.246	96.810	18.856	93.897	96.642	18.785
SA-H-.1	94.492	97.231	17.915	95.330	97.191	18.005	94.590	97.748	17.965
SA-H-.3	94.113	96.365	17.911	94.632	96.763	17.987	94.077	96.788	17.949
SA-H-.5	93.829	95.561	17.888	94.514	97.824	17.980	94.216	96.972	17.935
SA-H-.7	93.513	95.958	17.866	94.278	96.610	17.964	94.063	96.749	17.917
SA-H-.9	93.734	96.045	17.850	94.461	97.143	17.956	94.264	96.525	17.889
SA-H-1.1	94.350	96.589	18.659	94.729	97.701	18.755	94.548	97.547	18.714
SA-H-1.3	93.797	96.323	18.651	94.546	96.950	18.745	94.001	96.517	18.693
SA-H-1.5	93.450	95.886	18.631	94.525	97.363	18.725	94.077	96.400	18.659
SA-H-1.7	93.718	96.187	18.613	94.504	96.993	18.676	93.911	96.404	18.645
SA-H-1.9	93.640	95.776	18.568	94.504	96.714	18.661	94.167	96.641	18.621
SA-H-2.1	93.908	96.313	18.971	94.428	96.560	19.077	94.229	96.257	19.022
SA-H-2.3	93.482	95.642	18.893	94.310	96.786	18.978	93.800	96.713	18.926
SA-H-2.5	93.734	96.199	18.812	94.149	96.563	18.889	93.841	97.101	18.839
SA-H-2.7	93.592	96.488	18.734	94.203	96.853	18.821	94.035	96.532	18.758
SA-H-2.9	93.750	95.606	18.667	94.310	96.773	18.733	93.987	97.268	18.673
SA-H-3.1	94.003	96.146	18.901	94.622	97.116	18.993	94.091	96.759	18.944
SA-H-3.3	93.245	95.479	18.818	94.063	96.410	18.902	93.869	96.855	18.859
SA-H-3.5	93.482	95.879	18.757	94.031	96.201	18.834	93.724	96.333	18.783
SA-H-3.7	93.576	96.013	18.694	94.085	96.908	18.770	93.876	95.909	18.707
SA-H-3.9	93.592	95.616	18.624	94.096	96.097	18.697	94.021	96.834	18.637

Table B.6 Results for 6x6x6-problem size with makespan as the performance criterion

6 x 6 x 6 Problems									
Heuristic	S/R = 2			S/R = 5			S/R = 10		
	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	0.064	100.00	100.00	0.063	100.00	100.00	0.066
HIT-M	101.21	101.31	0.065	100.42	100.30	0.066	100.38	100.37	0.066
CDS	97.962	98.740	0.691	97.065	97.582	0.698	98.478	99.130	0.696
CDS-M-1	98.585	99.403	0.704	97.178	97.588	0.710	98.605	99.273	0.703
CDS-M-2	96.389	97.897	1.296	95.457	96.732	1.371	96.629	98.935	1.411
CDS-M-3	97.068	98.574	1.276	95.714	97.110	1.358	96.534	99.142	1.539
NEH	96.943	94.979	1.803	96.631	95.229	1.816	97.052	95.990	1.796
NEH-M-1	96.683	94.763	1.787	96.720	95.287	1.808	97.200	96.230	1.785
TABU-R	95.381	98.407	295.819	93.777	95.579	275.953	93.939	96.790	268.786
T-R-M-1	94.883	96.838	272.629	93.214	95.075	261.804	93.427	96.278	257.656
T-R-M-2	95.517	97.868	273.168	93.640	95.334	264.150	93.939	97.006	261.153
TABU-H	94.634	96.360	264.136	93.117	95.091	266.147	93.575	96.109	260.962
T-H-M-1	93.558	96.164	260.484	92.394	94.135	267.978	93.131	95.832	259.234
T-H-M-2	93.955	96.069	264.631	92.683	94.674	267.685	93.432	96.061	264.175
SA-R-.1	97.181	98.684	30.932	94.516	95.916	31.044	95.060	97.218	30.793
SA-R-.3	96.173	98.232	30.878	94.026	95.261	31.005	94.272	97.610	30.757
SA-R-.5	96.185	97.656	30.843	93.881	95.205	30.965	94.061	96.662	30.726
SA-R-.7	95.958	98.533	30.800	93.656	96.174	30.921	94.056	96.521	30.673
SA-R-.9	95.766	98.710	30.762	93.929	95.669	30.878	94.262	96.836	30.631
SA-R-1.1	96.841	99.691	32.350	94.742	96.294	32.458	95.075	97.883	32.199
SA-R-1.3	96.004	98.745	32.294	94.066	95.996	32.423	94.114	96.739	32.166
SA-R-1.5	95.834	98.314	32.257	93.769	95.243	32.376	93.987	97.346	32.109
SA-R-1.7	95.992	99.178	32.204	93.367	95.067	32.348	94.019	96.145	32.098
SA-R-1.9	95.947	98.562	32.174	93.921	96.262	32.283	93.918	96.865	32.030
SA-R-2.1	97.181	99.683	32.118	94.468	96.200	32.266	94.684	96.970	31.984
SA-R-2.3	95.641	98.095	32.001	93.849	96.264	32.142	94.019	97.124	31.864
SA-R-2.5	95.766	98.661	31.915	93.415	96.050	32.039	93.670	97.112	31.757
SA-R-2.7	95.472	97.870	31.827	93.351	94.753	31.944	93.453	96.355	31.661
SA-R-2.9	95.755	98.565	31.754	93.511	95.571	31.879	93.728	95.965	31.591
SA-R-3.1	96.524	97.912	32.688	94.106	96.533	32.816	94.431	97.416	32.534
SA-R-3.3	95.800	98.251	32.578	93.600	95.797	32.706	93.670	96.804	32.416
SA-R-3.5	95.415	97.917	32.491	93.375	95.662	32.615	93.702	96.839	32.326
SA-R-3.7	95.223	97.641	32.425	93.262	95.430	32.538	93.606	97.688	32.249
SA-R-3.9	95.800	97.774	32.351	93.616	95.678	32.467	93.797	96.679	32.177
SA-H-.1	96.151	97.961	30.809	94.766	96.001	30.937	94.906	97.193	30.689
SA-H-.3	95.947	97.745	30.781	93.986	95.625	30.891	94.008	97.086	30.658
SA-H-.5	96.015	98.115	30.738	93.825	95.075	30.863	94.209	97.218	30.622
SA-H-.7	95.732	98.262	30.696	93.849	95.833	30.827	93.960	96.878	30.579
SA-H-.9	95.653	97.465	30.655	93.616	96.222	30.784	94.008	97.467	30.537
SA-H-1.1	96.389	98.851	32.371	94.645	96.468	32.496	94.774	96.257	32.225
SA-H-1.3	95.913	98.290	32.291	93.753	95.722	32.412	93.955	96.757	32.154
SA-H-1.5	95.517	98.001	32.191	93.560	95.627	32.336	93.828	97.007	32.073
SA-H-1.7	95.596	97.984	32.110	93.415	95.455	32.247	94.024	96.872	31.983
SA-H-1.9	95.517	98.137	32.023	93.544	95.518	32.170	93.675	96.829	31.900
SA-H-2.1	96.106	98.117	32.304	94.155	95.846	32.440	94.219	97.567	32.147
SA-H-2.3	95.562	97.214	32.194	93.407	94.605	32.327	93.538	96.615	32.048
SA-H-2.5	95.121	98.113	32.139	93.238	95.705	32.262	93.517	97.106	31.973
SA-H-2.7	95.336	97.499	32.072	93.278	95.286	32.188	93.649	97.125	31.905
SA-H-2.9	95.426	97.952	32.030	93.415	95.285	32.137	93.760	96.700	31.845
SA-H-3.1	95.789	98.481	32.431	94.235	95.372	32.572	94.077	98.163	32.303
SA-H-3.3	95.381	97.515	32.334	93.576	95.641	32.458	93.696	96.142	32.175
SA-H-3.5	95.234	98.467	32.272	93.342	95.236	32.383	93.532	96.426	32.097
SA-H-3.7	95.415	98.782	32.193	93.310	95.911	32.317	93.416	96.418	32.023
SA-H-3.9	95.324	97.665	32.136	93.190	95.509	32.256	93.591	97.629	31.961

Table B.7 Results for 5x6x8 problem size with makespan as the performance criterion

5 x 6 x 8 Problems									
	S/R = 2			S/R = 5			S/R = 10		
Heuristic	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	0.095	100.00	100.00	0.079	100.00	100.00	0.097
HIT-M	100.88	100.96	0.084	100.90	100.85	0.082	100.73	100.70	0.084
CDS	97.275	97.556	0.786	96.934	97.513	0.781	97.656	98.343	0.783
CDS-M-1	97.582	98.139	0.808	97.331	97.781	0.785	97.898	98.577	0.792
CDS-M-2	95.875	97.021	1.422	95.550	96.840	1.416	96.132	97.419	1.418
CDS-M-3	96.094	97.456	1.483	95.533	97.235	1.559	96.177	97.479	1.363
NEH	95.404	93.370	2.898	95.889	93.975	2.922	96.441	95.463	2.906
NEH-M-1	95.142	93.136	2.890	95.815	93.919	2.891	96.329	95.511	2.882
TABU-R	95.361	97.692	367.996	94.638	96.502	348.300	94.131	96.635	330.856
T-R-M-1	94.989	96.551	347.243	93.900	96.084	339.412	93.771	96.211	323.978
T-R-M-2	95.787	97.829	345.916	95.102	96.887	334.682	94.614	97.243	329.652
TABU-H	94.737	96.663	333.404	94.091	96.355	332.061	93.816	97.161	313.892
T-H-M-1	93.654	95.855	328.654	93.237	95.196	345.419	93.164	96.220	317.175
T-H-M-2	94.168	96.449	339.598	93.453	95.378	349.074	93.378	96.169	319.309
SA-R-.1	96.389	98.932	33.844	95.011	97.180	33.825	94.671	97.346	33.806
SA-R-.3	95.941	98.340	33.835	94.572	96.871	33.803	94.316	96.812	33.792
SA-R-.5	95.744	98.094	33.801	94.547	96.377	33.771	94.002	96.791	33.769
SA-R-.7	95.623	97.663	33.783	94.538	96.179	33.746	94.142	96.963	33.724
SA-R-.9	95.722	98.473	33.751	94.837	96.966	33.714	94.373	97.420	33.718
SA-R-1.1	96.225	98.281	35.547	95.044	97.281	35.515	94.446	97.287	35.507
SA-R-1.3	95.689	98.016	35.495	94.613	96.759	35.470	94.142	96.608	35.447
SA-R-1.5	95.339	97.166	35.436	94.049	96.180	35.408	93.827	96.856	35.391
SA-R-1.7	95.372	97.955	35.367	94.538	97.283	35.362	94.018	96.866	35.336
SA-R-1.9	95.930	97.962	35.313	94.787	96.490	35.275	94.137	97.727	35.260
SA-R-2.1	96.061	99.265	35.236	94.729	96.360	35.169	93.962	97.299	35.180
SA-R-2.3	95.415	98.158	35.100	93.991	96.086	35.032	93.912	96.832	35.015
SA-R-2.5	95.196	98.673	34.985	94.066	95.776	34.932	93.333	96.473	34.906
SA-R-2.7	95.207	97.695	34.904	93.991	95.740	34.862	93.597	96.827	34.813
SA-R-2.9	95.941	98.758	34.830	94.273	96.247	34.762	93.923	97.000	34.736
SA-R-3.1	96.028	98.267	35.634	94.986	97.111	35.592	94.159	96.512	35.592
SA-R-3.3	95.251	97.929	35.514	93.967	96.170	35.460	93.591	96.467	35.443
SA-R-3.5	95.404	97.867	35.418	93.743	95.901	35.368	93.619	96.702	35.345
SA-R-3.7	95.207	97.451	35.345	94.132	96.239	35.272	93.810	96.682	35.256
SA-R-3.9	95.415	97.701	35.266	94.298	96.285	35.205	93.917	97.595	35.164
SA-H-.1	96.061	98.389	33.782	95.077	96.318	33.744	94.873	97.002	33.735
SA-H-.3	95.503	97.971	33.750	94.447	96.671	33.722	94.249	96.534	33.704
SA-H-.5	95.437	98.130	33.722	94.331	96.172	33.691	93.945	97.047	33.675
SA-H-.7	95.054	96.982	33.692	94.298	95.827	33.665	94.047	96.798	33.651
SA-H-.9	95.120	98.179	33.656	94.257	95.517	33.630	93.850	96.921	33.613
SA-H-1.1	95.076	97.150	35.564	94.688	97.091	35.532	94.198	96.981	35.517
SA-H-1.3	95.339	98.048	35.465	94.281	96.727	35.429	94.086	97.029	35.412
SA-H-1.5	95.251	97.604	35.641	94.174	96.564	35.330	93.844	96.767	35.319
SA-H-1.7	95.229	97.504	35.262	94.124	96.448	35.222	93.810	96.977	35.215
SA-H-1.9	94.890	96.988	35.166	94.049	96.299	35.133	93.760	96.785	35.126
SA-H-2.1	95.393	97.915	35.486	94.422	96.199	35.432	94.058	96.802	35.437
SA-H-2.3	95.306	97.452	35.385	94.124	96.297	35.337	93.726	96.439	35.322
SA-H-2.5	95.098	97.747	35.308	93.892	95.841	35.254	93.636	96.830	35.218
SA-H-2.7	95.021	98.058	35.230	93.967	96.440	35.182	93.602	96.628	35.145
SA-H-2.9	94.967	97.207	35.187	94.107	96.039	35.132	93.653	96.917	35.086
SA-H-3.1	95.733	97.643	35.564	94.588	96.062	35.524	94.052	96.696	35.507
SA-H-3.3	95.196	97.908	35.397	94.049	96.056	35.349	93.524	96.476	35.337
SA-H-3.5	94.901	97.635	35.275	93.801	95.998	35.219	93.580	96.244	35.185
SA-H-3.7	94.945	98.452	35.152	93.884	95.964	35.098	93.496	96.069	35.057
SA-H-3.9	95.054	97.698	35.051	94.025	96.620	34.973	93.653	96.802	34.954

Table B.8 Results for 8x8x8-problem size with makespan as the performance criterion

8 x 8 x 8 Problems									
Heuristic	**S/R = 2**			**S/R = 5**			**S/R = 10**		
	RELM	AFM	CPU	RELM	AFM	CPU	RELM	AFM	CPU
HITOMI	100.00	100.00	.165	100.00	100.00	.159	100.00	100.00	.152
HIT-M	100.82	101.28	.159	100.62	100.67	.161	100.56	100.60	.162
CDS	95.580	98.103	2.731	97.209	98.026	2.730	96.714	98.578	2.739
CDS-M-1	96.458	97.948	2.764	97.548	98.323	2.772	96.767	98.718	2.765
CDS-M-2	94.680	97.256	5.795	95.762	97.704	5.739	94.684	98.078	5.922
CDS-M-3	94.932	97.383	5.223	95.675	98.121	6.406	94.914	98.081	5.487
NEH	94.932	94.779	9.295	95.650	94.864	9.301	95.278	95.138	9.291
NEH-M-1	94.857	94.786	9.281	95.880	95.117	9.279	95.342	95.246	9.273
TABU-R	93.903	93.903	1740.430	93.243	96.243	1748.639	91.748	95.973	1698.847
T-R-M-1	92.888	96.344	1789.194	92.848	95.654	1754.686	91.451	95.228	1718.279
T-R-M-2	93.822	97.531	1748.662	93.320	96.785	1784.515	92.045	95.814	1733.074
TABU-H	92.827	96.567	1720.709	92.684	95.734	1718.779	91.599	94.802	1718.170
T-H-M-1	92.030	95.623	1753.431	91.915	94.888	1715.076	91.054	94.821	1703.673
T-H-M-2	92.425	95.907	1725.521	92.294	95.403	1740.152	91.242	94.951	1714.923
SA-R-.1	96.941	98.961	71.551	96.399	98.518	71.602	95.193	97.911	71.598
SA-R-.3	95.811	98.441	71.496	95.372	97.725	71.551	93.930	97.253	71.551
SA-R-.5	95.525	98.248	71.437	95.280	98.213	71.511	93.828	97.299	71.497
SA-R-.7	95.150	98.653	71.383	94.942	98.179	71.436	93.375	96.847	71.444
SA-R-.9	95.450	98.660	71.309	95.024	97.253	71.375	93.446	97.360	71.373
SA-R-1.1	95.899	99.147	74.565	96.137	98.315	74.624	95.370	96.961	74.623
SA-R-1.3	95.109	98.568	74.532	95.347	97.934	74.589	93.881	97.700	74.586
SA-R-1.5	95.273	98.791	74.473	94.926	97.460	74.540	93.531	96.724	74.538
SA-R-1.7	95.123	98.312	74.435	94.967	97.800	74.488	93.325	96.537	74.482
SA-R-1.9	95.252	97.837	74.377	94.777	98.149	74.423	93.032	96.895	74.435
SA-R-2.1	96.158	99.827	75.135	95.993	98.909	75.164	94.984	97.442	75.159
SA-R-2.3	95.116	98.722	74.897	94.721	98.112	74.915	93.460	96.757	74.866
SA-R-2.5	94.762	98.447	94.762	94.146	97.408	74.702	92.866	97.039	74.641
SA-R-2.7	94.850	98.475	74.515	94.562	97.274	74.530	92.897	97.334	74.458
SA-R-2.9	94.728	98.563	74.340	94.552	96.782	74.361	92.837	97.196	74.317
SA-R-3.1	95.988	98.968	76.127	96.091	98.799	76.144	94.602	97.521	76.157
SA-R-3.3	95.102	98.382	76.004	94.911	97.157	76.022	93.446	96.879	75.982
SA-R-3.5	94.762	97.831	75.910	94.593	97.692	75.942	92.862	96.177	75.893
SA-R-3.7	94.857	98.289	75.873	94.259	97.837	75.872	92.724	96.160	75.843
SA-R-3.9	94.925	97.643	75.822	94.403	97.295	75.831	92.851	96.198	75.784
SA-H-.1	95.838	98.783	72.156	95.937	98.288	72.224	94.708	96.971	72.216
SA-H-.3	95.238	98.813	72.120	95.162	97.519	72.187	93.898	97.052	72.198
SA-H-.5	95.068	98.217	72.073	94.654	97.597	72.138	93.424	97.036	72.144
SA-H-.7	94.986	97.986	72.029	94.536	96.588	72.097	93.103	97.262	72.100
SA-H-.9	94.898	98.018	71.991	94.685	97.266	72.058	93.156	97.409	72.045
SA-H-1.1	95.879	98.186	74.967	95.650	97.915	75.045	94.652	97.900	75.032
SA-H-1.3	95.518	98.932	74.879	95.003	97.237	74.942	93.615	97.066	74.951
SA-H-1.5	95.252	99.140	74.801	94.649	97.242	74.851	93.442	96.107	74.864
SA-H-1.7	94.939	98.348	74.710	94.444	96.775	74.759	93.035	96.430	74.758
SA-H-1.9	94.993	98.604	74.598	94.423	97.342	74.681	93.035	96.347	74.679
SA-H-2.1	95.593	98.687	75.068	95.367	98.007	75.103	94.496	97.315	75.073
SA-H-2.3	94.898	98.520	74.848	94.577	97.691	74.858	93.764	96.442	74.803
SA-H-2.5	94.762	97.812	74.670	94.387	97.447	74.670	93.067	96.315	74.633
SA-H-2.7	94.659	97.951	74.521	94.280	96.844	74.529	92.869	95.829	74.482
SA-H-2.9	94.694	98.271	74.389	94.429	97.453	74.395	92.745	96.541	74.662
SA-H-3.1	95.715	98.089	75.331	95.378	97.697	75.386	94.153	97.265	75.349
SA-H-3.3	94.918	97.835	75.130	94.603	97.630	75.145	93.092	96.873	75.119
SA-H-3.5	94.700	98.119	74.989	94.485	97.116	74.985	92.862	96.508	74.941
SA-H-3.7	94.755	98.290	74.843	93.977	97.022	74.872	92.699	96.313	74.824
SA-H-3.9	94.414	98.019	74.771	94.362	97.335	74.770	92.717	96.176	74.720

APPENDIX C

SELECTED PARTS SAMPLE FOR THE CASE STUDY

Tables C.1 lists the sample of parts, and the machines currently used for their processing, employed in the case study described in Chapter 5. The figures in body of the table are the operations sequence for each part on the necessary machines for it.

Table C.1 Selected parts and machines currently used for their processing

See Excel Files

See Excel Files

See Excel Files

See Excel Files

See Excel Files

See Excel Files

See Excel Files

APPENDIX D

BEST HEURISTICS' CODE LISTINGS

Codes for the best performing GS heuristic versions studied in the research are listed below. A complete code for the original Hitomi is provided. Then the other heuristics are presented without the subroutine MyComputeMakespan that is found in Hitomi's listing. This is the proposed timetabling procedure in Sec. 3.4.2. The subroutine is the same in the others so it need not be repeated. Similarly, typical subroutines used in different heuristic codes are not repeated.

The listings assume problem sizes of ixjxk of 3x4x5. Statements for data entry and printing of results are not presented completely but are reduced as follows.

```
INPUT #2, P(1, 1, 1), P(1, 1, 2), ... , P(1, 5, 4)
INPUT #2, P(2, 1, 1), P(2, 1, 2), ... , P(2, 5, 4)
INPUT #2, P(3, 1, 1), P(3, 1, 2), ... , P(3, 5, 4)
INPUT #2, Setup(1, 1), Setup(1, 2), ... , Setup(3, 4)
```

These statements mean reading processing times for each job ; P(i, j, k), a line for each family. The last line is reading the setup time for each family i on each machine k.

See Basic codes

ARABIC SUMMARY

ملخص الرسالـة

يطلق مصطلح جدولـة المجموعـات على نمـاذج الجدولـة التـي تقسم فيهـا الأجـزاء إلـى عـائلات تبعـا لمبادئ تقنية المجموعات Group Technology و قد أدي هذا الأسلوب إلى استحداث نموذج لمسائل جدولة الإنتاج من مرحلتين : أولا جدولة العائلات و ثانيا جدولة الأجزاء في كل عائلة.

فوائد مثل هذا الأسلوب تشمل تقليل أزمنة الإعداد و تبسيط عملية الجدولة بشكل عام ، كذلك فـإن هذا الأسلوب مناسب للاتجاهات الحالية في نظم التخطيط و التحكم فـي عمليـات الإنتـاج و التـي تشير إلـى اتجاهـات للتحـول مـن الإنتـاج الكمـي إلـى نظـم الـدفعات Batch production system و كـذلك زيـادة التنـوع فـي المنتجات و دورة عمر أقصر للمنتج و التوسع في استخدام مبادئ تقنية المجموعات و خلايا الإنتاج.

يدرس هذا البحث نموذج جدولة المجموعات في خلية إنتاجيـة انسيابية استاتيكية ، مخصصـة لتشغيل عدد من العائلات ، و قد تمت دراسـة عـدد مـن النمـاذج التنقيبيـة لجدولـة المجموعـات ، باستهداف تقليل الـزمن الكلي Makespan و مجموع نهايات أزمنة تشغيل الأجزاء Total flow time ، كل على حدة.

جـرى اقتـراح عـدد مـن التعـديلات علـى النمـاذج المدروسـة ، مـن أجـل استكشـاف إمكاناتهـا و كـذلك استكشاف خصـائص نمـوذج جدولـة المجموعـات ، و إضـافة إلـى هـذا فقـد اقتـرح نمـوذج حسـاب أزمنـة البـدء و الانتهاء و الزمن الكلي للتشغيل ، في خلية مخصصة لعدد من العائلات مع وجود أزمنة تشغيل صـفرية لـبعض العمليات . كما أجريت دراسة ميدانية استهدفت التعرف على مدى قابلية نموذج جدولة المجموعات للتطبيـق فـي ظروف نظام إنتاج تقليدي قائم بالفعل و يعمل بأسلوب الدفعات ، و ذلك بدون تكوين خلايا إنتاجية .

و قد بينت النتائج النظرية تحسن أداء نماذج الجدولة المدروسـة باسـتخدام التعـديلات المقترحـة فـي هـذه الرسالة ، وظهرت من النتائج أهمية مراعاة التفاعل و التأثير المتبادل بين مرحلتـى الجدولـة فـي نمـوذج جدولـة المجموعات و ضرورة أخذ هذا التفاعل في الاعتبار عند تكوين نماذج الجدولة التنقيبية لجدولة المجموعات.

و قد و جد أن الأسـاليب التكرارية هي الأفضل مـا بين النمـاذج المدروسـة ، لـيس فقط لكونهـا الأفضـل آداءا و لكن لقدرتها علـى مراعـاة تفاعـل المـرحلتين كذلك , و وجـد أن نمـاذج Tabu search المعدلـة فـي هـذا البحـث هـي الأفضـل فـي حالـة الـزمن الكلـي Makespan بينمـا كانـت أسـاليب Simulated annealing المعدلـة كذلك ، هـي الأفضـل فـي حالـة الــ Total flow time ، و لكن نظـرا لان أسـاليب Simulated

annealing يسوء أداؤها مع زيادة حجم المسألة ، بينما تستطيع أساليب Tabu search الاحتفاظ بمستواها فقد اعتبر أسلوب Tabu search هو المفضل في هذا في الحالتين .

و ظهر من دراسة النتائج أن الـ Tabu search يحتاج إلى إعادة تعريف الذاكرة طويلة المدى Long Term Memory (LTM) المستخدمة فيه و أن تحتوي الـ LTM على معلومات أخري مستنتجة من عملية البحث بخلاف ما هو موجود في الـ Tabu search المقدم في [12] ، كما أنه يجب استخدام LTM في كل من مرحلتى الجدولة.

و بالنسبة للـ Simulated annealing فقد أظهرت النتائج أن استخدام دالة احتمال قبول ترتبط بالتغير في قيمة دالة الهدف ، هو الأفضل من كونها غير معتمدة على التغير في قيمة دالة الهدف ، و أن الـ Simulated annealing يحتاج إلى إضافة درجة من التحكم في تأثير الأرقام العشوائية في أدائه .

كما أوضح البحث ضرورة مراعاة احتمال وجود الأزمنة الصفرية في عمليات حساب أزمنة ابتداء و انتهاء تشغيل الأجزاء و حساب الزمن الكلي ، و قد ظهرت قدرة نموذج الحسابات المقترح على مراعاة ذلك في حالة وجود عدد من عائلات الأجزاء ، بينما لو حظ أن محاولة مراعاة وجود الأزمنة الصفرية في بنية أساليب الجدولة لا تبدو مجدية.

و بينت مراعاة الأزمنة الصفرية أنه قد لا يصح دائما ، تعريف الزمن الكلي ، بأنه الفترة الزمنية من بداية تشغيل أول جزء على أول ماكينة إلى نهاية تشغيل أخر جزء على آخر ماكينة ، و الأكثر صحة من هذا أن يعرف الزمن الكلي بأنه أكبر زمن انتهاء ، حيث وجد أن الزمن الكلي لا يرتبط اشتراطا بآخر جزء أو آخر ماكينة ، و من ناحية أخري وجد أن استهداف الزمن الكلي يؤدي إلى مجموع أزمنة نهايات الأجزاء Total flow time جيد ، بينما العكس غير صحيح .

و أخيرا فقد بينت الدراسة الميدانية أنه من الممكن تطبيق نموذج جدولة المجموعات في نظام إنتاج تقليدى يعمل بأسلوب الدفعات ، بدون تكوين خلايا إنتاجية في الواقع ، و هو ما يعني لإمكان تحقيق مزايا تقنية المجموعات بدون استثمارات مالية كبيرة ، و للوصول إلى ذلك فإنه يتعين مراعاة إيجاد عائلات الأجزاء في المراحل الأولى لإعداد صفحات التشغيل للأجزاء , في حين أن محاولة التحول إلى نموذج جدولة المجموعات ابتداءا من أمر واقع و من صفحات تشغيل قائمة فهو أسلوب لا يمكن ضمان نتائجه.

وزارة التعليم العالي
المعهد العالي للتكنولوجيا ببنها
قسم تكنولوجيا الهندسة الميكانيكية

دراسة أساليب جدولة المجموعات في خلية انسيابية

رسالة ماجستير مقدمة من

مجدي السيد عبد الرحمن هلال
بكالوريوس الهندسة و التكنولوجيا (تكنولوجيا التصنيع)

أجيزت من :

أ.د. فاروق أحمد الجيار
أستاذ الهندسة الصناعية
كلية الهندسة – جامعة حلوان

أ.د. عبد اللطيف محمد عبد اللطيف هريدى
أستاذ الهندسة الصناعية
كلية الهندسة – جامعة عين شمس

أ.د. أحمد سليمان حزين
أستاذ الهندسة الميكانيكية
عميد المعهد العالي للتكنولوجيا ببنها

د. ميرفت عبد الستار بدر
قسم الهندسة الميكانيكية
المركز القومي للبحوث - القاهرة

1999

وزارة التعليم العالي
المعهد العالي للتكنولوجيا ببنها
قسم تكنولوجيا الهندسة الميكانيكية

دراسة أساليب جدولة المجموعات في خلية انسيابية

رسالة مقدمة من

مجدي السيد عبد الرحمن هلال

إلى المعهد العالي للتكنولوجيا ببنها كجزء من متطلبات الحصول على

درجة الماجستير في تكنولوجيا الهندسة الميكانيكية

تحت إشراف

أ.د. أحمد سليمان حزين

أستاذ الهندسة الميكانيكية و عميد المعهد العالي للتكنولوجيا ببنها

د. ميرفت عبد الستار بدر

الباحثة بقسم الهندسة الميكانيكية بالمركز القومي للبحوث بالقاهرة

د. هاني محمود عساف

الأكاديمية الحديثة لعلوم الحاسب و للإدارة – المعادي - بالقاهرة

1999