

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]



Université d'Ottawa • University of Ottawa

Modelling and Analysis of Integrated Machine-level Planning Problems for Automated Manufacturing

A thesis submitted to the
Faculty of the Graduate Studies and Research
as a partial fulfilment of the requirements for the degree of
Doctor in Philosophy in Mechanical Engineering

by

Farhad KOLAHAN

Ottawa-Carleton Institute for Mechanical and Aeronautical Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5

© Farhad Kolahan, Ottawa, Ontario, Canada, 1999



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-46527-6

Canada

ABSTRACT

The wide application of NC technologies and the advance in integrated manufacturing have significantly improved productivity. This however also leads to complexity in shop floor planning, particularly in machine-level planning due to the increased flexibility in the selection of machining parameters, tools, and tool paths. Part sequencing, tool replacement and machining speed selection for metal cutting in general and tool set selection, machining speed specifications as well as path sequencing for hole making in particular have a direct impact on manufacturing economics and hence have drawn much attention of many researchers. However, these problems have been often solved in isolation to each other, thereby causing inconsistent and conflicting planning actions on the shop floor. As a result, the solutions obtained in such a way can not be used for real shop floor planning. At the best a lengthy process is needed to resolve the conflict between the separately obtained solutions. Such a decision process obviously does not meet the need of modern manufacturing environment where quick and consistent planning decisions are imperative.

The main purpose of this study is to model and solve several combined planning problems facing today's manufacturing industry. These include a) part sequencing and tool replacement with sequence-dependent setup times and probabilistic tool life; b) Just-In-Time (JIT) part scheduling with variable processing times and sequence dependent setups; and c) tool set selection, machining speed specification, operation sequencing and path selection for hole making operations. These problems are combinatorial in nature and are often classified as NP-complete. Consequently, optimal solutions may not be

obtained within polynomial times. In this dissertation, tabu search technique has been employed to solve the above combined planning problems. The computational results have shown that these problems can be efficiently solved and consistent decisions can be made based on the solutions. The effects of some important parameters such as initial solutions, move selection, termination criteria, and tabu list size on the search performance have also been examined.

ACKNOWLEDGMENTS

The preparation of this thesis has been a long and demanding task, disrupted by several unfortunate incidents including the departure of my beloved father from this world. His death at the time that I have just completed my doctoral work marks a great loss for me and my family. However, one of the most satisfying moments in putting the final touches to this dissertation is having the opportunity to express my sincere appreciations to the people who have supported me during the course of this study. In that respect, I am deeply grateful to my supervisor Dr. M. Liang for not only his outstanding supervision and continuous guidance but also for his caring and friendship which go far beyond the professional obligations. I also wish to thank the members of my advisory committee, Dr. M. Chen, Dr. B. Dhillon, Dr. J.M. Thizy and Dr. D. Necsulescu, for their helpful comments and suggestions.

The financial supports for this research provided by the Government of Islamic Republic of Iran are gratefully acknowledged.

Finally, my most sincere thanks are due to those who are very dear to me - my parents and my family. Their prayers and encouragements have contributed much to the completion of this thesis. I am particularly indebted to my beloved but much-neglected son, Ali, and to my wonderful and supportive wife, Toktam, who sacrificed their own comfort and gave me unconditional love and support throughout this period.

And above all, many thanks to God who makes everything possible.

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
NOMENCLATURE	ix
Chapter 1 INTRODUCTION	1
1.1 Overview	1
1.2 Objectives and expected contributions	2
1.3 Selection of the solution technique	3
1.4 Concepts and definitions	4
1.5 Organization of the thesis	6
Chapter 2 LITERATURE REVIEW	7
2.1 Machine scheduling for advanced manufacturing	7
2.1.1 Classic sequencing problems	8
2.1.2 Single machine planning based on completion time	9
2.1.3 Multiple criteria scheduling problems	13
2.1.4 Scheduling with tooling consideration	15
2.1.5 Tool life and tool replacement policies	18
2.1.6 Scheduling with controllable processing times	21
2.1.7 Operation sequencing for hole making	23
2.2 Tabu search	25
2.2.1 Tabu search methodology	25
2.2.2 Applications of tabu search in machine scheduling	28
2.2.3 Refinement of tabu search method	32

Chapter 3 OPERATION SEQUENCING WITH TOOL REPLACEMENT

CONSIDERATION	35
3.1 Background	35
3.2 Problem statement	37
3.3 Problem formulations and analysis	39
3.3.1 Planning for minimization of expected production cost . . .	39
3.3.2 Planning for minimization of expected production cost and JIT sequencing	41
3.3.3 Planning for minimization of expected production cost and flow time cost	43
3.4 Solution procedure	46
3.4.1 Development of tabu search algorithm	47
3.5 Illustrative examples and results	50
3.5.1 Example problem for Model 3-1	53
3.5.2 Example problem for Model 3-2	57
3.5.3 Example problem for Model 3-3	60
3.6 Discussions	62
3.7 Performance of the tabu search algorithm	65

Chapter 4 OPERATION SEQUENCING WITH PROCESSING TIME

ADJUSTMENT	70
4.1 Background	70
4.1.1 Scope and purpose	70
4.1.2 Process planning and JIT goal	71
4.2 Problem statement	72
4.3 Problem formulation and analysis	74
4.3.1 Planning for JIT production	74
4.3.2 Planning for minimization of weighted flow time	76
4.4 Tabu search algorithm for JIT sequencing	79
4.4.1 Single sequence cost minimization model	82
4.4.2 Alternative solution procedure for Model 4-3	83

4.5	Tabu search algorithm for minimization of weighted flow time	90
4.6	Computational results and discussions	92
4.6.1	Example problem for Model 4-1	92
4.6.2	Example problem for Model 4-2	96
4.6.3	Discussions	98
4.7	Refinement of the tabu search algorithm	104
4.7.1	Neighbourhood generation and move selection	104
4.7.2	Adaptive neighbourhood generation (ANG)	106
4.7.3	The improved algorithm	110
4.7.4	Comparison of neighbourhood generation methods and discussions	113
Chapter 5	PLANNING FOR HOLE-MAKING OPERATIONS	118
5.1	Background	118
5.1.1	Scope and purpose	118
5.1.2	Issues in hole making operations	119
5.2	Problem statement	120
5.3	Problem formulation	125
5.4	Tabu search algorithm for hole making problem	127
5.5	Case study	130
5.6	Discussions	136
5.6.1	Solution	136
5.6.2	Search performance	138
5.7	Improvement of the tabu search algorithm	139
Chapter 6	SUMMARY AND FUTURE TASKS	144
6.1	Summary and conclusions	144
6.2	Recommendations for future studies	147
REFERENCES		149

LIST OF FIGURES

Figure 3-1	Process reliability and expected production cost with different tool spare levels	64
Figure 3-2	Convergence curves for the 20-tool example problem	66
Figure 3-3	Convergence curves for the 100-tool example problem	67
Figure 4-1	Convergence curves for the Model 4-1 example problem with different starting sequences	101
Figure 4-2	Piece-wise convergence curve when search restarts from a random sequence	103
Figure 4-3	Transposition range and objective value	109
Figure 4-4	Convergence curves for different neighbourhoods	116
Figure 5-1	A schematic representation of alternative sets of tools for hole making	121
Figure 5-2	Upper holder of the plastic injection mould (courtesy of Komtech Plastics Corp.)	131
Figure 5-3	Convergence curves for different neighbourhood mechanisms	142

LIST OF TABLES

Table 3-1	Operation times (min.) for the example problems	51
Table 3-2	Setup times for the example problems (min.)	52
Table 3-3	Tool and raw material cost data (\$)	53
Table 3-4	Solution for the Model 3-1 example problem with different spares .	55
Table 3-5	Tool replacement intervals and spare requirements ($M_i=6$)	56
Table 3-6	Due dates and cost data for the Model 3-2 example problem	57
Table 3-7	Solution for Model 3-2 example problem ($M_i = 8$)	59
Table 3-8	Job flow cost for Model 3-3 example problem	60
Table 3-9	Solution for Model 3-3 example problem ($M_i = 8$)	61
Table 3-10	The effect of tabu-list size on search performance	68
Table 3-11	The effect of path diversification	69
Table 4-1	Comparison between Algorithm 4-2 and LINDO	89
Table 4-2	Processing times, due dates, and cost data for the example problem	93
Table 4-3	Solution to the Model 4-1 example problem	95
Table 4-4	Solution for the Model 4-2 example problem	97
Table 4-5	Final costs for different starting sequences and diversification strategies	100
Table 4-6	The effect of tabu-list size on search performance	104
Table 4-7	Comparison of different neighbourhood generation methods	115
Table 4-8	Average costs for the ANG method with different diversification strategies (tabu-list size=10, phase size=20)	117
Table 5-1	Tool diameter, cost data, and specified feed rate	133
Table 5-2	Tool switch times (min.)	133
Table 5-3	Possible tool-hole combinations used in the example problem	134
Table 5-4	Operations sequence and their corresponding cutting speeds	136
Table 5-5	Operations sequence and their cutting speeds for the example problem when tool switch costs are reduced by 50%	137

NOMENCLATURE

i	tool type index (in ascending order according to tool diameter in case of drilling) $i=1,\dots,I$
j,k	job index (hole index in case of drilling), $j=1,\dots,J$, $k=1,\dots,J$
l,ll	position index in a sequence
s	job sequence index, denoting a specific permutation of J jobs
α_j	cost of reducing processing time of job j by one unit (compression cost)
a_j	penalty cost per unit time tardiness for job j
d	cost per unit non-productive travelling distance
β_j	cost of increasing processing time of job j by one unit (extension cost)
b_j	penalty cost per unit time earliness for job j
A_j, B_j, H_j	elements in the saving rate function for job j
c	setup cost per unit time (tool switch cost per unit time in case of drilling)
C	machining cost per unit time
$\hat{C}_{ij}$	combined tool and machining costs when tool type i is used on hole j
d_j	due date of job j
D_i	diameter of tool i
$\bar{D}_j$	final size of hole j
L_j	depth of hole j , including the clearance
e_{ij}	depth of cut when tool type i performing a cutting operation on hole j
E_j	earliness of job j
f_i	recommended feed rate for tool type i
F^n_j	cost saving gained by increasing processing time of job j by one unit in iteration n
F_j	cost saving gained by decreasing processing time of job j by one unit
$G(s)$	total cost associated with sequence s
$h_{ij}(t)$	hazard rate when tool i is used for job j
I_j	a set of tools required for machining operations on job j
$\bar{I}_j$	a set of tools sorted in ascending order for hole j

J	total number of jobs
$\mu(s,l)$	the job associated with the l th position in sequence s
m_{ij}	a tool replacement indicator; if tool i is replaced immediately before processing job j , $m_{ij}=1$; otherwise, $m_{ij}=0$
M_i	available number of spare tools of type i
N_j	number of tools in $\bar{I}_j$
$O(i,j)$	the index of the operation performed by tool i on job j
Π_{ij}	cumulative dollar value on job j up to the time when the operation performed by tool i is completed
π_j	raw material cost of job j
p_{jk}	non-productive travelling distance between hole j and hole k
P_j	current processing time of job j
P_j^N	normal processing time of job j
P_j^L	minimum possible processing time of job j
P_j^U	maximum allowed processing time of job j
ΔP_j^n	time increment for job j in iteration n
Q_i	cost of a tool of type i
r_{ij}	reliability of tool i at the end of the operation for job j if it is not replaced immediately before processing job j
R_{ij}	reliability of tool i at the end of the operation for job j if it is replaced immediately before processing job j
RT_j	waiting time for part j
ρ_j^n	maximum allowable time increase for job j in iteration n ; $0 \leq \rho_j^n \leq P_j^U - P_j^L$
t_{0j}	setup time required by job j if it is the first job processed in the sequence
t_{kj}	setup time required by job j if it is processed immediately before job k
$t_{i'',j}$	tool switch time between current tool type, i'' , and tool i required by hole j
T_j	tardiness of job j
$\bar{T}_{ij}$	life of tool type i performing cutting operation on hole j
$\bar{T}_{ij}^*(s,l)$	optimum life of tool type i performing a cutting operation on hole j in l th position of sequence s

$\tau_i(l)$	a set of jobs completed by the current tool of type i after processing the job in l th position of a sequence
T_{ij}	machining time required by tool i for job j (hole j in case of drilling)
$T_{ij(s,l)}^*$	optimum processing time of operation ij located in l th position of sequence s
$\bar{U}$	total number of possible operations
u_j	cost due to one unit flow time for job j
V_{ij}	cutting speed of tool i performing an operation on hole j
V_{ij}^*	optimum cutting speed for tool i operating on hole j
W_s	a 0-1 integer variable, $W_s = 1$, if sequence s is selected; 0, otherwise
$x_{i'i''jk}$	a 0-1 integer variable, $x_{i'i''jk} = 1$ if tool i replaces tool i'' to drill hole j which is located in the path between holes l and k and has been drilled by tool i' ; 0, otherwise
X_j	amount of compressed processing time of job j ; $0 \leq X_j \leq P_j^N - P_j^L$
Y_j	amount of extended processing time of job j ; $0 \leq Y_j \leq P_j^U - P_j^N$
$N(s)$	set of job sequences in the neighbourhood of s
$N_a(s)$	set of job sequences in the adaptively generated neighbourhood of s
$ N(s) $	total number of job sequences in the neighbourhood of s
C_{best}	the lowest objective function value obtained so far
S_{best}	the best job sequence found so far
λ_{best}	the maximum cost reduction rate for the ANG method
M_{max}	maximum allowable number of moves
M_{ctr}	a counter used to record the total number of moves made so far
N_{max}	allowable number of moves in each search phase
N_{ctr}	a counter used to record the number of moves made so far in each phase
T_{max}	maximum allowable search time
T_{list}	tabu list
T_{size}	tabu-list size
M_{big}	a big integer number
s^*	the best job sequence in the current neighbourhood
s^{**}	the best job sequence in the immediate previous neighbourhood

Dedicated

to my wife, Toklam; and my son, Ali

and

to my mother and the memory of my father

Chapter 1

INTRODUCTION

1.1 Overview

In the technologically advanced world that we are in today, the rapidly changing market demands and intense competition in manufacturing have revealed the need for production systems with high flexibility and efficiency. The ever increasing popularity of single machine facilities in recent years is mainly due to their high flexibility and productivity. Single machine facilities include a wide range of stand-alone processing units from single machining centres to assembly lines. Any manufacturing firm that is not engaged in mass production of single items could have "single machine scheduling problems" to a certain extent. A classic example of this scenario would be a job shop with multiple machines where one critical machine poses a bottleneck in production.

An automated machining centre represents heavy capital investment which can be justified only when the machining centre is operated efficiently. This greatly depends on planning decisions. The single machine problem appears to be simple in the traditional manufacturing environments. In the advanced manufacturing context, however, due to wide variety of parts to be processed, the large number of tools required by the parts, and various planning goals, providing quick and consistent solutions to all the planning problems becomes a very challenging task. This has not been addressed adequately in the literature and hence gives rise to the work reported in this thesis.

1.2 Objectives and expected contributions

Planning for effective utilization of existing production systems has always been a challenging issue. In recent years, machine scheduling problems have been extensively studied by researchers. Many mathematical models and solution procedures have been developed to seek both optimum and approximate solutions for single machine sequencing and scheduling problems.

Although there has been an enormous research effort over the years in machine scheduling area, the outcomes have had little effect on the real scheduling activities. This could be due to the fact that most of the existing approaches are typically (a) too slow to react to the changing shopfloor conditions, (b) based on over-simplified formulations and lacking a realistic representation of the problem in hand, and (c) based on a single objective function that ignores important problem specifications and constraints. The objectives of this research are therefore to develop integrated modelling approaches and efficient solution algorithms for the following problems and their variations:

- 1) The combined job sequencing and tool replacement problem;
- 2) The combined JIT job scheduling and processing time allocation problem; and
- 3) The combined tool selection, machining speed optimization and operation sequencing decisions for hole making.

For each problem, different search parameters will be tested and their effects in terms of solution quality and computational time will be examined. This will help to develop a general guideline for the selection of the appropriate parameter settings for each problem.

1.3 Selection of the solution technique

As stated above, this research focuses on two major aspects of the single machine planning problems:

- a) integrating several important planning problems to avoid infeasible or conflicting operational decisions, and
- b) developing an efficient solution procedure for solving these integrated problems.

The selection of the solution procedure is therefore based on the following considerations. First, it is attempted to solve different problems using a good solution procedure rather than solve a single problem using many different techniques. Secondly, instead of spending a large amount of time to search for optimal solution, it is desirable to quickly find good and consistent solutions.

In this study, tabu search, one of the most widely used search techniques, is selected as the solution method for all the problems analyzed in this research. The main reasons are:

- I) Tabu search is one of the least sensitive search procedures and requires very little tuning. Many other modern heuristics such as simulating annealing are sensitive to a number of parameters. Considerable amount of tuning time and many search trials are often needed to find a good solution. In tabu search, on the other hand, tabu list size is probably the only parameter that may affect the solution quality. The effect of tabu list size is not very significant on the solution either in the

computations carried out in this thesis. Therefore, tabu search is relatively more robust than its counterparts. As a result, the lengthy tuning process can be avoided. This is particularly important in today's dynamic manufacturing environment.

- II) It can easily be implemented to solve scheduling problems with different objective functions and constraints.
- III) Tabu search has been reported to outperform other methods in terms of solution quality and/or computational time in solving similar problems (e.g. Skorin-Kapov and Vakharia 1993; Reeves 1993; Hao *et al.* 1996; Wen and Yeh 1997; and Ouenniche and Bector 1998).

1.4 Concepts and definitions

Over the past few decades, much research has been carried out on single machine planning problems. Due to the diversification of existing literature on planning problems, it is necessary to provide some of the definitions and classification schemes used in this area to facilitate discussions in the literature review and in the model development. Some of the following definitions are adopted from French (1982).

Sequencing: Sequencing is basically the task of determining the order in which a sequence, or a permutation of jobs, is processed. The processing sequence contains no explicit information about the times at which each operation starts and finishes.

Scheduling: In addition to sequencing information, scheduling also provides the start and completion times of each operation. Therefore, a schedule includes both

sequencing and timetabling information. It should be noted that in literature the above terms have often been used interchangeably.

Performance measure: A performance measure usually indicates a specific (but not necessarily unique) objective in scheduling or sequencing. There are numerous criteria used as the measures of performance in scheduling literature. The time based performance measures, however, are the most widely used in literature. This could include the criteria based on due dates such as total earliness or tardiness and number of tardy jobs or measures based on completion times such as mean flow time and total production time (makespan).

Performance measures can be further classified as *regular* and *non-regular*. A regular measure of performance is defined as a monotonic function of the completion time whereas a non-regular measure may or may not be monotonic in terms of completion time. Regular performance measures such as number of tardy jobs, maximum tardiness, and makespan, have been used in traditional scheduling problems. The non-regular performance measures are often related to the scheduling problems emerging in the advanced manufacturing environment. The single machine weighted earliness-tardiness (SMWET) is a typical problem with non-regular performance measure.

Computational complexity: Basically scheduling problems, in terms of computational complexity, are classified into two categories: **P** and **NP** classes. The problems that can be optimally solved in polynomial time are known as class **P** problems. The class **NP** consists of those problems for which only algorithms with exponential behaviour have been found. In other word, for **NP** problems there is no known algorithm

that can generate guaranteed optima in an execution time that may be expressed as a finite polynomial of the problem dimension. Usually, by convention, only those algorithms with polynomial time behaviour are considered to be computationally efficient. In literature, heuristic algorithm has often been proposed to solve NP problems.

In many cases, manufacturing planning problems are classified as system level, machine level, and part level problems. Facility layout and machine grouping are the examples of system level planning. At the machine level, job sequencing and tool provision issues are among the most widely studied problems. Finally, tool path planning and operation sequencing can be considered as the part level planning problems.

1.5 Organization of the thesis

In the next chapter a review of existing research in scheduling and sequencing is presented. This is followed by a detailed elaboration of tabu search method, its main components, and its applications in scheduling. Chapters 3 to 5 are dedicated to the development of the models and required algorithms to implement them. The computational results of the proposed search method for each model are also presented in the same chapters. Finally, Chapter 6 summarizes the results accomplished in this study and outlines some recommendations for future research.

Chapter 2

LITERATURE REVIEW

The machine level planning problems have been extensively studied over the years. In this chapter, relevant literature on single machine scheduling problems will be reviewed. The survey is organized in accordance with the problems addressed in this research. In the following, the most prominent literature on the single machine scheduling with different performance measures will be reviewed. This includes an overview of related research on tool management problems and optimization of hole making operations. A brief review of tabu search technique and its applications in scheduling problems is also presented in this chapter.

2.1 Machine scheduling for advanced manufacturing

Machine scheduling is one of the richest and most promising fields in operations research with a vast application in automated manufacturing environments. During the past few decades, different classes of these problems have been identified, based upon their performance measures and computational complexities. Some well-known measures are the sum of completion times, the maximum tardiness, and the total weighted tardiness and earliness. Depending on the performance measures and the computational complexities, the solution procedures for these type of planning problems may range from optimal algorithms to approximate heuristics.

2.1.1 Classic sequencing problems

The great deal of research efforts has resulted in efficient optimal algorithms for some traditional scheduling problems. For instance, Jackson (1955) showed that sequencing jobs according to their earliest due date (EDD) will minimize the maximum tardiness. Similarly, if inserted idle time is not allowed, the maximum earliness is minimized when jobs are sequenced in a nondecreasing order of their slack times (MST rule). The slack time for a job is defined as the difference between the job's due date and its processing time. Smith (1956) proved that to minimize mean flow time, it is sufficient to sequence them based on the shortest processing time (SPT rule), i.e., jobs are sequenced in increasing order of their processing times. There are many situations in which the penalty for a late job remains the same regardless of how late that job may be. For such cases, Moore (1968) developed an exact algorithm that minimizes the number of tardy jobs for the single machine problem.

However, the majority of machine scheduling problems cannot be solved efficiently and are still open to explore. In addition, the changing market demands and the current trend of industry towards the implementation of more sophisticated production philosophies such as just-in-time (JIT) production, have introduced a range of problems with different specifications and performance measures. The class of sequencing problems with earliness and tardiness penalties as the performance criterion is a typical example of these problems. The following sections will mainly focus on the research efforts on these types of the problems.

2.1.2 Single machine planning based on completion time

The scheduling problems involving completion time and due date may be the most extensively studied planning problems. The main criteria in this category include maximum tardiness, total earliness and tardiness, mean flow time, and total flow time.

Among these problems, single machine scheduling with common due date has received considerable attention in literature. Bector *et al.* (1991) developed a constructive algorithm so as to minimize a penalty function by determining the optimal sequence of jobs and their corresponding optimal due dates. The proposed linear cost function consists of weighted sum of earliness, tardiness and common due date assignment. Sarin *et al.* (1991) analyzed the common due date sequencing problem with stochastic job processing times. The objective is to minimize the sum of expected incompleteness costs resulted from tardy jobs. The nonlinear loss function used in this procedure, however, fails to account for earliness penalties and is inefficient to provide optimal solution for large size problems. De *et al.* (1993) developed a solution procedure based on dynamic programming to minimize the earliness and tardiness penalties of jobs with respect to their common due dates. Zheng *et al.* (1993) analyzed the problem of single machine scheduling to minimize the total cost of flow time, earliness and tardiness where all jobs have a common due date. A branch and bound algorithm was proposed to solve this problem. Later on, Weng and Ventura (1994) showed that some of the properties developed by Zheng *et al.* (1993) for an optimal schedule may not necessarily hold. They provided some modifications that guarantee the validity of these properties and enhance the solution procedure in finding the final solution. Lee and Kim (1995) proposed a

genetic algorithm approach to minimize the sum of weighted earliness and tardiness penalties from a common due date. They conducted comprehensive computational tests and showed that genetic algorithm provided good solutions in a reasonable time. Recently, Lam and Cai (1998) studied the problem of scheduling n jobs on a single machine to minimize the weighted earliness and tardiness of job completions from a common due date which is a fuzzy number governed by a triangular membership function. A fuzzy distance function is introduced first to measure the deviations of job completions from the fuzzy due date. Then, the property of an optimal schedule is obtained and a polynomial algorithm that can find the optimal schedule under certain conditions is developed. Numerical results are also reported to show the effectiveness of the algorithm in general cases where these conditions are not satisfied.

A more generalized form of early/tardy scheduling problem is one where each job has a distinct due date. Along this line, a dynamic programming state-space relaxation method was implemented by Abdul-Razaq and Potts (1988) to minimize the sum of earliness and tardiness penalties in a single machine sequencing problem. Unfortunately, this procedure can be successfully used only to solve small-sized problems due to its prohibitive storage requirements. For the same problem, Ow and Morton (1989) suggested a beam search method in which only a limited number of solution paths are explored in parallel. The intent of their approach was to search quickly with no backtracking. Thus the optimal solution is not guaranteed by the method. For distinct due date problems which allow inserted ideal time, Yano and Kim (1991) proposed a heuristic procedure that uses dominance properties to reduce the number of sequences

under consideration. Davis and Kanet (1993) presented a timetabling procedure to minimize the total earliness and tardiness of jobs. This procedure, however, has to be embedded in an enumerative algorithm such as branch and bound to search the entire domain of job permutations. A genetic algorithm is developed by Lee and Choi (1995) to address single machine scheduling with distinct due dates and general penalty weights. An exact algorithm is used to determine the optimal starting time of each job in a given sequence by inserting idle times. Prominent near optimal sequences can then be obtained via genetic algorithm. Lyu *et al.* (1996) implemented simulated annealing (SA) technique to solve the single machine early/tardy scheduling problem. They concluded that SA can provide good solution to this problem in a reasonable time. However, they pointed out that this algorithm is very sensitive to its control parameters and the problem specifications. Taking into account the sequence dependent setup times, Wang and Wang (1997) addressed the single machine earliness/tardiness scheduling problem with different due dates. The objective is to minimize the total cost, which includes the earliness and tardiness penalty and the total setup cost. For this problem, a hybrid algorithm has been developed by combining the heuristic with a genetic algorithm. Their experimental results show that the proposed algorithm is efficient. To minimize the total earliness penalty, Gordon and Strusevich (1999) considered the single machine scheduling problem in which the due dates are obtained by adding a slack time to the job processing times. They assumed that a schedule is feasible if there are no tardy jobs and the job sequence satisfies given precedence constraints. The problem is to determine a slack time and then to find a feasible schedule so as to minimize an earliness penalty function. They have

shown that this problem is NP-hard and can be solved efficiently only for special cases.

Another important category of performance measures in machine scheduling is the one related to the job flow time. Flow time is usually a measure that represents manufacturers concerns since minimizing flow time implies minimizing work in process inventory. Some of the commonly used measures in this category are mean flow time and in process inventory which have relatively straightforward optimal solution algorithms. In recent years, scheduling research involving more sophisticated form of such measures have received some attention from researchers in order to respond to the highly competitive manufacturing environment. These new problems are usually combinatorial in nature and have non-regular performance measures. Merten and Muller (1972) were among the first who introduced the completion time variance (CTV) as a non-regular performance measure. For single machine scheduling, Gupta *et al.* (1993) applied genetic algorithm (GA) to minimize the squared deviation from mean flow time. They found that the control parameters of GA technique greatly affect the solution quality for this type of problems and specific set of these parameters should be selected for successful implementation of GAs. Veral and Mohan (1996) reported a simulation analysis to address the due-date setting problem in single-machine job shops. They considered the non-linearity in job flow times and prescribed a static due-date setting method that allocates flow allowances to the jobs accordingly. Their simulation results have demonstrated that the proposed due date setting approach improves overall system performance and is superior to the traditional due-date assignment methods. Similar measures have also been investigated by other researchers such as Bagchi (1989), Gupta

et al. (1990) and Kellerer *et al.* (1996).

Studies concerning single criterion problems in scheduling literature are wide spread. However, they often fail to give a complete representation of the system under consideration. This is because these types of studies aim to optimize a single performance measure in isolation, usually ignoring other contributing factors and constraints. Consequently, the outcome may not be applicable to real life problems. This has given rise to multicriteria scheduling studies that will be reviewed in the next section.

2.1.3 Multiple criteria scheduling problems

During the last two decades, single machine scheduling with multiple criteria has been studied by a number of researchers. In literature, there are generally three approaches to formulate multicriteria problems: weighting of criteria, secondary criterion and efficient set generation (Tsiushuang *et al.* 1997). In the first approach, weights are assigned to subcriteria to convert them to a single criterion problem. The secondary criterion method attempts to optimize some subcriteria while some other subcriteria are regarded as the problem constraints which must be satisfied. The last approach generates all *efficient* (nondominated) schedules based on the performance measures. The decision maker then has to make explicit trade offs among these schedules and select the best one depending upon the problem in hand.

Although many performance measures have been considered in the multicriteria scheduling literature, many of these studies include some forms of earliness /tardiness and job flow times as performance measures. Along this line, Fry *et al.* (1987) developed

a model that combines total flow time, total earliness and total tardiness to form a single objective function. They formulated this problem as a mixed integer programming model and proposed a branch and bound algorithm to solve it. Lin (1993) studied a single machine scheduling problem to minimize weighted sum of earliness and tardiness subject to the maximum allowed tardiness. Lagrangian relaxation is used to obtain a lower bound for the objective function. A branch and bound algorithm is then formulated to find the optimal solution. Kondakci *et al.* (1996) tackled the problem of minimizing total flow time and maximum tardiness on a single machine. They developed an algorithm to find the optimal solution for any given nondecreasing function of the two criteria by generating a small subset of efficient schedules.

Recently, Koksalan *et al.* (1998) considered the bicriteria scheduling problem of minimizing maximum earliness and total flow time on a single machine for the cases when machine idle time is allowed and when it is not allowed. Heuristic procedures were developed to generate all approximately efficient sequences. These procedures can then be used in conjunction with a branch and bound algorithm to find the best sequence. Tsiushuang *et al.* (1997) addressed the single machine sequencing problem to minimize total weighted earliness subject to maximum tardiness for each job. A heuristic procedure and a branch and bound algorithm have been developed to solve different cases of the problem.

Emmons (1975) proposed a branch and bound algorithm to minimize the total flow time with the minimum number of tardy jobs. His approach incorporates a secondary objective of minimizing the total flow time in addition to the primary objective

of minimizing the number of tardy job for a given deadline. The proposed algorithm is applied to reduce the size of the branch and bound tree. Kyparisis and Douligeris (1993) extended this problem for the case when an optimal selection of jobs is also required. They used a heuristic for general cases of the problem and showed that Emmon's algorithm can be modified to yield an optimal solution for some special cases. Kondakci and Bekiroglu (1997) addressed the simultaneous minimization of these two criteria. They developed several theorems that can be used to curtail the size of branch and bound tree by eliminating the inferior schedules.

Comprehensive multicriteria scheduling surveys have been carried out by Dileepan and Sen (1988), Raghavchari (1988), Fry *et al.* (1989), Hoogeveen (1992), and Chen and Buflin (1993), among others.

2.1.4 Scheduling with tooling consideration

Tool replacement and operation sequencing are two of the most important issues in process planning which have a significant effect on total production cost. However, in most studies, these issues have been treated separately. As indicated by Bard (1988), "Although the single machine scheduling problem has been studied extensively, the added complication of tooling undermines the usefulness of the much of the current results". Having identified the importance of tooling, Bard (1988) suggested a heuristic which sequences jobs with an objective of minimizing the total number of tool switches. The total number of tools required to process all jobs is assumed to be greater than the tool magazine capacity. The problem is formulated as a nonlinear integer program and solved

using a Lagrangian relaxation method. Tang and Denardo (1988) addressed the same problem assuming that tool magazine must be fully loaded at each instant. The objective is to sequence the jobs and tools with a minimum number of tool switches. They proposed a greedy heuristic to sequence jobs. Then, the *Keep Tool Needed Soonest* (KTNS) policy is used to minimize the total number of tool switches for the tool loading problem. The "tool replacement" problem is generally defined as the minimization of the tool change intervals due to tool wear or breakage (e.g. Zhau *et al.* 1990). However, Bard (1988) as well as Tang and Denardo (1988) used the term tool replacement to address the tool switch required by job changes.

Nayanzin (1989) studied the task of choosing optimum flows of parts and tools in flexible manufacturing systems (FMS). The principle of grouping parts into part families and tools into toolsets is employed to formulate the problem. It is then solved by reducing it to a problem of maximum coverage using a U-graph, which is a variant of the arithmetical graph. Bard and Feo (1989) studied the problem of minimizing the total setup, tool replacement and machining times for individual batches subject to tool magazine capacity and material removal constraints. Their approach requires that all feasible tool paths be generated before being considered by the optimization algorithm. Chandra *et al.* (1993) added the due date constraint to job and tool sequencing to minimize the setup and processing times. The problem is formulated as a nonlinear integer programming model. Two solution procedures are proposed: a dynamic programming method to optimally solve small sized problems and a randomization heuristic for large size problems.

Ghosh *et al.* (1992) carried out a simulation study to investigate the impact of alternative tool assignment and job dispatching rules on the job shop operation. Their findings showed that, as the setup time increases and tool availability decreases, the performance of the system is mostly affected by the tool assignment rule used to manage tools in the system. They concluded that an effective tool assignment strategy should be based on job sequence information as well as tool related information. Shewchuk and Chang (1995) studied the single machine scheduling with tooling constraints in which tools are considered to be recyclable resources. The objective is to find the job sequence, tool type quantities, and tool recycling schedule such that the sum of job completion times and quantity of tools allocated are both minimized. However, their algorithms require that either the job schedule or quantity of tools allocated be fixed in advance. For the single tool type problem, Liu *et al.* (1997) employed tool reliability to jointly obtain the preventive tool replacement times and the sequence of operations in an FMS over a finite time horizon. A constructive procedure is used to determine preventive tool replacement and to select the next operation based on the age of the tool and the available operations at each stage. Some noteworthy studies and surveys in the area of machine scheduling with tooling consideration are given in Carrie and Perera (1986), Veeramani *et al.* (1992), Gray *et al.* (1993), Avci and Akturk (1996), and Crama (1997).

One of the shortcomings in most of the above studies is that tool switch due to different operation requirements and tool replacement due to possible in process failure have been treated separately. The studies that aim at minimizing number of tool switches (or setup cost due to tool switches) often overlook the cost associated with tool wear or

in process failure. Similarly, the issue of tool switch scheduling has also been ignored in the research concerning preventive tool replacement.

2.1.5 Tool life and tool replacement policies

Tool changes that are based on the arrival of parts to the machine usually fall into two categories: tool changes due to product variety and tool changes due to tool wear or tool failure. The research related to the former case has been given in previous section. The latter is addressed in literature as the "tool replacement" problem. Generally, a tool may be replaced once it produces unsatisfactory parts or prior to it if its "economic tool life" is first reached.

Tool replacement strategy and tool lives are related. Tool life (or *expected* tool life) must be known in advance in order to decide its replacement intervals. Both deterministic and probabilistic techniques have been used to determine the economic lives of tools. In the former approach, empirical expressions, based on machining parameters, are used to calculate tool life. Taylor (1907), in his pioneering research, developed the classic relationship between average tool life and cutting speed. Later on, Cook (1973) proposed an extended tool life expression to account for the effects of feed rate, depth of cut as well as cutting speed on tool life. In addition, most of the machinery handbooks also used empirical formulas to determine tool life for machining operations. In recent years, it has been recognized that tool life is a random variable and should be described probabilistically. Literature indicates that standard distributions such as normal, Weibull, and exponential distributions as well as their combinations can be justified to describe the

life of a tool under certain machining conditions (e.g. Ramalingam and Watson 1977; Hitomi *et al.* 1979; and Iakovou *et al.* 1996).

Tool replacement strategies usually aim at specifying the tool change schedules to minimize the production cost including machining cost, setup cost and in-process breakage cost. Taking into account tool life economics, Mittal and Lewis (1989) developed a mixed integer programming model to minimize the total machining time, tool change time, and tool travel time. To obtain a minimum required number of good parts, Palei and Zubarev (1990) developed a model to minimize the overall production cost and optimize the number of tool changes due to tool failure, while parts machined with failed tools are considered to be scrapped. Zhou *et al.* (1990) presented a non linear tool replacement model based on tool wear status to minimize machining cost. Maccarini *et al.* (1991), have considered the reliability of tools to evaluate the effect of machining cycle on the final product cost. Recently, Makis (1996) analyzed tool replacement problem with asymmetric quadratic loss function. Although computationally prohibitive for large scale problems, his approach provides optimal initial setting and tool replacement time by solving two sets of nonlinear equations. Jianqiang and Moi Keow (1997) have used lognormal distribution to fit the tool life distribution caused by wear. This tool reliability function was then employed to determine optimal tool replacement intervals which would minimize the machining cost.

Despite its importance in machining economy, there are only a few studies that have considered simultaneous determination of the optimal cutting speed and tool replacement policy. Commare *et al.* (1983) addressed this problem assuming tool

replacement time as an integer multiple of the unit machining time. The optimal cutting speed and tool replacement interval are then obtained by applying a two-dimensional search in the solution domain. Using Taylor's equation, Sheikh *et al.* (1985) proposed an analytical formulation to determine cutting speed and tool replacement policy under various conditions. Koulamas (1991) has used geometric programming to address a similar problem. Lately, Iakovou *et al.* (1996) proposed solution procedures for the single-operation-single-tool problem with stochastically distributed tool lives. They have shown that if tool life has a Gamma distribution, the objective function is separable and the problem can be solved easily. Most studies on tool replacement and machining conditions optimization are limited to single-tool-type and single-part-type problems. Billatos and Kendall (1991) are probably among the first who investigated the tool replacement problem for multi-tool-type and multi-part-type systems. They presented a general optimization model to minimize the production cost. Although premature tool failure cost is taken into account, a linear random tool wear function is used instead of the actual nonlinear one. In addition, since their focus was on the optimization of machining condition, the job sequence was assumed to be known and fixed in advance.

In reality, however, a tool replacement decision is strongly affected by and, in turn, directly influences the part sequencing decision because of their common objective and constraints. This, therefore, calls for joint investigation of part sequencing and tool replacement problems.

2.1.6 Scheduling with controllable processing times

Most scheduling related research has treated job processing time as a fixed and given parameter. In many cases, however, the processing times can be controlled within a certain range. The concept of controllable processing time is justified in situations where jobs can be completed in shorter or longer durations by increasing or decreasing additional resources. Yet, the research corresponding to this class of problems is relatively sparse and, to the best of our knowledge, there are only a handful of papers dealing with the controllable job processing times.

Vickson (1980) was among the first who pioneered the research on single machine scheduling with compressible processing times. He has shown that, to minimize the average flow cost, the problem can be formulated as an assignment problem that is easily solvable. According to his results, to minimize the average flow cost, each job is either processed normally or maximally compressed in the optimal solution. In the same paper, an exact algorithm has been developed to minimize the maximum tardiness for the case where job processing times can be compressed to a certain limit. However, the proposed algorithm for the maximum tardiness case provides the optimum processing times only for a given sequence and does not deal with the part sequencing problem. Daniels and Sarin (1989) analyzed the problem of joint sequencing and resource allocation to minimize the number of tardy jobs. They provided theoretical results that help to develop the tradeoff curve between the number of tardy jobs and the amount of resource allocated to reduce job processing times. Lee (1991) presented a parametric study of minimizing total job processing cost as well as the average flow cost.

Panwalker and Rajagopalan (1992) proposed a constructive algorithm to solve single machine sequencing problem and common due date assignment with compressible processing times. The objective function is the sum of penalties based on earliness, tardiness, and processing time compressions. The proposed algorithm determines optimum processing times for all jobs, an optimum sequence, and the smallest value of a common due date. Alidaee and Ahmadian (1993) extended the results of Panwalker and Rajagopalan (1992) to the parallel machine scheduling case. More recently, Cheng *et al.* (1996) considered the due date assignment and single machine scheduling problem in which the job processing times are controllable variables. The objective was to determine optimal due dates, optimal job sequence, and optimal processing time compression to minimize a penalty function including earliness, tardiness, due date, and processing time compression costs. Two models were developed and the optimality conditions for both models were established. These models can be formulated as assignment problems which are solvable in polynomial times.

Zhang *et al.* (1996) addressed the bicriteria scheduling problem of minimizing the cost of weighted flow times and the cost of processing time compressions. They developed an exact algorithm for the special case of equal processing times and equal compression costs. They also proposed a branch and bound approach to solve the general case of this problem. Cheng *et al.* (1998) analyzed the single machine scheduling problem with controllable processing times to minimize the sum of job compression cost and the cost associated with the weighted number of late jobs. The problem was shown to be NP-hard even when all jobs have a common due dates and identical lateness cost.

They have developed a dynamic programming algorithm as well as some heuristic procedures for solving this problem. A comprehensive survey and a good summary of known research on this subject can be found in Nowicki and Zdrzalka (1990) and the references therein.

Although in recent years machine scheduling with compressible processing time has been an active field of research, the sequencing issue of this problem for more realistic cases where the processing times can be either reduced or increased within certain range has not been addressed in literature. Furthermore, the effect of sequence dependent setup times has often been ignored and in most of the studies due dates are considered to be common for all jobs. These limitations make the results difficult to use for real life sequencing decisions.

2.1.7 Operation sequencing for hole making

Hole making is a very common operation in metal cutting industry. This operation accounts for more machining time and cost than any other operations. Thus reducing the cost of hole making is an important consideration in reducing the total production cost.

Hole making operations include drilling, reaming, and tapping. The main difference between hole making and other machining operations is the tool travel time. One survey showed that tool and part movements take on average 70% of the total time in a manufacturing process (Merchant 1985). The tool travel time, or more precisely non-cutting tool travel, in hole making operations could make up a large portion of total operational time. Hence, the proper determination of the operation sequence and the

corresponding machining speed used to perform each operation are crucial in reducing the total production cost.

While much work has been carried out to determine machining parameters for a single operation, the issue of operation sequencing has not been jointly addressed in literature. The studies on similar problems in punching operations are also scarce. In this direction, Walas and Askin (1984) and Chauney *et al.* (1987) proposed heuristic algorithms based on travelling salesman problem to minimize total tool travel distance. Using an artificial intelligence approach, Ssemakula and Rangachar (1989) proposed a method to generate an operation sequence applicable to a variety of manufacturing processes. Roychoudhury and Muth (1995) examined several heuristic techniques for NC punch press operation sequencing.

Luong and Spedding (1995) are among the few who addressed process planning in hole making operations. They developed a generic knowledge based system for process planning and cost estimation in hole making. Based on input data, the proposed system can recommend the appropriate tools, their sequence, and their respective machining conditions for each individual hole. The manufacturing cost is then calculated based on the recommended process plan. A similar approach was taken by Khoshnevis and Tan (1995) to develop a rule-base module for hole making. Their system can be used to provide a process plan chosen among all possible operation sequences for a given feature. Taiber (1996) presented a search procedure to minimize the number of tool changes, non-cutting tool path, machining time, and tool cost for prismatic workpieces. At each iteration, a good solution is found for each of these criteria, using different algorithms.

The threshold accepting heuristic method is then used to navigate the search through the solution space. Taiber's computational experiments showed a 15% cost reduction over 25 minutes of search time for a workpiece consisting of 48 manufacturing features and 31 tool types. Usher *et al.* (1997) presented an object-oriented approach to generate and rank alternative tool sets for the part process plan. The system is designed as a part of a dynamic process planning system and is able to extract the required data from the product model of the CAD file.

Nevertheless, there are two main drawbacks in existing literature on this type of optimization problem. First, in punching operation the main concern is to minimize those costs related to non-cutting tool travel only. It was also assumed that for each operation the cutting speed was fixed and the problems of tool selection and tool assignment did not arise (Roychoudhury and Muth 1995). Second, every feature (hole) is treated separately and global optimization of the overall process plan is ignored. Therefore, it is necessary to develop a solution procedure to simultaneously minimize the costs of machining, tools, tool travel, and tool switch for hole making.

2.2 Tabu search

2.2.1 Tabu search methodology

This section intends to give a brief introduction to the tabu search and its main elements in a general manner. The modifications that may be made to apply this technique to the problems stated in this research will be elaborated in the relevant sections.

In a broad sense, tabu search is a high-level iterative procedure that provides a

framework for a neighbourhood search to escape from local optima. It involves the exploration of a problem's solution space through the iterative investigation of solution neighbourhoods. The search process starts from a feasible solution and moves stepwise to a neighbouring solution so that after a number of moves an optimal or near-optimal solution is obtained. To make a move, a set of neighbouring solutions around the current solution is generated and evaluated according to the problem specifications and its constraints. Then, a move is made to the best allowable solution in the neighbourhood. However, unlike other descent search techniques, such a move may not necessarily improve the objective function.

Another important feature of tabu search is the use of tabu list to keep track of solutions selected in the past iterations. A tabu list contains a pre-defined number of previous moves which are not allowed at the current iteration. This can be thought of as a short term memory and is used to avoid returning to the portion of solution space that has already been searched. In addition to the tabu list, many tabu search applications make use of diversification strategies (long term memory) to re-route the search to new regions. These features are designed to prevent cycling and to expand the search area. The basic components of tabu search are explained below.

a) Starting point: The search has to commence from an initial feasible solution. This could be any feasible solution, say s , that satisfies specifications and constraints of the problem.

b) Neighbourhood: For a given solution, s , the neighbourhood $N(s)$ is a set of feasible solutions which is directly generated by performing one transition in the current solution,

s , within the feasible range.

In the scheduling literature, *pairwise interchange* is probably the most widely used operator to make such a transition. In this method, a solution is obtained by switching the jobs in positions i and j . The complete pairwise interchanges of a J -job problem leads to $|N(s)| = J(J-1)/2$ neighbours. The *extraction and reinsertion* is another operator with which the neighbourhood of s containing all solutions is obtained by extracting the job in position i and inserting it right after (or before) the job in position j . The neighbourhood size for the extraction and reinsertion approach is almost twice as large as the one obtained with pairwise interchange and thus this mechanism appears to be more computationally demanding. Furthermore, as indicated in literature (e.g. Adenso-Diaz, 1992; and Chen *et al.*, 1998), none of these two neighbours seems to outperform the other in terms of solutions quality for a given run time. Therefore, in this research only the pairwise interchange is used as the basic neighbourhood generation mechanism.

c) *Move*: A move is the transition from the best solution, s^{**} , in the previous neighbourhood to the best permissible solution, s^* , that has the lowest $G(s)$ value in the current neighbourhood. The stepwise transition from one solution to another allows the search to reach an optimal or close-to-optimal solution after a number of moves. However, a single move, by itself, may not necessarily improve the current value of objective function. This distinguishes tabu search from other traditional techniques such as hill climbing that require each move to be an improving step. Throughout the search, the best solution found so far, $Cbest$, and its corresponding sequence, $Sbest$, will be recorded and updated.

d) Tabu list: One of the important features of tabu search is its ability to avoid being trapped in local optima by constructing a list of tabu moves. A tabu list, T_list , includes a certain number of previous moves which are not allowed at the current iteration. Once a move from s'' to s^* is made, s'' is added to the top of tabu list and the oldest member of T_list is removed. Thus, returning back to this s'' is forbidden for the next T_size iterations. This can exclude, to some extent, those moves which lead to possible cycling.

The size of the tabu list can affect the search performance. Although a longer list may prevent cycling, it requires more computer scanning and may limit the search domain. The best tabu list size appears to be problem dependent and there is no fixed rule to follow in determining tabu list size so far.

e) Termination criteria: The last element necessary for tabu search is a termination criterion. In general, the search can be stopped after a certain number of iterations, $Mmax$, is completed, after a pre-defined of computational time, $Tmax$, is reached, or when no improvement can be obtained in a specific number of moves.

The details of this technique are well documented in Glover (1989, 1990a, 1990b) and Hertz and de Werra (1991).

2.2.2 Applications of tabu search in single machine scheduling

Tabu search has proven to be an effective approach to a wide spectrum of engineering problems. Nowhere has this success been more remarked than in production scheduling. A complete review of published applications of this technique in production scheduling can be found in Barnes and Laguna (1991).

Brandimarte (1993) utilized tabu search to jointly solve routine and scheduling problem in a flexible job shop in which the assignment of operations to machines is not priori fixed. The objective is to minimize the makespan and the total weighted tardiness. Punnen and Aneja (1993) applied tabu search to two cases of job assignment problem in a multi-machine flow shop. The objective was to minimize total processing times over all possible categories. They also studied the effects of tabu list size and different diversification strategies on solution quality. Comprehensive computational tests were carried out by Sinclair (1993) to study the performance of several heuristics, including simulated annealing, genetic algorithm, and tabu search in solving combinatorial optimization problems. Using real data, Sinclair compared the performance of these methods in solving a typical case of the quadratic assignment problem. The computational results showed that tabu search is superior as far as solution quality is concerned. Srivastava and Chen (1993) proposed several versions of tabu search to tackle the part type selection in an FMS. In their study, tabu search outperformed simulated annealing in both solution quality and computational efficiency. Barnes and Chambers (1995) applied tabu search to minimize the makespan in the job shop scheduling problem. It is assumed that a set of distinct operations is performed by a number of machines and no machine can complete more than one operation at a time. The search starts from the best solution given by different dispatching rules. It then progresses to the next feasible solution by reversing the order of two adjacent critical path operations performed by the same machine. The procedure is terminated after a specific number of moves has been made without improving the objective function. The search then continues by removing

all tabu restrictions and selecting the best solution found as the starting point.

More recently, Crauwels *et. al* (1996) compared the performance of four local search methods, namely multi-start descent, simulated annealing, tabu search, and genetic algorithm, in solving single machine scheduling problem. The problem involves scheduling a number of jobs that are already partitioned into families on a single machine so that the number of late jobs is minimized. Although the setup time is taken into account, it is sequence-independent and setup is needed only when the operation switches from one family to another. To reduce the time needed to evaluate the entire neighbourhood at each iteration, the proposed tabu search is truncated by accepting the first non-tabu move which improves the objective function value. Nowicki and Zdrzalka (1996) presented a tabu search algorithm to solve a single machine scheduling problem with major and minor setup times. They have considered a case in which jobs are grouped into families that require major setup times and where jobs within families need minor setups. The decision involves dividing the families into batches, sequencing different jobs in the same batch, and sequencing different batches to minimize the cost. This problem was solved for two performance measures: the maximum weighted tardiness and the total weighted tardiness. The computational results showed that this problem could be solved efficiently using tabu search technique. A tabu search algorithm was adopted by Hao *et al.* (1996) to assign a common due date to all jobs and to sequence a set of jobs on a single machine. The objective was to determine an optimal job sequence and the associated optimal due date which minimized the sum of weighted earliness and tardiness penalties. Their computational results revealed that tabu search

outperformed the existing optimal and heuristic procedures. Yagiura and Ibaraki (1996) investigated the performance of four local search heuristics in solving the single machine scheduling problem. The search methods they compared included multi-start local search (MLS), genetic algorithm (GA), simulated annealing (SA) and tabu search (TS). Their results have shown that the definition of neighbourhood is very important for all of MLS, SA and TS. Their findings also indicated that TS could be more effective than other search techniques with proper parameter settings. Recently, Liaw (1999) employed tabu search to find minimum makespan in a nonpreemptive open shop floor. For this problem, the neighbourhood structure is defined using blocks of operations on a critical path. This neighbourhood is then evaluated by an efficient selective procedure. By running his algorithm against 60 benchmark problems, Liaw confirmed that the algorithm could find extremely good solutions for all of the test problems in a reasonable amount of computation time.

Though tabu search has proven to be an effective tool for solving many combinatorial optimization problems, the computational time needed for some applications may still be too long. Particularly, the size of neighbourhood and move selection strategy seem to have a great impact on search efficiency. As a result, there is a growing interest in improving tabu search efficiency and determining the best values for its parameters. In the next section, some notable studies that contributed to improving tabu search technique for solving scheduling problems will be reviewed.

2.2.3 Refinement of tabu search method

The enhancement of the original tabu search method has drawn considerable attention in recent years. Efforts have been made to improve the search performance by finding the best value of its parameters. Among these, special attention has been paid to limit the size of neighbourhood at each iteration. This is due to the fact that, the time taken for each iteration, and hence the computation time, heavily depend on the number of candidate moves and the size of the neighbourhood to be evaluated.

To this end, Andeso-Diaz (1992) introduced the concept of restricted neighbourhood to reduce the computational time in solving the weighted tardiness flowshop problem. His approach was based on the observation that, if the objective function contains tardiness (or earliness) penalty, larger transposition range tends to be used in the first few iterations and the transposition range becomes smaller as the search progresses. Thus, Andeso-Diaz proposed that the entire neighbourhood be evaluated in the first few iterations and, within a number of iterations, the transposition range be consecutively reduced to a pre-specified number of neighbouring jobs. Hubscher and Glover (1994) applied tabu search to minimize the makespan in a parallel machine flow shop environment. To increase the search efficiency, a candidate list containing the most promissable moves was used to limit the number of candidate moves at each iteration. Their findings showed a considerable improvement in terms of computational time by applying this strategy. Chen *et al.* (1998) applied tabu search to minimize total flow time in a flexible flow line. They also conducted a factorial experimentation to evaluate the effects of four search parameters (initial solution, tabu list size, neighbourhood size and

type of move) on its performance. Their findings indicated that the search process was insensitive to the size of tabu list and other factor combinations. Hence, the solution quality may not be improved by manipulating the parameters. However, in terms of computational speed, the best run time was obtained when pairwise interchange was used as the neighbourhood generation mechanism and the entire neighbourhood was evaluated.

Another important parameter in tabu search is tabu list size which may affect the search performance in both solution quality and the computation time. Several studies have been conducted to investigate the effect of tabu list size. Some earlier work (e.g. Glover 1986) reported that the best tabu list size was approximately seven. However, it is now recognized that tabu list size should be selected according to such factors as neighbourhood size, type of move, and problem specifications. For the N-city symmetric travelling salesman problem (TSP), Knox and Glover (1989) examined tabu list sizes ranging from $N/5$ to $3N$. They found that the higher end of this range had slightly overall advantage. Consequently a $3N$ tabu list size was selected for their experiment. For the same problem, Malek *et al.* (1988) reported that list sizes ranging from seven to N (number of cities) gave good results. The above studies, however, identify the best tabu list size only for a fixed number of iterations. However, since the time needed to check the tabu status of a candidate move depends not only on iteration count but also on the size of the tabu list, the computational time rather than the number of iterations should be used to examine the effect of tabu list size. Taking this into consideration, Tsubakitani and Evans (1998) studied the problem of optimizing tabu list size for the symmetric TSP within a given computational time limit. Their finding revealed that the best tabu list

sizes, based upon computational time, were generally smaller than those previously reported. They also stated that, for a given problem size, a larger tabu list size was required when the size of the neighbourhood was small. Some notable literature on this topic also includes Taillard (1990), Reeves (1993), William (1994), Della Croce (1995), and James and Buchanan (1997).

Nevertheless, as reviewed in this chapter, two main drawbacks exist in most of the previous work in literature. First, several individual operational decisions are interlocked and concurrently affect the optimality/feasibility of the final results. Unfortunately, in many studies, these decision problems have often been treated individually, thereby undermining the practical usefulness of the research outcome. Secondly, due to the tedious modelling process and computational burden in searching the optimal solution, any real application of the methods is virtually precluded on today's shop floor. The need for integrated approaches and efficient solution methods addressing the machine-level planning problem in advanced manufacturing systems is the main motivation of this study.

Chapter 3

OPERATION SEQUENCING WITH TOOL REPLACEMENT CONSIDERATION

3.1 Background

Automated machining centres have been widely used over the years. The popularity of machining centres is mostly due to their high flexibility and efficiency in processing a range of operations of various parts. An automated machining centre, however, represents heavy capital investment, which can be justified only when the machining centre is effectively operated. Operational decisions such as part sequencing and tool replacement have significant impact on operation effectiveness.

Although a rich body of literature is available in dealing with job sequencing and tool provisioning in various contexts, joint consideration of these problems has not been addressed adequately. The studies conducted in the area of job sequencing are mostly based on the assumption that the setup costs (or times) are either negligible or are included in their processing times. In addition, the matter of tool reliability and its associated costs are ignored and it is usually assumed that tools are available and completely reliable for processing all assigned jobs. Random tool failures, especially in-process failures, are not taken into account. Some important issues such as defective part cost, the effect of tool spare on total expected production cost, and the feasibility of the tool provisioning decision made in machine loading stage can be investigated only when tool replacement decision is accommodated. These undermine the usefulness of much of the current results due to the lack of proper consideration of the effect of tool

provisioning and setup times on the system performance (e.g. Bard 1988, Carrie and Perera 1987). By the same token, tool provisioning policies have been studied under the assumption that job sequence is fixed and known throughout the production period.

In practice, however, job sequencing decision is strongly affected by and, in turn, directly influences the tool provision decision due to their mutual effects. For instance, whether or not a tool is replaced for the next operation will depend on which job is scheduled for the next operation. If the expected remaining tool life is greater than the processing time of the next job, it may be desirable not to replace the tool. Otherwise, the tool should be replaced. Consequently, job sequencing and tool replacement problems should be treated as interrelated decisions in the production planning.

Single machine sequencing with sequence-dependent setup times has been recognized as an NP-complete problem (Rinnooy Kan 1976) which cannot be optimally solved in polynomial time. In the stated problem, the cost components, except the setup cost, are not deterministic in nature. Their values are not only sequence-dependent but also position-dependent. That is, the expected cost for processing a job is affected by both its immediate predecessor job and historical decisions. This further complicates the problem and precludes optimal solutions for any meaningful size problems. To provide quick and good solutions for daily shop floor decisions, a tabu search approach will be used to jointly solve job sequencing and tool replacement problems on an automated machining centre. The approach proposed in this research is different from those in current literature as it simultaneously provides the sequence of jobs and tool replacement intervals. The analysis of the problem is given in the following section.

3.2 Problem statement

Consider an automated machining centre with an automated tool changer and a tool magazine of limited capacity. A number of operations of various jobs can be processed on the machining centre provided that the required tools are available in the tool magazine. Different jobs may have similar or quite different fixturing requirements. Each job has a fixed operation sequence defined by a set of required tools. Once a job is setup, unless a tool fails during a machining operation, it will stay on the machine until all of its operations are completed. Each tool may have different tool life distributions when processing different jobs. The problem under investigation involves multiple tasks: i) to determine the tool replacement interval and the number of required tool spares; and ii) to find the best job processing sequence so that, once the tool replacement decisions have been properly made, the total *expected* production cost is minimized. Here, the total expected production cost includes the setup cost, defective cost, tool cost, and machining cost which are explained as follows.

Setup cost : In an automated machining centre, the setup times are required by the following activities. (a) load required tools, (b) change tools, (c) change part holders, (d) chuck up parts, (e) load NC programs, and possibly (f) perform "dry" runs. Since the required tools are loaded onto the tool magazine at the beginning of the production period and the required time is not affected by job sequence, the setup time for (a) can be excluded from this analysis. Further, as tool changes are normally done between operations, tool change times are usually aggregated together with processing times. The aggregation of tool change times with processing times is justified by the fact that the

sequence of operations on each job is fixed. Activities (c) to (f) are required in most cases when the process shifts from one job to another and are regarded as the main components of the setups. Since different jobs may have similar or quite different fixturing requirements, the job setup times are generally sequence-dependent. Thus the sequencing decision has to be properly made to minimize setup time or cost.

Defective cost : In a machining process, tools may fail randomly. The failure distributions of a tool may vary when different jobs are processed. If a tool fails while processing a part, it is assumed that the part becomes defective. Therefore, the total defective cost depends on the tool replacement decisions. The defective cost is the sum of raw material cost and cumulative machining and tool costs added to the part up to the point when the in-process failure occurs.

Tool cost : Normally, there is a limited number of tool spares available for each tool type on the tool magazine. A tool in use may or may not be replaced, depending on the trade-off between the tool cost and the possible defective cost. The cost of a tool is added to the total cost whenever it is replaced prior to an operation. It should be noted that the cost of the replaced tool due to the in-process failure is not considered as the tool cost since it is already included in the defective cost.

Machining cost : This cost occurs whenever a machining operation is performed. The required machining time (and therefore the machining cost) for each operation is usually known in advance. This is because the speed and feed rate are well controlled in an automated machining centre. As a result, the machining time and cost can be easily documented.

3.3 Problem formulation and analysis

The problem stated above is formulated according to three important performance measures or planning goals, i.e., a) minimization of production cost; b) minimization of production and flow time costs; and c) minimization of production cost with just-in-time consideration. The associated models are presented in the following three subsections.

3.3.1 Planning for minimization of expected production cost

Mathematically, the combined job sequencing and tool replacement problem with the goal of minimizing the total expected processing cost can be formulated as follows:

Model 3-1

$$\text{Min}_s \{ G(s) \} = \text{Min}_s \left\{ \sum_{l=1}^J \left[\sum_{i \in I_{\mu(s,l)}} \min (Zr_{i\mu(s,l)}, ZR_{i\mu(s,l)}) + c t_{\mu(s,l-1)\mu(s,l)} \right] \right\} \quad (3-1)$$

subject to:

$$\sum_{l=1}^J m_{i\mu(s,l)} \leq M_i \quad \forall s, i \in I_{\mu(s,l)} \quad (3-2)$$

Where

$$m_{i\mu(s,l)} = \begin{cases} 1 & \text{if } ZR_{i\mu(s,l)} < Zr_{i\mu(s,l)} \\ 0 & \text{Otherwise} \end{cases} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-3)$$

$$Zr_{i\mu(s,l)} = (Q_i + \Pi_{i\mu(s,l)}) (1 - r_{i\mu(s,l)}) + (C \Upsilon_{i\mu(s,l)} r_{i\mu(s,l)}) \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-4)$$

$$ZR_{i\mu(s,l)} = (Q_i + \Pi_{i\mu(s,l)}) (1 - R_{i\mu(s,l)}) + (C \Upsilon_{i\mu(s,l)} R_{i\mu(s,l)}) + Q_i \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-5)$$

$$\Pi_{i\mu(s,l)} = \pi_{\mu(s,l)} + \sum_{n=1}^{O(i,\mu(s,l))} C \Upsilon_{n\mu(s,l)} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-6)$$

$$R_{i\mu(s,l)} = \exp \left(- \int_0^{\Upsilon_{i\mu(s,l)}} h_{i\mu(s,l)}(t) dt \right) \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-7)$$

$$r_{i\mu(s,l)} = \left(\prod_{q \in \tau_i(l)} R_{iq} \right) R_{i\mu(s,l)} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-8)$$

In this case, a job j requires a set of tools, I_j , and each tool i performs one operation of the job according to the given sequence. The processing time, Υ_{ij} , for each (i,j) combination is known. Each tool may or may not be replaced at the beginning of an operation. The probabilities of having a failed tool and a scrap job at each operation may vary depending on whether the tool is replaced at the beginning of the operation, and are given by $(1-R_{ij})$ and $(1-r_{ij})$ respectively. In the former case, the expected operation cost includes machining cost, the cost of a new tool plus a cost due to the possible in process tool failure. In the latter case, although there is no tool replacement cost, the cost due to the tool failure during the operation could be relatively high because the operation is performed by a used tool. It is noted that although a replaced tool may also fail (for the second time), the probability of such an event (and its consequent cost) is very low and therefore it is excluded from consideration.

The objective function consists of two types of costs: setup cost $ct_{\mu(s,l-1),\mu(s,l)}$ which is deterministic and sequence-dependent, and tool replacement cost $Zr_{i\mu(s,l)}$ (or $ZR_{i\mu(s,l)}$) which is probabilistic and dependent on sequence as well as historical decisions. If a used

tool i is replaced with a new spare tool immediately before processing job $\mu(s,l)$, the sum of the expected tool cost, defective job cost and machining cost is $Zr_{i\mu(s,l)}$ as shown in Equation (3-4); otherwise the sum is $ZR_{i\mu(s,l)}$ which is expressed by Equation (3-5). The value of $Zr_{i\mu(s,l)}$ is directly affected by the event of in-process tool failure. If a tool fails, the job is scrapped with a cost of $\Pi_{i\mu(s,l)}$ which includes the raw material cost and all dollar value added to the job so far, and a new tool has to be mounted with a cost of Q_i . The probability of an in-process failure is $1-r_{i\mu(s,l)}$. If no in-process tool failure occurs, the only cost is the machining cost with a probability of $r_{i\mu(s,l)}$. As shown in Equation (3-5), if a used tool i is replaced immediately before processing job $\mu(s,l)$, the chance of having an in-process tool failure is reduced to $1-R_{i\mu(s,l)}$ and, accordingly, the expected tool and defective job costs are lower. A term Q_i is also added to $ZR_{i\mu(s,l)}$ to reflect the new tool cost due to the planned tool replacement. In Equations (3-4) and (3-5), the defective job cost, tool reliabilities $R_{i\mu(s,l)}$ and $r_{i\mu(s,l)}$ for each tool-job combination are computed or updated, using Equations (3-6), (3-7) and (3-8), respectively. Solving the model involves two interactive decisions: tool replacement and job sequencing. The tool replacement decision is made in the inner layer of the model by choosing either $Zr_{i\mu(s,l)}$ or $ZR_{i\mu(s,l)}$, whichever is smaller, subject to tool availability constraint (3-2). In the outer layer of the model is the sequencing problem with sequence-dependent setup times.

3.3.2 Planning for minimization of expected production cost and JIT sequencing

Just-In-Time scheduling is one of the important goals in modern production planning. The details of this concept will be elaborated in the next chapter. In brief, in many cases

the cost of a schedule is directly related to the amount by which a job's due date has been missed. Accordingly, to avoid tardiness or earliness penalties it may be desirable to process each job as close to its due date as possible. In view of this, the JIT objective can also be incorporated into our planning decision. This leads to the following model.

Model 3-2

$$\begin{aligned} \text{Min}_s \{ G(s) \} = \\ \text{Min}_s \left\{ \sum_{l=1}^J \left[\sum_{i \in I_{\mu(s,l)}} \min(Zr_{i\mu(s,l)}, ZR_{i\mu(s,l)}) + c t_{\mu(s,l-1)\mu(s,l)} \right] + a_{\mu(s,l)} T_{\mu(s,l)} + b_{\mu(s,l)} E_{\mu(s,l)} \right\} \end{aligned} \quad (3-9)$$

subject to:

$$\sum_{l=1}^J m_{i\mu(s,l)} \leq M_i \quad \forall s, i \in I_{\mu(s,l)} \quad (3-10)$$

$$\sum_{k=1}^l \left(\sum_{i \in I_{\mu(s,l)}} \Upsilon_{i\mu(s,l)} + t_{\mu(s,k-1)\mu(s,k)} \right) - d_{\mu(s,l)} \leq T_{\mu(s,l)} \quad \forall s, l \quad (3-11)$$

$$d_{\mu(s,l)} - \sum_{k=1}^l \left(\sum_{i \in I_{\mu(s,l)}} \Upsilon_{i\mu(s,l)} + t_{\mu(s,k-1)\mu(s,k)} \right) \leq E_{\mu(s,l)} \quad \forall s, l \quad (3-12)$$

$$E_{\mu(s,l)} \geq 0 \quad \forall s, l \quad (3-13)$$

$$T_{\mu(s,l)} \geq 0 \quad \forall s, l \quad (3-14)$$

Where

$$m_{i\mu(s,l)} = \begin{cases} 1 & \text{if } ZR_{i\mu(s,l)} < Zr_{i\mu(s,l)} \\ 0 & \text{Otherwise} \end{cases} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-15)$$

In the above model, the last two terms added to the objective function represent the cost associated with the deviation from the jobs due dates. Constraints (3-11) and (3-12) respectively determine the amount of tardiness or earliness of each job in the sequence. These two constraints along with constraints (3-13) and (3-14) also ensure that no job can be early and tardy at the same time. The remaining cost components and their definitions are the same as those explained in the previous section.

3.3.3 Planning for minimization of expected production cost and flow time cost

In Section 3.3.1 we developed a probabilistic mathematical model to minimize total expected cost including tool cost, setup cost, defective job cost and machining cost. The model aimed to provide an optimal job sequence and the associated tool replacement intervals. The objective function of that model however did not include any cost component related to flow time of the jobs.

The minimization of flow time is important not only because it will lead to lower work-in-process inventory cost but also because flow time, in a sense, represents a kind of service quality. It is therefore important to incorporate such a concern into the planning model. It is noted that job flow times are considered to be probabilistic in nature. The new model is given below.

Model 3-3

$$\begin{aligned} \text{Min}_s \{ G(s) \} = \\ \text{Min}_s \left\{ \sum_{l=1}^J \left[\sum_{i \in I_{\mu(s,l)}} g_1 + c t_{\mu(s,l-1)\mu(s,l)} + u_{\mu(s,l)} \left(R T_{\mu(s,l)} + \sum_{i \in I_{\mu(s,l)}} g_2 + t_{\mu(s,l-1)\mu(s,l)} \right) \right] \right\} \end{aligned} \quad (3-16)$$

subject to:

$$\sum_{l=1}^J m_{i\mu(s,l)} \leq M_i \quad \forall s, i \in I_{\mu(s,l)} \quad (3-17)$$

Where

$$g_1 = \max \left(Z r_{i\mu(s,l)} \cdot m'_{i\mu(s,l)}, Z R_{i\mu(s,l)} \cdot m_{i\mu(s,l)} \right) \quad (3-18)$$

$$g_2 = \max \left(R r_{i\mu(s,l)} \cdot m'_{i\mu(s,l)}, R R_{i\mu(s,l)} \cdot m_{i\mu(s,l)} \right)$$

$$m_{i\mu(s,l)} = \begin{cases} 1, & \text{if } Z R_{i\mu(s,l)} + u_{\mu(s,l)} \cdot R R_{i\mu(s,l)} < Z r_{i\mu(s,l)} + u_{\mu(s,l)} \cdot R r_{i\mu(s,l)} \\ 0, & \text{Otherwise} \end{cases} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-19)$$

$$m'_{i\mu(s,l)} = 1 - m_{i\mu(s,l)} \quad (3-20)$$

$$Z r_{i\mu(s,l)} = (Q_i + \Pi_{i\mu(s,l)}) (1 - r_{i\mu(s,l)}) + (C \Upsilon_{i\mu(s,l)} r_{i\mu(s,l)}) \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-21)$$

$$Z R_{i\mu(s,l)} = (Q_i + \Pi_{i\mu(s,l)}) (1 - R_{i\mu(s,l)}) + (C \Upsilon_{i\mu(s,l)} R_{i\mu(s,l)}) + Q_i \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-22)$$

$$\Pi_{i\mu(s,l)} = \pi_{\mu(s,l)} + \sum_{n=1}^{O(i,\mu(s,l))} C \Upsilon_{n\mu(s,l)} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-23)$$

$$R r_{i\mu(s,l)} = \Upsilon_{i\mu(s,l)} + (1 - r_{i\mu(s,l)}) \Upsilon_{i\mu(s,l)} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-24)$$

$$R R_{i\mu(s,l)} = \Upsilon_{i\mu(s,l)} + (1 - R_{i\mu(s,l)}) \Upsilon_{i\mu(s,l)} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-25)$$

$$RT_{\mu(s,l)} = RT_{\mu(s,l-1)} + \sum_{i \in I_{\mu(s,l-1)}} \max(Rr_{i\mu(s,l-1)}m', RR_{i\mu(s,l-1)}m) + t_{\mu(s,l-1)\mu(s,l)} \quad \forall s, l \quad (3-26)$$

$$R_{i\mu(s,l)} = \exp \left(- \int_0^{\tau_{i\mu(s,l)}} h_{i\mu(s,l)}(t) dt \right) \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-27)$$

$$r_{i\mu(s,l)} = \left(\prod_{q \in \tau_i(l)} R_{iq} \right) R_{i\mu(s,l)} \quad \forall s, l, i \in I_{\mu(s,l)} \quad (3-28)$$

In this model, u_j is the unit flow time cost of part j and RT_j the waiting time of part j . The objective function consists of two main cost components: a) the expected processing cost including the setup cost and the cost of expected defective part and failed tool due to in process tool failure (this component represents the main concern of the manufacturer), and b) the cost of expected flow time of the jobs in the system (this cost represents the customer's concern). The former cost component has been explained before. The latter cost is also affected by tool replacement decision and is probabilistic in nature.

Here, the expected flow time of a job through the system includes its expected waiting time, given by Equation (3-26), and its expected processing time, determined by Equations (3-24) and (3-25). The 0-1 decision variable m' is added to the model to make the effect of tool replacement decision consistent in both costs. That is, the aggregate cost of expected job's flow cost together with expected processing cost determine whether a tool should be replaced at the beginning of each operation.

3.4 Solution procedure

In general, the sequencing problem with sequence-dependent processing times or costs on a conventional machine is equivalent to the travelling salesman problem (TSP) (Bellmore and Nemhauser 1968) -- a well known NP-complete problem (Rinnooy Kan 1976). Optimal solutions can be feasibly obtained only for small problems. This forces researchers to seek approximate algorithms. An excellent survey of approximate algorithms for the TSP has been carried out by Golden *et al.* (1980). They classify the heuristics into three categories: construction procedure, improvement procedure (or branch exchange procedure), and composite procedure (a combination of construction and branch exchange procedures). By comparing the performance of some well known heuristics in solving five 100-node problems, they conclude that a three-step composite procedure generally provides more accurate solution. They also point out that the three-step composite procedure is computationally slower. However, the branch exchange heuristic, a critical step in the three-step procedure, can be applied only to the problems in which the "distance" between any two adjacent nodes is fixed. For instance, in the following two "tours",

a-b-c-d-e-f-g and a-b-e-f-c-d-g

the "distance" between *c* and *d* is fixed whether their *positions* are between *b* and *e* or between *f* and *g*. However, when in-process failure cost is taken into account, the incremental cost ("distance") between a pair of adjacent jobs is affected by all previous tool replacement decisions. Therefore, the incremental cost or "distance" between any two adjacent jobs is *position-dependent*, i.e., even if the sequence of a pair of adjacent

jobs is fixed, the incremental cost of adding the second job of the pair to the "tour" may be different if their positions relative to other jobs in the "tour" are changed. This point further complicates the problem and prohibits the application of any branch exchange heuristics. Moreover, the real-world shop floor practice is more in favour of quick and relatively good heuristics. To this end, tabu search approach, a more general technique for solving combinatorial optimization problems, is adopted to solve this problem.

3.4.1 Development of tabu search algorithm

This section is devoted to the development of a tabu search algorithm for solving Model 3-1. It is noted that the proposed algorithm can be modified to solve Model 3-2 and Model 3-3. The implementation of the proposed solution procedure is illustrated below.

A feasible solution in this case is represented by a sequence s defined by a permutation of the J jobs, subject to constraint (3-2). The set of neighbour solutions $N(s)$ of the present sequence s is generated by pairwise exchanging job positions in s . The best neighbour in $N(s)$ is the one that has the lowest $G(s)$ value and satisfies constraint (3-2). At each iteration a permissible move is made from s'' , the best job sequence in the previous neighbourhood, towards s^* , the best job sequence in the current neighbourhood. The search can be terminated when either a pre-specified number of moves, $Mmax$, has been completed or the maximum allowed computational time, $Tmax$, is reached. Since the computational time, rather than the number of moves, is more important in the shop floor planning the maximum allowed time, $Tmax$, will be used as the termination criterion in the proposed algorithm. If it is desirable to use the maximum number of

moves as the termination criterion, the algorithm can be modified by simply replacing $Tmax$ with $Mmax$. The proposed tabu search algorithm is given as follows:

Algorithm 3-1

Step 1 Initialization

- (1) Read input data $c, C, h_{ij}(t), I_j, M_i, \pi_j, Q_i, t_{jk}, T_{ij}$, and specify maximum allowed search time $Tmax$ and tabu list size T_size .
- (2) Set $Cbest = M_big$ (a big number), $M_ctr = 0$, $T_list = \{\phi\}$, and $Sbest = \{\phi\}$
- (3) Find a starting sequence s^{**} and compute the associated production cost $G(s^{**})$, set $G(s^*) = M_big$.

Step 2 Search

- (1) Generate a feasible neighbour sequence s for s^{**} . If s is not in the current tabu list, calculate $G(s)$ and update the best sequence in the neighbourhood: $s^* \leftarrow s$ if $G(s) < G(s^*)$; otherwise, discard s . Repeat this for all feasible neighbours of s^{**} .
- (2) Make a move: set $G(s^{**}) \leftarrow G(s^*)$, $s^{**} \leftarrow s^*$ and update tabu list T_list .
- (3) If an improvement is observed, i.e., if $G(s^{**}) < Cbest$, update the best solution: $Cbest \leftarrow G(s^{**})$, $Sbest \leftarrow s^{**}$.
- (4) If the maximum allowed search time $Tmax$ is over, stop. Otherwise, update the number of moves made so far in the current phase: $N_ctr \leftarrow N_ctr + 1$. If $N_ctr > Nmax$, go to step 3; otherwise, go to the

beginning of this step.

Step 3 Diversification (re-initiate a search phase)

Set $N_ctr=0$, clear tabu list, and diversify the search path by choosing one of the following strategies:

- (1) Restart from s^{**} , the last job sequence.
- (2) Restart from S_{best} , the best job sequence obtained so far.
- (3) Restart from a randomly selected job sequence.

Go to step 2.

Step 4 Modification of search parameters (optional -- for comparison purpose)

One or both of the following options may be used.

- (1) Change tabu list size T_size and go to (2) of step 1.
- (2) Change the length of each search phase N_{max} and go to (2) of step 1.

The diversification strategies in this algorithm are used to enhance the search performance and to further reduce the chance of cycling. As shown in the algorithm, the search process is divided into a number of search phases when a diversification strategy is applied. Each of the search phases contains N_{max} moves. At the end of each phase, the tabu list is cleared and a diversification option, which could be the same or a different one used in the previous phase, is selected for the next phase. If diversification is not preferred, a sufficiently large N_{max} can be used and the entire search process includes only a single phase. The algorithm is coded in C and its application in solving the proposed models will be demonstrated in the next section.

3.5 Illustrative examples and results

The application of tabu search to solve the proposed models will be illustrated by sequencing 30 jobs on a machining centre with 20 types of tools. Although comprehensive data are needed for solving these problems, most of the data can be obtained from the automated manufacturing environment. For instance, the machining and setup times may be found from the process plans of associated jobs; tool and raw material cost data are available in purchasing department; and tool reliability can be obtained, based on shop floor data (Ramalingam *et al.* 1987). Furthermore, the input data can be read from the spreadsheet. Therefore, it is feasible to use the proposed approach as a decision tool for shop floor planning.

In the following examples, tool lives are assumed to follow the Weibull distribution. It is noted that, in the computer code, the Weibull distribution can be replaced by other tool life distributions if appropriate. The scale parameter α and shape parameter β of the Weibull distribution are assumed to be known for every tool-job combination and distributed in the following ranges (Ramalingam *et al.* 1987):

$$\alpha = [74, 1245] \quad \text{and} \quad \beta = [0.531, 1.680]$$

The above data are obtained from a number of machining tests and are used in this study for illustrative purposes only. Clearly, they may not be valid for modern tools which may have higher service lives. Both machining cost and setup cost are assumed to be \$2/minutes for all operations. The required machining time of each tool-job combination is listed in Table 3-1. The setup times are sequence-dependent and shown in Table 3-2. Initial job costs (raw material costs) and tool costs are given in Table 3-3.

Table 3-1 Operation times (min.) for the example problems

Tool type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Job No.																				
1	---	2.0	---	---	---	1.5	---	3.0	---	---	---	1.0	---	---	---	---	---	3.5	1.2	1.8
2	---	---	3.5	2.5	---	---	---	---	3.0	---	---	---	---	3.5	2.5	---	1.2	---	3.0	---
3	2.5	1.3	---	2.5	2.0	---	---	---	---	3.5	3.0	---	1.2	---	---	---	2.5	---	---	1.0
4	3.5	---	2.5	---	---	1.5	2.3	---	---	1.0	---	---	0.6	2.5	---	---	---	---	---	---
5	---	4.0	3.0	---	2.0	---	---	---	5.0	---	2.0	---	---	---	1.0	1.5	---	---	---	---
6	---	---	---	1.8	---	---	4.0	2.0	---	---	---	2.5	3.5	---	---	---	1.0	0.7	---	---
7	---	2.5	---	---	---	1.0	---	2.5	---	---	---	3.0	---	---	---	---	---	3.5	1.0	0.5
8	---	---	2.0	2.2	---	---	---	---	3.0	---	---	---	---	3.5	2.5	---	1.0	---	3.0	---
9	2.0	6.0	---	1.0	1.5	---	---	---	---	3.0	3.0	---	1.0	---	---	---	2.0	---	---	2.0
10	3.2	---	2.8	---	---	1.0	2.5	---	---	1.5	---	---	2.0	2.2	---	---	---	---	---	---
11	---	3.5	2.0	---	2.5	---	---	---	1.0	---	2.5	---	---	---	4.0	1.5	---	---	---	---
12	---	---	---	4.2	---	---	3.5	2.5	---	---	---	2.3	1.3	---	---	---	2.5	0.5	---	---
13	---	2.0	---	---	---	1.5	---	2.5	---	---	---	0.8	---	---	---	---	---	3.4	1.0	2.0
14	---	---	3.6	2.4	---	---	---	---	3.0	---	---	---	---	1.2	2.4	---	1.2	---	3.0	---
15	1.4	0.5	---	2.3	1.8	---	---	---	---	3.5	3.0	---	1.5	---	---	---	2.5	---	---	1.5
16	1.1	---	2.5	---	---	1.4	2.2	---	---	1.8	---	---	1.0	2.0	---	---	---	---	---	---
17	---	4.0	2.5	---	2.5	---	---	---	1.2	---	2.5	---	---	---	0.5	1.8	---	---	---	---
18	---	---	---	3.0	---	---	3.5	2.5	---	---	---	2.0	1.5	---	---	---	2.5	3.0	---	---
19	---	1.4	---	---	---	1.0	---	2.5	---	---	---	3.0	---	---	---	---	---	0.6	1.2	1.8
20	---	---	1.7	2.5	---	---	---	---	3.0	---	---	---	---	3.5	1.4	---	3.5	---	3.0	---
21	2.4	5.5	---	2.1	1.6	---	---	---	---	1.0	0.7	---	3.8	---	---	---	2.4	---	---	1.0
22	3.5	---	1.4	---	---	1.0	2.5	---	---	2.0	---	---	3.0	2.6	---	---	---	---	---	---
23	---	1.0	1.0	---	2.2	---	---	---	0.5	---	2.5	---	---	---	4.0	1.8	---	---	---	---
24	---	---	---	0.5	---	---	4.0	2.0	---	---	---	2.5	3.8	---	---	---	2.2	3.0	---	---
25	---	2.8	---	---	---	1.0	---	2.2	---	---	---	0.7	---	---	---	---	---	3.5	1.0	2.0
26	---	---	4.0	2.0	---	---	---	---	3.0	---	---	---	---	3.0	2.5	---	1.0	---	2.0	---
27	2.0	1.5	---	2.5	2.0	---	---	---	---	0.5	3.0	---	3.5	---	---	---	2.5	---	---	1.0
28	3.0	---	2.2	---	---	1.5	2.1	---	---	1.7	---	---	5.0	2.0	---	---	---	---	---	---
29	---	4.0	3.0	---	2.0	---	---	---	1.0	---	2.5	---	---	---	4.0	1.5	---	---	---	---
30	---	---	---	4.5	---	---	3.5	2.0	---	---	---	2.5	1.0	---	---	---	2.5	3.0	---	---

"—" means no operation is assigned

Table 3-2 Setup times for the example problems (min.)

Predecessor job	Successor job																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0*	4.0	6.5	1.0	0	4.0	2.0	3.5	3.0	4.0	7.0	3.0	5.0	3.0	1.0	2.0	2.0	6.0	2.5	5.0	1.5	4.0	8.0	1.0	3.5	5.5	8.0	2.0	1.0	7.5	0.5
1	---	2.0	0.5	1.0	2.0	2.5	3.0	1.5	1.0	2.7	0	2.5	2.0	2.5	1.0	1.0	1.5	1.5	2.0	1.5	1.5	0	2.0	2.5	1.5	3.5	1.0	0.5	1.0	7.5
2	2.0	---	1.5	0.5	2.0	2.0	1.0	2.7	1.0	7.5	1.0	0.5	0.5	2.0	2.5	0.5	1.5	2.0	2.5	2.0	0.5	2.0	0	1.5	3.0	1.0	0.5	2.5	1.0	2.5
3	0.5	1.5	---	1.0	2.0	2.5	3.0	2.5	1.5	2.2	0	1.7	1.0	1.5	2.0	2.5	2.0	1.0	0.5	1.0	0	1.5	2.0	2.0	7.5	1.0	3.0	1.0	2.5	3.0
4	1.0	0	1.2	---	1.5	2.0	2.5	3.0	2.5	1.0	2.0	1.0	1.0	1.5	2.0	0.5	0.5	1.0	2.5	1.5	1.5	2.0	2.5	2.0	2.0	1.5	0.5	1.0	2.0	7.5
5	2.0	2.0	2.0	1.5	---	1.5	2.0	3.0	2.0	2.0	2.0	0.5	1.0	2.0	2.5	2.0	0.5	2.2	2.0	2.5	0.5	1.0	2.5	2.0	2.0	1.5	3.0	1.0	2.0	3.0
6	2.5	2.5	2.0	2.5	1.5	---	2.5	3.0	1.7	1.0	2.0	3.0	2.5	2.0	0.5	0.5	1.0	1.0	2.5	1.0	7.5	0.5	1.7	1.0	7.5	1.0	3.0	2.5	1.0	2.0
7	3.0	1.0	3.0	2.5	2.0	2.5	---	1.0	2.0	1.0	7.5	2.5	7.5	2.0	1.0	1.5	2.0	2.0	0.5	2.2	0	1.0	3.0	1.5	2.0	1.5	2.0	2.5	1.0	0.5
8	2.5	2.0	1.1	3.0	3.0	2.5	1.0	---	1.5	7.5	0	0.5	2.0	3.0	2.5	1.0	1.5	1.2	0.5	2.0	1.8	1.5	2.5	2.0	3.0	2.0	2.5	1.0	1.0	2.0
9	1.0	1.0	1.5	2.5	2.0	2.0	2.0	1.5	---	1.0	2.0	2.0	2.5	0.5	2.2	2.0	3.0	0.5	7.5	2.5	0.5	1.0	0	1.5	2.0	2.0	7.5	1.0	3.0	1.0
10	2.2	0.5	2.2	1.0	0.5	1.0	1.0	1.5	1.0	---	2.0	1.5	0	2.5	1.0	0.5	2.5	2.0	0.5	2.0	1.0	7.5	2.5	7.5	2.0	1.0	1.0	1.5	2.0	2.0
11	2.0	2.0	2.0	1.5	3.0	1.5	2.0	3.0	2.0	1.0	---	0.5	0.5	1.0	2.5	3.0	2.5	1.5	1.5	1.0	0	1.0	1.0	2.5	3.0	2.5	1.5	2.2	0	3.0
12	2.5	2.0	2.0	2.5	1.5	5.0	2.5	3.0	2.5	1.0	2.0	---	3.0	7.5	1.0	1.0	3.0	0.5	1.5	2.0	1.7	1.5	2.5	2.0	3.0	2.0	2.5	1.0	1.0	2.0
13	0.5	1.5	1.0	1.0	2.0	2.5	3.0	2.5	1.5	2.2	0	3.0	---	2.0	1.5	3.0	1.5	2.5	2.2	1.0	0	1.0	2.0	2.5	3.0	7.5	1.5	2.2	2.0	3.0
14	1.5	3.0	1.0	2.5	1.5	2.0	2.0	3.0	2.5	1.0	2.0	3.0	2.0	---	1.0	3.0	0.5	1.5	1.0	1.0	1.0	7.5	2.5	7.5	2.0	1.0	1.0	1.5	2.0	2.0
15	2.0	2.0	2.0	1.5	0	1.5	2.0	3.0	1.0	1.0	2.0	0.5	2.0	1.0	---	1.0	3.0	1.5	3.0	2.0	2.2	0	1.0	1.0	1.5	2.0	1.2	2.0	1.0	0.5
16	2.5	2.0	2.0	1.0	1.5	2.5	2.5	3.0	2.5	1.0	2.0	1.0	3.0	7.5	1.5	---	3.0	0.5	1.5	2.0	0.5	1.0	0	1.5	2.5	2.0	7.5	1.0	3.0	1.0
17	3.0	1.0	3.0	2.5	2.0	2.5	0	1.0	2.0	3.0	7.5	0	2.5	3.0	2.0	2.2	---	0.5	2.5	1.0	1.0	1.5	2.5	2.0	3.0	2.0	2.5	1.0	1.0	2.0
18	2.5	2.0	0	3.0	0.5	2.5	1.0	2.2	1.5	7.5	1.5	3.0	3.0	2.0	1.5	1.0	1.5	---	2.0	1.0	1.0	7.5	2.5	7.5	2.8	2.8	1.0	1.5	2.0	2.0
19	1.0	1.0	1.5	2.5	2.0	3.0	2.0	2.5	1.0	1.0	2.0	3.0	7.5	2.5	1.0	3.5	0.5	2.0	---	1.0	2.2	0	1.0	1.0	1.5	2.0	2.5	2.0	1.0	0.5
20	2.2	7.5	2.2	1.0	1.7	1.5	1.0	7.5	1.0	0	2.0	3.0	2.5	1.0	1.0	3.0	0.5	1.5	2.0	---	0.5	1.0	0	1.5	0.5	2.0	7.5	1.0	3.0	1.0
21	2.5	2.0	2.0	2.0	1.5	2.2	2.5	0.5	2.5	1.0	2.0	3.0	2.5	2.0	0.5	0.5	1.0	1.0	2.5	1.0	---	0.5	1.0	1.0	7.5	2.5	3.0	2.5	1.0	2.0
22	3.0	1.0	3.0	2.5	2.0	2.5	2.7	1.0	2.0	1.0	7.5	2.5	7.5	2.0	1.0	1.0	1.5	2.0	2.0	0.5	2.2	---	1.0	1.0	1.5	2.0	2.5	2.0	1.0	0.5
23	2.5	1.0	2.2	3.0	3.0	2.5	1.0	2.5	1.5	7.5	1.0	0.5	2.0	3.0	2.5	1.0	1.5	1.2	0.5	3.0	1.0	1.5	---	2.0	3.0	2.0	2.5	1.0	1.0	2.0
24	1.0	1.0	1.5	2.5	2.0	3.0	2.0	1.5	2.5	1.0	2.0	2.0	2.5	0.5	1.0	2.0	3.0	0.5	1.8	2.5	0.5	1.0	0	---	2.0	2.0	7.5	1.0	3.0	3.0
25	2.5	0.5	0.5	1.0	2.0	2.5	0.5	2.5	1.0	2.2	0	2.5	2.0	2.5	1.0	1.0	1.5	1.5	2.5	2.5	1.5	0	2.0	2.5	---	3.5	1.0	1.5	1.0	3.0
26	2.0	1.7	1.5	0	2.0	2.0	1.0	2.0	1.0	7.5	1.0	0.5	0.5	2.0	2.5	0.5	1.5	0.5	2.5	2.0	2.0	2.0	0	1.5	1.2	---	0.5	2.5	1.0	2.0
27	0.5	1.5	2.7	1.0	2.0	2.5	3.0	2.5	1.5	2.2	0	1.0	1.5	2.0	2.5	2.0	1.0	0.5	1.0	0	2.5	2.0	2.5	2.0	7.5	1.0	---	1.0	2.5	2.5
28	2.2	0.5	2.2	1.0	1.0	1.0	1.0	1.5	1.0	1.7	2.0	1.5	0.5	2.5	1.0	0.5	2.5	3.0	0.5	2.0	1.0	2.0	2.5	7.5	2.0	1.0	1.0	---	2.0	3.0
29	2.0	2.0	2.0	1.5	0	1.5	2.0	3.0	2.0	1.0	2.7	0.5	0.5	1.0	2.5	3.0	2.5	1.5	1.5	1.0	0	1.0	1.0	2.5	3.0	2.5	1.5	7.5	---	2.0
30	2.5	2.0	2.0	2.5	1.5	0	2.5	3.0	2.7	1.0	2.0	2.5	3.0	7.5	2.0	2.0	3.0	0.5	1.5	2.0	1.0	1.5	2.5	2.0	3.0	2.0	2.5	1.0	1.0	---

"---": Not applicable * Initial setup times

Table 3-3 Tool and raw material cost data (\$)

Tool	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Q_i	10	12	14	5	8	14	5	10	7	6	8	7	15	12	11	10	5	6	8	14
Job	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
π_j (x10)	18	25	16	20	40	35	15	10	12	60	22	18	24	44	45	57	28	35	40	20
Job	21	22	23	24	25	26	27	28	29	30										
π_j (x10)	10	30	50	38	58	44	21	25	40	30										

3.5.1 Example problem for Model 3-1

In this section, the proposed algorithm is applied to solve an illustrative problem for Model 3-1, i.e., planning for minimization of expected production cost. To examine the impact of tool spare level on the total expected production cost and average production reliability, seven tool spare levels, $M_i = 0, 1, 2, 3, 4, 5$, and 6, are used as tool resource constraints. The size of neighbourhood for the 30-job problem is $|N(s)| = 435$. This means that a total of 435 adjacent solutions must be evaluated before making a move.

The C code of the algorithm was run on a 486 PC. To test the effects of tabu list size and the diversification options on the search process, computations based on the following scenarios are performed:

Termination criterion	Search phase length N_{max}	Tabu list size T_size
$T_{max} = 20$ min.	M_big	7,10,20,25,50
$M_{max} = 200$ moves	20,40,100,200	5,10,20

For the T_{max} termination criterion, if a small phase, e.g., 20 moves, is used the last phase may be incomplete. The reason is that the search process will stop whenever the total search time reaches 20 min. regardless of the number of moves completed for the last phase. Therefore, we specify the entire search process as a single phase by using a sufficiently large N_{max} as the phase length. For the M_{max} criterion, tabu list sizes greater than 20 may not be used since otherwise all moves would become tabu (notice that the smallest phase size is 20).

The computational results are summarized in Tables 3-4 and 3-5. Table 3-4 shows the final job sequence for each tool spare level based on T_{max} termination criterion. Only the solutions associated with the best tabu list sizes are listed (details are listed in Table 3-6). Table 3-4 also shows the process reliability for each job, average reliability for processing the 30 jobs, actual tool cost, and total expected production cost for each job sequence and tool spare level. The tool replacement intervals and the actual tool consumption are shown in Table 3-5 for $M_i=6$. Based on the output in Tables 3-4 and 3-5, the job sequencing and tool replacement decisions can be made simultaneously.

Table 3-4 Solution for the Model 3-1 example problem with different spare levels

M_i	0	1	2	3	4	5	6
	Seq. Rel.	Seq. Rel.	Seq. Rel.	Seq. Rel.	Seq. Rel.	Seq. Rel.	Seq. Rel.
19	.93	25 .86	19 .93	30 .94	30 .94	18 .88	30 .94
16	.85	16 .85	16 .85	19 .93	20 .87	26 .83	10 .81
10	.70	4 .73	23 .88	16 .85	22 .84	29 .88	5 .88
23	.86	5 .86	17 .87	5 .88	4 .78	19 .92	15 .91
22	.60	15 .89	5 .87	4 .69	5 .84	1 .79	3 .79
14	.67	20 .79	20 .75	3 .83	15 .91	15 .90	1 .79
26	.55	24 .90	1 .76	21 .71	24 .90	7 .78	9 .67
29	.65	8 .68	24 .91	13 .78	13 .79	21 .81	29 .86
17	.60	27 .77	8 .63	9 .83	9 .75	3 .70	16 .87
30	.90	21 .71	4 .71	10 .70	10 .81	16 .86	11 .86
24	.83	10 .79	3 .79	8 .79	8 .72	11 .84	8 .73
5	.54	30 .91	12 .80	7 .70	17 .89	28 .76	28 .79
11	.49	14 .82	26 .82	15 .70	7 .77	5 .87	24 .90
15	.67	22 .63	9 .69	14 .92	18 .86	12 .84	12 .82
25	.69	28 .53	15 .91	12 .85	16 .86	6 .88	19 .93
13	.60	26 .61	27 .81	27 .84	2 .72	10 .82	17 .90
27	.56	6 .82	21 .75	22 .81	21 .82	8 .74	23 .86
28	.29	23 .72	22 .84	23 .86	14 .85	17 .89	7 .81
21	.48	17 .66	10 .71	17 .87	12 .81	30 .93	27 .83
3	.41	29 .62	28 .59	20 .86	25 .86	4 .80	2 .74
4	.23	2 .36	18 .86	28 .81	3 .73	14 .84	21 .76
20	.30	19 .82	14 .70	25 .78	23 .89	25 .86	18 .86
2	.24	18 .79	29 .73	18 .84	1 .78	20 .81	4 .79
9	.32	11 .51	25 .86	26 .86	29 .85	24 .92	22 .85
8	.20	13 .69	2 .51	29 .76	26 .81	13 .82	14 .87
18	.55	1 .60	30 .88	24 .80	19 .89	23 .89	20 .83
6	.49	3 .53	6 .78	6 .91	27 .82	27 .83	26 .82
12	.42	9 .46	11 .60	11 .81	28 .79	22 .83	6 .87
7	.36	12 .51	13 .77	1 .73	11 .75	2 .71	25 .85
1	.31	7 .47	7 .67	2 .78	6 .86	9 .72	13 .81
TC(\$)	0	187	364	472	552	578	594
EC(\$)	6175	4344	3646	3265	3214	3155	3153
AR(%)	54.5	69.5	77.5	81.4	82.6	83.1	83.4

Note: (1) The results are based on the best tabu list sizes and 20 min. termination criterion
(2) M_i : Tool spare level TC: Tool spare cost
EC: Expected production cost AR: Average process reliability

Table 3-5 Tool replacement intervals and spare requirements ($M_1=6$)

Tool type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Job No.	Replacement intervals																			
30	--	--	--	0	--	--	0	0	--	--	--	0	0	--	--	--	0	0	--	--
10	0	--	0	--	--	0	1	--	--	0	--	--	0	0	--	--	--	--	--	--
5	--	0	0	--	0	--	--	--	0	--	0	--	--	--	0	0	--	--	--	--
15	1	0	--	1	0	--	--	--	--	1	0	--	0	--	--	--	0	--	--	0
3	0	0	--	0	0	--	--	--	--	0	0	--	0	--	--	--	0	--	--	0
1	--	0	--	--	--	0	--	0	--	--	--	1	--	--	--	--	--	0	0	0
9	0	0	--	0	0	--	--	--	--	0	0	--	0	--	--	--	0	--	--	0
29	--	1	1	--	1	--	--	--	1	--	1	--	--	--	0	0	--	--	--	--
16	1	--	1	--	--	1	0	--	--	0	--	--	1	1	--	--	--	--	--	--
11	--	0	0	--	0	--	--	--	0	--	0	--	--	--	1	0	--	--	--	--
8	--	--	0	0	--	--	--	--	0	--	--	--	--	0	0	--	1	--	0	--
28	0	--	1	--	--	0	0	--	--	1	--	--	0	1	--	--	--	--	--	--
24	--	--	--	0	--	--	0	1	--	--	--	0	0	--	--	--	0	1	--	--
12	--	--	--	0	--	--	1	0	--	--	--	0	0	--	--	--	0	0	--	--
19	--	0	--	--	--	1	--	0	--	--	--	1	--	--	--	--	--	0	1	1
17	--	0	0	--	0	--	--	--	1	--	0	--	--	--	1	1	--	--	--	--
23	--	0	1	--	0	--	--	--	0	--	0	--	--	--	0	0	--	--	--	--
7	--	0	--	--	--	0	--	0	--	--	--	0	--	--	--	--	--	0	0	0
27	1	0	--	1	0	--	--	--	--	1	0	--	0	--	--	--	1	--	--	0
2	--	--	0	0	--	--	--	--	0	--	--	--	--	0	1	--	0	--	1	--
21	0	0	--	0	0	--	--	--	--	0	0	--	0	--	--	--	0	--	--	0
18	--	--	--	0	--	--	1	1	--	--	--	0	1	--	--	--	0	1	--	--
4	0	--	0	--	--	0	1	--	--	0	--	--	0	1	--	--	--	--	--	--
22	1	--	1	--	--	1	0	--	--	0	--	--	0	0	--	--	--	--	--	--
14	--	--	0	1	--	--	--	--	1	--	--	--	--	1	1	--	1	--	1	--
20	--	--	1	0	--	--	--	--	0	--	--	--	--	0	0	--	0	--	0	--
26	--	--	0	0	--	--	--	--	1	--	--	--	--	1	1	--	0	--	1	--
6	--	--	--	0	--	--	0	0	--	--	--	1	0	--	--	--	0	0	--	--
25	--	1	--	--	--	0	--	0	--	--	--	0	--	--	--	--	--	0	1	1
13	--	0	--	--	--	0	--	1	--	--	--	1	--	--	--	--	--	1	0	0
spares	4	2	6	3	1	3	4	3	4	3	1	3	2	5	5	1	3	3	5	2

" -- " no operation is assigned

" 1 " tool replacement is required

" 0 " no tool replacement is required

3.5.2 Example problem for Model 3-2

In section 3.4.1, a tabu search algorithm was developed to solve the general case of sequencing problem with tool replacement consideration. This algorithm may also be used to solve Model 3-2 which includes minimization of expected production cost and JIT sequencing. The main modification required for the proposed tabu search algorithm is in the cost function, $G(s)$, which is used to calculate the cost of each sequence. Here, the cost of each candidate move, $G(s)$, is calculated based on the objective function of Model 3-2 (Equation 3-9) and the constraints therein. In addition, different set of information including the completion times, and earliness and tardiness of jobs have to be recorded and kept track of during the search process.

The following example illustrates the application of the tabu search algorithm in solving Model 3-2. To solve this problem, additional data regarding job due dates and earliness and tardiness penalties are needed. These data are given in Table 3-6.

Table 3-6 Due dates and cost data for the Model 3-2 example problem

Job	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
d_j (min)	72	360	300	30	270	330	432	390	180	210	228	420	210	465	315
a_j (\$/min)	5.0	4.0	4.0	1.5	6.5	4.0	6.0	4.5	5.0	4.5	1.5	2.0	4.0	3.5	1.0
b_j (\$/min)	5.0	3.0	3.0	1.5	1.5	4.0	3.0	2.0	3.0	1.0	1.0	0.5	3.0	0.5	1.0
Job	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
d_j (min)	150	210	282	480	405	30	120	240	480	360	345	60	90	240	144
a_j (\$/min)	8.5	2.5	3.0	8.0	4.0	5.5	6.0	5.5	4.5	7.0	5.0	5.0	5.5	3.5	5.0
b_j (\$/min)	0.5	1.0	0.5	1.0	1.5	2.0	2.0	2.0	1.0	4.0	2.0	1.0	1.5	2.5	1.5

For this example, the computations were performed for 15 minutes in a single phase to find the best search parameters. To determine the actual tool spares requirement without any restriction, the available tool copies for each tool type, M_i , were set at a high number (8 copies for each tool type). The results for the best tabu list size and initial solution are displayed in Table 3-7.

As shown by Table 3-7, the job sequence is quite different from that obtained by solving Model 3-1, i.e., when the objective is solely to minimize expected production cost. The total expected cost for this sequence is \$5085 which includes \$654 tool spare cost and \$1795 earliness and tardiness penalties. The added JIT objective has also increased the tool spare cost while the average process reliability slightly decreases to about 82.5%. In this sequence, a few jobs (jobs 10, 11, 12, and 15) have very high deviations from their due dates. The interesting point about this situation is that these are the jobs with relatively low penalties. As a result, in the final solution they take large deviations from their due dates so that the overall earliness/tardiness cost of the sequence is minimized. This complies with the objective of the proposed model that aims to minimize the total weighted earliness and tardiness of all jobs in the final solution.

Table 3-7 Solution for Model 3-2 example problem ($M_i = 8$)

Tool type		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Seq ^(a)	ET	Replacement intervals																			
4	16	0	---	0	---	---	0	0	---	---	0	---	---	0	0	---	---	---	---	---	---
21	-6	0	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	0	---	---	0
27	2	1	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	0	---	---	0
1	0	---	0	---	---	---	0	---	0	---	---	---	0	---	---	---	---	---	0	0	0
28	0	0	---	0	---	---	0	0	---	---	0	---	---	0	0	---	---	---	---	---	---
22	12	1	---	0	---	---	1	0	---	---	1	---	---	1	1	---	---	---	---	---	---
16	29	1	---	0	---	---	1	0	---	---	1	---	---	0	1	---	---	---	---	---	---
30	3	---	---	---	0	---	---	0	1	---	---	---	0	0	---	---	---	0	1	---	---
10	53	1	---	1	---	---	0	1	---	---	1	---	---	0	1	---	---	---	---	---	---
9	0	0	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	0	---	---	0
17	12	---	1	0	---	1	---	---	---	0	---	1	---	---	---	0	0	---	---	---	---
13	-4	---	0	---	---	---	0	---	0	---	---	---	0	---	---	---	---	---	0	0	1
23	11	---	0	0	---	1	---	---	---	0	---	0	---	---	---	0	0	---	---	---	---
29	-8	---	0	0	---	0	---	---	---	0	---	0	---	---	---	1	1	---	---	---	---
5	3	---	0	1	---	0	---	---	---	0	---	0	---	---	---	1	0	---	---	---	---
18	-5	---	---	---	1	---	---	0	1	---	---	---	0	0	---	---	---	0	1	---	---
11	-77	---	0	0	---	0	---	---	---	1	---	0	---	---	---	0	0	---	---	---	---
3	-26	1	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	1	---	---	0
6	-14	---	---	---	1	---	---	1	0	---	---	---	1	0	---	---	---	0	0	---	---
26	-18	---	---	1	0	---	---	---	---	0	---	---	---	---	1	1	---	1	---	1	---
25	-16	---	1	---	---	---	1	---	0	---	---	---	1	---	---	---	---	0	1	0	---
2	-36	---	---	0	0	---	---	---	---	0	---	---	---	---	1	0	---	0	---	0	---
8	-25	---	---	0	0	---	---	---	---	0	---	---	---	---	1	0	---	0	---	0	---
20	-31	---	---	0	0	---	---	---	---	1	---	---	---	---	0	1	---	0	---	1	---
7	-19	---	0	---	---	---	0	---	0	---	---	---	0	---	---	---	---	---	1	0	0
14	-5	---	---	1	0	---	---	---	---	1	---	---	---	---	1	1	---	1	---	1	---
19	-3	---	0	---	---	---	0	---	0	---	---	---	1	---	---	---	---	---	0	1	1
24	-22	---	---	---	1	---	---	1	0	---	---	---	0	1	---	---	---	1	0	---	---
12	-102	---	---	---	0	---	---	1	1	---	---	---	0	0	---	---	---	0	1	---	---
15	-226	1	1	---	0	0	---	---	---	---	1	1	---	0	---	---	---	0	---	---	0
Spares ^(b)		6	3	5	3	1	3	4	3	3	4	2	3	2	7	5	1	4	4	5	2

"---" no operation is assigned

"1" tool replacement is required

"0" no tool replacement is required

^(a) : Total expected cost = \$5085

^(b) : Tool spare cost = \$654

ET: Tardiness or earliness of jobs

3.5.3 Example problem for Model 3-3

In this section an example problem will be solved to demonstrate the application of the proposed solution procedure for Model 3-3. In this model, the objective is to simultaneously determine the sequence of jobs and the tool replacement intervals so that the cumulative expected production cost and flow time cost are minimized. Once again the cost function, $G(s)$, in the original algorithm has to be modified to accommodate the objective function and constraints developed in Section 3.3.3.

In order to calculate the job flow cost in this model, the unit flow cost for each job has to be known. These weights are given in the following table.

Table 3-8 Job flow cost for Model 3-3 example problem

Job	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_j (\$/min)	1.0	0.4	0.7	0.3	0.6	0.2	0.4	1.0	0.2	0.5	1.1	0.8	0.5	1.0	0.2
Job	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
u_j (\$/min)	0.5	0.4	0.6	1.0	0.5	0.9	0.2	1.2	0.9	0.5	0.5	0.3	1.2	1.0	0.5

The rest of the required data are the same as those given in Section 3.5. The computational results based on 15 minutes search time and the best search parameters are presented in Table 3-9.

Table 3-9 Solution for Model 3-3 example problem ($M_i = 8$)

Tool type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Seq ^(a) FT	Replacement intervals																			
23 14	---	0	0	---	0	---	---	---	0	---	0	---	---	---	0	0	---	---	---	---
19 26	---	0	---	---	---	0	---	0	---	---	---	0	---	---	---	---	---	0	0	0
11 45	---	0	0	---	0	---	---	---	0	---	0	---	---	---	1	0	---	---	---	---
1 61	---	0	---	---	---	0	---	0	---	---	---	0	---	---	---	---	---	0	0	0
14 80	---	---	1	0	---	---	---	---	0	---	---	---	---	0	1	---	0	---	1	---
28 99	0	---	1	---	---	0	0	---	---	0	---	---	0	0	---	---	---	---	---	---
8 118	---	---	0	0	---	---	---	---	0	---	---	---	---	0	0	---	0	---	0	---
29 137	---	0	0	---	0	---	---	---	1	---	0	---	---	---	1	1	---	---	---	---
24 157	---	---	---	0	---	---	0	1	---	---	---	0	0	---	---	---	1	1	---	---
21 178	0	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	0	---	---	0
16 190	1	---	1	---	---	1	1	---	---	1	---	---	1	1	---	---	---	---	---	---
12 208	---	---	---	0	---	---	0	0	---	---	---	0	0	---	---	---	0	0	---	---
25 224	---	1	---	---	---	1	---	0	---	---	---	1	---	---	---	---	---	0	1	1
13 240	---	0	---	---	---	0	---	1	---	---	---	0	---	---	---	---	---	1	0	0
18 260	---	---	---	1	---	---	1	1	---	---	---	0	0	---	---	---	1	1	---	---
3 279	0	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	0	---	---	0
26 298	---	---	0	1	---	---	---	---	0	---	---	---	---	1	1	---	1	---	1	---
5 318	---	1	0	---	1	---	---	---	1	---	1	---	---	---	1	0	---	---	---	---
10 334	1	---	1	---	---	1	1	---	---	1	---	---	1	1	---	---	---	---	---	---
20 355	---	---	0	1	---	---	---	---	0	---	---	---	---	0	0	---	0	---	0	---
30 375	---	---	---	0	---	---	0	0	---	---	---	1	0	---	---	---	0	0	---	---
17 393	---	0	0	---	0	---	---	---	1	---	0	---	---	---	1	0	---	---	---	---
2 413	---	---	1	0	---	---	---	---	0	---	---	---	---	1	0	---	1	---	1	---
4 427	1	---	0	---	---	0	1	---	---	0	---	---	0	1	---	---	---	---	---	---
15 447	1	0	---	0	0	---	---	---	---	1	0	---	0	---	---	---	0	---	---	1
7 463	---	0	---	---	---	0	---	0	---	---	---	0	---	---	---	---	---	0	0	0
27 482	0	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	0	---	---	0
22 502	0	---	0	---	---	1	0	---	---	0	---	---	1	0	---	---	---	---	---	---
9 525	0	0	---	0	0	---	---	---	---	0	0	---	0	---	---	---	0	---	---	0
6 544	---	---	---	0	---	---	0	1	---	---	---	0	0	---	---	---	0	1	---	---
Spares ^(b)	4	2	5	3	1	4	4	4	3	3	1	2	3	5	6	1	4	4	4	2

"—" no operation is assigned

" 1 " tool replacement is required

" 0 " no tool replacement is required

^(a) : Total expected cost =\$7126

^(b) : Tool spare cost =\$619

FT : Flow time of jobs (min.)

Table 3-9 shows the final sequence of jobs, the associated tool replacement intervals and the actual tool consumption for each tool type. The total expected cost is \$7126 in which \$619 is the total spare cost, \$3810 is the jobs flow cost and rest is the expected processing cost. The average reliability for processing the jobs in the above sequence is 83%. As expected, in this sequence those jobs with the higher flow costs (e.g., jobs 23, 19, and 11) are given higher priority in terms of processing sequence while the jobs with the lower flow costs are left to be processed at the end. It should be pointed out that in the example problems for Models 3-2 and 3-3 there is no restriction on the number of available copies for each tool type. By doing so the actual number of required spares can be determined. However, this causes also the earliness and tardiness penalties (in Model 3-2) and flow time cost (in Model 3-3) to play a predominate role in determining the final sequence of jobs in these examples.

3.6 Discussion

The following findings perceived from the computational results could be useful for shop floor decisions. The discussions made in this section are mostly based on the computational results obtained for the example problem of Model 3-1. A part of the work presented in this chapter has been reported in Kolahan *et al.* (1995) and Kolahan and Liang (1994).

1. The performance of the machining centre is directly affected by tool spare level

The total expected production cost can be significantly reduced and a more

reliable production process can be achieved by increasing tool spare level. For instance, as shown in Table 3-4, if only one spare is provided for each tool type, the expected production cost is \$4,344 and the average process reliability (the arithmetic average of the process reliabilities of all the 30 jobs) is only 69.5%. As the spare level increases to 2, the total expected production cost reduces to \$3,646 while the average process reliability increases to 77.5%. The spare tool costs obtained for the example problems are the tool costs which are actually required rather than that of the available spare tools.

2. The effect of tool spare has a saturation point

As can be seen in Table 3-4, although the performance of the machining centre can be further improved by providing more spare tools, the marginal effect diminishes gradually as more spare tools are added. To explore this phenomenon, we performed additional computations by increasing tool spare level up to 10. The results are plotted in Figure 3-1. As shown in this figure, when the tool spare level reaches a certain point, 6 spare tools in this case, this effect becomes saturated. That is, the extra spare tools no longer bring any benefit and thus are redundant.

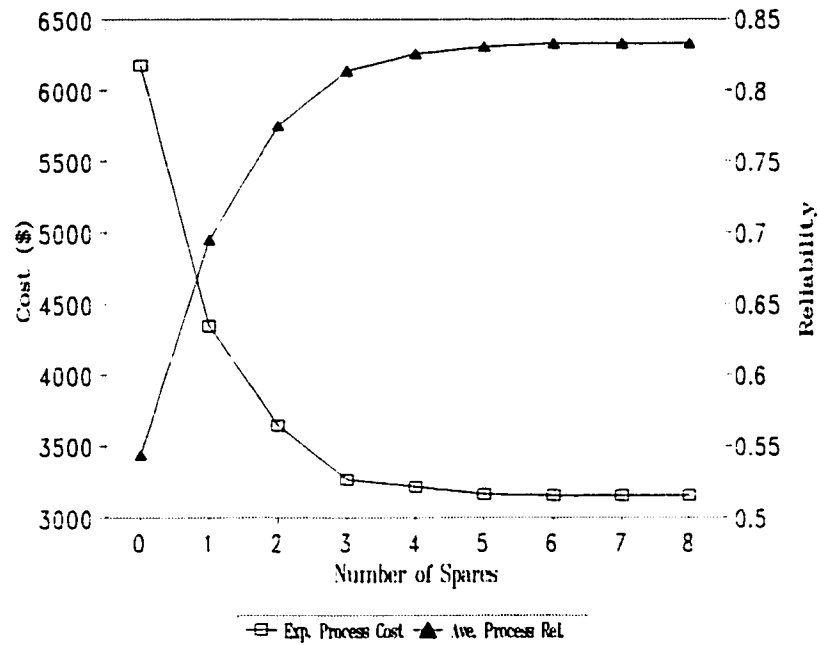


Figure 3-1 Process reliability and expected production cost with different tool spare levels

3. Tool replacement intervals are not fixed

This is shown in Table 3-5. Consider tool type 2 which is required by jobs #5, #15, #3, #1, #9, #29, #11, #19, #17, #23, #7, #27, #21, #25 and #13 positioned 3rd, 4th, 5th, 6th, 7th, 8th, 10th, 15th, 16th, 17th, 18th, 19th, 21st, 29th, and 30th, respectively, in the sequence. The first tool replacement is required before processing job #29, the sixth job to be processed by tool type 2. The replacement interval is 13.8 minutes ($=T_{2,5}+T_{2,15}+T_{2,3}+T_{2,1}+T_{2,9}$) in terms of cutting time, and 5 jobs in terms of workpieces. While the second tool replacement is planned before processing job #25, the eighth job to be machined by the second copy of tool type 2. The replacement interval now becomes 19.4 minutes in terms of cutting time and 7 jobs in terms of workpieces.

Hence the two consecutive tool replacements for tool type 2 are different in terms of both cutting time and number of completed jobs. The variable replacement intervals are caused by the fact that each tool is used for different jobs. This suggests that the traditional tool replacement strategies such as scheduled tool replacement (replace a tool after a fixed time interval) and preventive planned tool replacement (replace a tool whenever a fixed number of jobs have been processed by the concerned tool) may not be suitable for multi-tool-type and multi-job-type problems.

Further, the tool provisioning decisions can also be made based on the expected tool consumption shown in Table 3-5: six spare tools of type 3 are required while only one spare is required for tool types 5,11 and 16. Hence, additional spare may be unnecessary.

3.7 Performance of the tabu search algorithm

1. Convergence

To examine the performance of the proposed tabu search algorithm, three convergence curves (associated with $M_i = 0,2,6$) for the Model 3-1 example problem are plotted in Figure 3-2. The convergence curves are based on the outputs corresponding to the best search parameters. The curves clearly demonstrate the cost improvement process. In all the three cases, most of the cost improvements are achieved within the first 200 seconds. No significant cost reduction can be achieved thereafter. To further examine the convergence behaviour, we have run the program with $M_i=6$ for 24 hours and the same was observed.

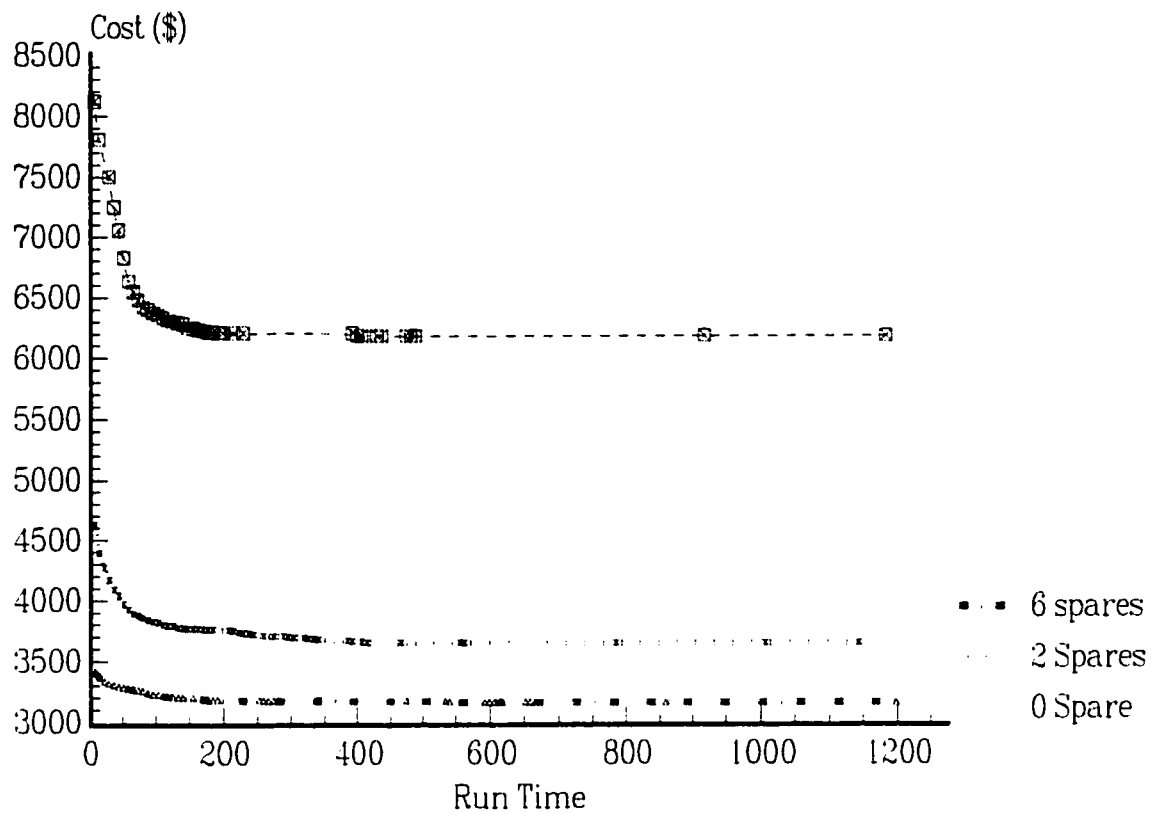


Figure 3-2 Convergence curves for the 20-tool example problem

To verify the feasibility of solving larger problems, a 100-tool problem has also been solved on the 486 PC and most improvement was achieved within the first 3 minutes of search time. The convergence curves ($M_i = 0,5,10$) for the 100-tool problem are shown in Figure 3-3.

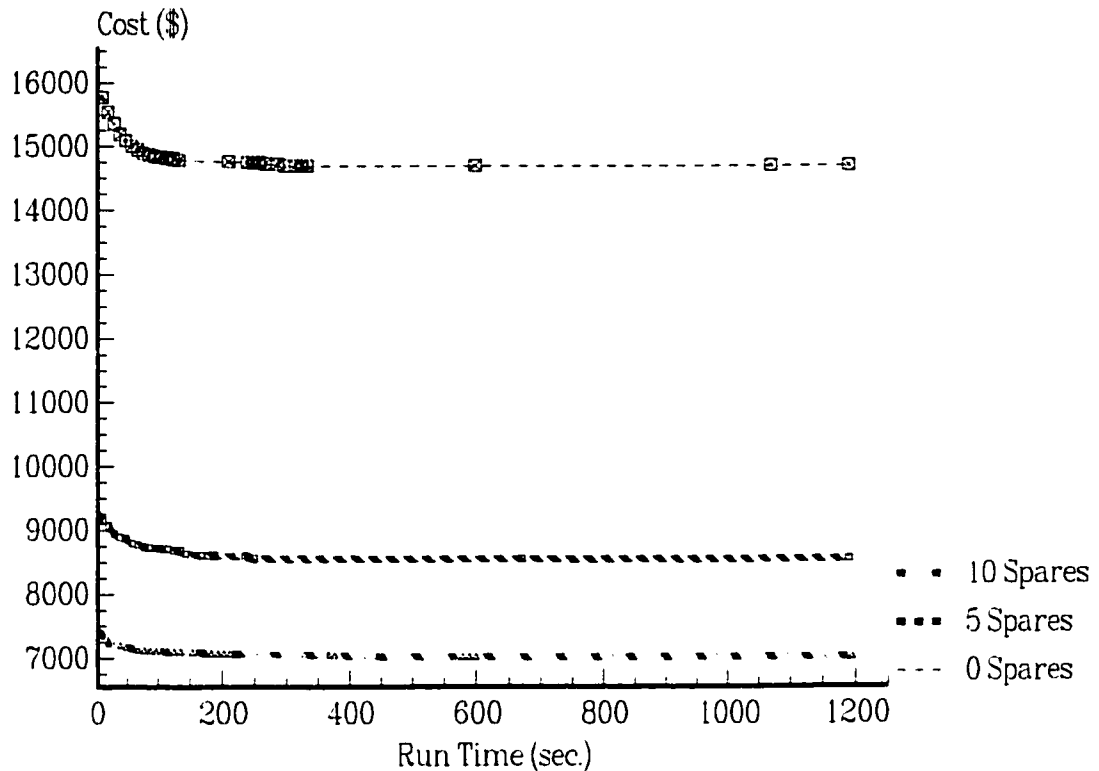


Figure 3-3 Convergence curves for the 100-tool example problem

2. Accuracy

The accuracy of the heuristic solutions is validated by comparing the optimal solution with the heuristic solution for a set of 10-job problems. The optimal solution of the 10-job problem is obtained by enumerating all possible ($10! = 3628800$) sequences. On the average it takes more than 46 hours to get optimal solution on a 486 PC using a C code. With the proposed algorithm, an accuracy of less than 2% error is obtained within 1 minute of search time.

Though the above observations cannot be generalized, we do believe this heuristic

can provide quick and very good solutions to such complicated problems.

3. Effect of tabu list size

Computational results for the Model 3-1 example problem with different tabu list sizes are summarized in Table 3-10. All runs are terminated after 20 minutes of search. Table 3-10 indicates that, for a fixed search duration, longer tabu lists generally provide better solutions, though the improvement is not very notable. However, since the search times are the same, longer tabu lists may be preferred.

Table 3-10 The effect of tabu list size on search performance

Tabu list Size	Expected production cost when $M_i =$						
	0	1	2	3	4	5	6
7	6203	4419	3645	3275	3238	3183	3172
10	6203	4419	3647	3351	3238	3154	3172
20	6203	4417	3647	3285	3238	3154	3155
25	6175	4417	3647	3285	3238	3154	3153
50	6175	4344	3645	3264	3214	3154	3155
100	6175	4344	3645	3266	3214	3154	3155

Further computations have also been carried out to investigate the effect of search path diversification. As mentioned earlier, the termination criterion is $M_{max} = 200$ moves and four phase sizes: 200,100,40, and 20 are used. After each search phase, the tabu list is cleared and the search restarts from the best job sequence obtained so far. The outputs for $M_i = 6$ are listed in Table 3-11. It is shown that the search is not sensitive

to this diversification strategy. This indicates that the "regular" tabu heuristic alone is sufficient to provide a reasonably good search path in this case.

Table 3-11 The effect of path diversification

Phase size	Expected production cost (\$) when tabu list size =		
	5	10	25
200	3172	3172	3153
100	3172	3155	3155
40	3172	3155	3155
20	3172	3155	3153

Chapter 4

OPERATION SEQUENCING WITH PROCESSING TIME ADJUSTMENT

4.1 Background

4.1.1 Scope and purpose

In Chapter 3, the collective effect of tool replacement decision and job sequencing on the total expected production cost has been analyzed. While the objective was to minimize the total processing cost, the models were formulated to address different issues related to the tool replacement decision in job sequencing. In the previous chapter, the processing times are treated as sequence-dependent but not decision variables. Due to the wide application of NC technology, processing time can be controlled to achieve modern planning goals such as Just-In-Time (JIT) delivery. The outcome of the last chapter, however, cannot be used for this purpose. It is therefore necessary to develop models and solution methods for operation sequencing with controllable processing times.

The main thrust of this chapter is devoted to achieve JIT production goal by utilizing the controllability of processing times. Associated solution method based on tabu search has been developed to solve large size problems. Mathematical models and tabu search algorithms have also been developed to solve scheduling problems with other planning goals such as minimization of weighted flow time in such a manufacturing context.

4.1.2 Process planning and JIT goal

The advent of automated manufacturing centres has driven manufacturing companies to adopt modern production philosophies and technologies. In particular, just-in-time (JIT) philosophy and CNC technology have been widely applied in recent years. In a JIT production environment, neither earliness nor tardiness is desirable. While a late job leads to loss of customer goodwill and the cost of hastening the shipment, an early job may result in such costs as inventory holding and cash commitment.

NC technology is commonly used in machining centres to improve process productivity and controllability. However, the good controllability of machining centres in adjusting feed rate and spindle speed, and hence processing time, has not yet been fully utilized to assist in achieving modern production goals such as JIT production. This is reflected in most conventional machine scheduling research in which the processing time of each job is assumed to be pre-determined and beyond the manager's control. The processing time is often selected based on a preferred combination of machining parameters for each individual job in isolation and the job sequencing is carried out using the pre-specified processing times. Therefore, the machine-level planning such as JIT production have little influence on the job-level decisions such as processing time determination.

Although the scheduling problem with controllable processing times has been investigated by some researchers (e.g. Vickson 1980, Daniel and Sarin 1989, Lee 1990), most of the studies consider only the time compression to reduce the total tardiness and do not address JIT sequencing goal, i.e., minimization of both tardiness and earliness

costs. Moreover, the effect of sequence-dependent setup times has often been ignored and due dates are assumed to be common for all jobs in the majority of these studies.

In summary, most of the previous studies either consider a JIT objective while overlooking the time controllability or explore time controllability (mostly only compression) without JIT objective. The sequence-dependent setup time is usually not taken into account. However, in many shop floor equipped with machining centres: a) both earliness and tardiness are undesirable; b) the processing time of a job can be both compressed and extended by adjusting feed rate, depth of cut and spindle speed at extra cost; and c) the setup time is usually sequence-dependent and each job may have its own distinct due date. Consequently, there is a need to further explore the scheduling problems arising in modern shop floors.

4.2 Problem statement

The problem under consideration involves sequencing of J jobs with distinct due dates, weighted earliness and tardiness penalties, and asymmetric sequence-dependent setup times on a machining centre. It is desirable to process each job just in time and any earliness or tardiness will lead to penalty cost. The setup times are sequence-dependent since the time required by changing tools and part holders depends, to some extent, on the similarity in tool and fixture requirements between the current job and the immediate previous one. The processing time of each job is controllable within a feasible range by adjusting feed rate, spindle speed and/or depth of cut. However, these adjustments would result in certain costs. The cost components to be considered are described below.

(a) *Tardiness and earliness penalties*: In JIT environment, if a job is completed prior to its due date, an earliness penalty will occur because of the increased inventory, cash commitment and possible shop floor congestion. The tardiness penalty usually occurs due to the loss of goodwill if the job is to be delivered to the customer or due to the waiting time if the job is to be processed by the next manufacturing stage. The amount of earliness or tardiness of a job depends on the processing and setup times of the concerned job and those of all the other jobs processed prior to it.

(b) *Compression and extension costs*: Generally, in a machining process, the processing cost of a job consists of: operating cost, tool cost and defective cost. The operating cost includes labour cost and machine overhead, and increases with time. The defective cost is mainly caused by severe tool wear and in-process tool breakages. When the process is sped up, the tool and defective costs increase due to the increased tool wear and in-process tool failure. In this study the *normal processing time* is defined as the processing time that minimizes the sum of these costs for a single job. Any deviation from the normal processing time will cause extra cost. The *additional* cost due to process speedup is referred to as the *compression cost* and the one caused by process slowdown is called *extension cost*.

Although the normal processing time is preferred for a single job, it may not necessarily be beneficial at the machine-level when all jobs' due dates are taken into account. It may be desirable to choose processing times other than the normal processing times for some jobs so that the sum of the earliness and tardiness penalties, and compression and extension costs is minimized at machine-level. The processing time

selection also has impact on the optimal job sequence. The optimal job sequence obtained based on a particular set of processing times may no longer be optimal when a different set of processing times is used.

Even in their simplest form, the sequencing problems involving tardiness and earliness are combinatorial in nature (Lenstra *et al* 1977, Garey *et al* 1988, and Oguz and Dincer 1994). The problem under consideration is a more general one which includes both sequence-dependent setups and earliness-tardiness issues and is even further complicated by the variable processing times. Hence any meaningful size problem may not be solved optimally within a reasonable computational time. In view of this, a tabu search approach is proposed to jointly address the job sequencing and processing time selection problems. The objective is to find the best trade-off between the JIT goal and the processing time compression and extension costs. The details are explained in following sections.

4.3 Problem formulation and analysis

4.3.1 Planning for JIT production

The JIT sequencing problem with variable processing times and sequence-dependent setup times can be formulated as the following nonlinear mixed integer programming model:

Model 4-1

$$\begin{aligned} & \text{Min } \{G(s)\} = \\ & \text{Min } \sum_{s=1}^{J!} W_s \sum_{l=1}^J (a_{\mu(s,l)} T_{\mu(s,l)} + b_{\mu(s,l)} E_{\mu(s,l)} + \alpha_{\mu(s,l)} X_{\mu(s,l)} + \beta_{\mu(s,l)} Y_{\mu(s,l)}) \end{aligned} \quad (4-1)$$

Subject to:

$$P_{\mu(s,l)} + X_{\mu(s,l)} \geq P_{\mu(s,l)}^N \quad \forall s, l \quad (4-2)$$

$$P_{\mu(s,l)} - Y_{\mu(s,l)} \leq P_{\mu(s,l)}^N \quad \forall s, l \quad (4-3)$$

$$P_{\mu(s,l)}^L \leq P_{\mu(s,l)} \leq P_{\mu(s,l)}^U \quad \forall s, l \quad (4-4)$$

$$\sum_{ll=1}^l (t_{\mu(s,ll-1)\mu(s,ll)} + P_{\mu(s,ll)}) - d_{\mu(s,l)} \leq T_{\mu(s,l)} \quad \forall s, l \quad (4-5)$$

$$d_{\mu(s,l)} - \sum_{ll=1}^l (t_{\mu(s,ll-1)\mu(s,ll)} + P_{\mu(s,ll)}) \leq E_{\mu(s,l)} \quad \forall s, l \quad (4-6)$$

$$E_{\mu(s,l)} \geq 0 \quad \forall s, l \quad (4-7)$$

$$T_{\mu(s,l)} \geq 0 \quad \forall s, l \quad (4-8)$$

$$\sum_{s=1}^{J!} W_s = 1 \quad (4-9)$$

Where

$$t_{\mu(s,ll-1), \mu(s,ll)} = t_{0 ll} \text{ when } ll=1.$$

The objective function contains four terms. The first two terms are respectively associated with tardiness and earliness while the last two are related to compression and extension costs. The objective function, in conjunction with constraint (4-9), states that only the sequence which minimizes the four cost components should be selected. Constraints (4-2) and (4-3) specify the relationship between the actual processing time and the normal processing time. The allowable range of processing

time is given by constraint (4-4). Constraints (4-5) and (4-6) define the relationship between a job's completion time, due date, tardiness and earliness. The completion time includes cumulative setup and processing times. It should be pointed out that the setup time of a job is not included in the processing time since it is sequence-dependent. It is also noted that the tardiness and earliness of a job cannot co-exist. This is ensured by jointly considering constraints (4-5) to (4-8).

Model 4-1 aims at simultaneously minimizing the cost associated with earliness and tardiness penalties as well as the extension and compression costs. If, however, JIT sequencing is the predominant goal, the number of JIT-sequenced jobs can be increased by using the following model:

Model 4-1-1

$$\underset{1 \leq s \leq J!}{Min} \{G(s)\} = \underset{1 \leq s \leq J!}{Min} \sum_{s=1}^{J!} W_s \sum_{l=1}^J (a_{\mu(s,l)} T_{\mu(s,l)} + b_{\mu(s,l)} E_{\mu(s,l)}) \quad (4-10)$$

Subject to constraints (4-2) to (4-9).

4.3.2 Planning for minimization of weighted flow time

Another commonly used measure for evaluating system performance is the job flow time (FT) which is the cumulative flow times of all jobs in the system. This planning goal represents service quality from the customer's point of view. In many cases it may be necessary to process and ship each job to the customer as soon as possible. Under such circumstances, the schedule's cost is directly related to the time each job

spends in the manufacturing system. Therefore, minimizing FT is an important planning goal to ensure customer satisfaction on one hand and to reduce work in process inventory on the other.

Minimization of flow time, or mean flow time, is a classic problem. It has been proven that shortest processing time (SPT) rule minimizes mean flow time where job processing times are fixed and setup times are not sequence dependent. Vikson (1980) has extended this problem for the case of equal unit flow cost and compressible job processing times. He demonstrated that this problem can be formulated as an assignment problem which is easily solvable. More recently, Lee (1991) studied the same problem to determine the tolerance range of job processing times for a given sequence. In both studies, it is shown that in the optimal sequence each job is either processed in normal processing time or compressed to its minimum allowed processing time. Nevertheless, in these studies the importance of sequence-dependent setup times has been overlooked. In addition, the flow costs, or weights, was assumed to be equal for all operations. These again undermine the practical usefulness of the results.

In view of the above, this thesis extends the FT concept to a broader scope: weighted flow time (WFT) with weighted processing time compression. The model incorporating these cost components is presented below:

Model 4-2

$$\begin{aligned} \text{Min}_{1 \leq s \leq J!} \{G(s)\} = \\ \text{Min} \left\{ \sum_{s=1}^{J!} W_s \left[\sum_{l=1}^J u_{\mu(s,l)} \sum_{ll=1}^l (t_{\mu(s,ll-1)\mu(s,ll)} + P_{\mu(s,ll)}) + \sum_{l=1}^J \alpha_{\mu(s,l)} X_{\mu(s,l)} \right] \right\} \end{aligned} \quad (4-11)$$

Subject to:

$$P_{\mu(s,l)} + X_{\mu(s,l)} \geq P_{\mu(s,l)}^N \quad \forall s, l \quad (4-12)$$

$$P_{\mu(s,l)} \geq P_{\mu(s,l)}^L \quad \forall s, l \quad (4-13)$$

$$\sum_{s=1}^{J!} W_s = 1 \quad (4-14)$$

Where

$$t_{\mu(s, ll-1), \mu(s, ll)} = t_{0 ll} \text{ when } ll=1.$$

This model is quite straightforward. The objective function includes the total weighted flow times and the cost associated with weighted compression processing times. As the extension of job processing times cannot improve the planning goal, it has been excluded from consideration. Constraint (4-12) determines the amount of job compression while the maximum allowable job compression is given in constraint (4-13). Finally, constraint (4-14) specifies that only one sequence can be selected as the final solution.

It is obvious that if the objective is solely to minimize total flow times for any given sequence, the minimum cost would occur if all jobs are processed with their minimum processing times. Therefore, the problem is to find a sequence that minimizes the weighted flow times and sequence-dependent setup times. Accordingly, Model 4-2 can be modified as follows to reflect these changes in the objective function:

Model 4-2-1

$$\text{Min}_{1 \leq s \leq J!} \{G(s)\} = \text{Min} \left\{ \sum_{s=1}^{J!} W_s \left[\sum_{l=1}^J u_{\mu(s,l)} \sum_{ll=1}^l \left(t_{\mu(s,ll-1)\mu(s,ll)} + P_{\mu(s,ll)}^L \right) \right] \right\} \quad (4-15)$$

Subject to:

$$\sum_{s=1}^{J!} W_s = 1 \quad (4-16)$$

Because of the non-linearity of the objective functions and the large number of 0-1 variables, W_s , Model 4-1 and Model 4-2 cannot be easily solved even for small size problems. Thus, an efficient solution procedure is required to solve these problems. In the following sections, a tabu search algorithm is first developed to provide quick and good solutions for the JIT scheduling problem (i.e., Model 4-1). Later, the algorithm will be modified to solve Model 4-2.

4.4 Tabu search algorithm for JIT sequencing

For the JIT sequencing problem, a solution is essentially a permutation of J jobs. There is a total of $J!$ possible sequences or permutations of the J jobs. A set of neighbouring solutions $N(s)$ of sequence s can be constructed by pairwise exchanging the jobs in s . A move is then made from s'' , the best job sequence of the immediate previous neighbourhood, to s^* , provided that s^* is not in the current tabu list. This procedure is repeated until a specified termination criterion is reached. The termination criterion can be either a maximum allowed search period or a maximum

allowed number of moves. The proposed tabu search algorithm for the JIT sequencing problem is presented below.

Algorithm 4-1

Step 1 Initialization

- (a) Set $T_list = \{\emptyset\}$, $S_best = \{\emptyset\}$, $M_ctr = 0$, $N_ctr = 0$, and $Cbest = M_big$
- (b) Read T_size , $Mmax$, $Nmax$, and $P_j^L, P_j^N, P_j^U, \alpha_j, \beta_j, a_j, b_j, d_j$ for $j = 1, \dots, J$.
- (c) Construct a starting sequence s^{**} and compute $G(s^{**})$

Step 2 Search

Step 2.1 Generate and evaluate neighbouring solutions

WHILE $M_ctr < Mmax$ DO

set $G(s^*) = M_big$;

DO $jj = 1$ to $J-1$

DO $kk = jj+1$ to J

generate a new neighbour s for s^{**} by interchanging the part in position jj with the part in position kk .

IF $s \in T_list$, discard s , and continue;

ELSE compute $G(s)$

IF $G(s) < G(s^*)$, set $G(s^*) \leftarrow G(s)$, and $s^* \leftarrow s$;

ELSE discard s , and continue;

ENDIF

ENDIF

ENDDO

ENDDO

Step 2.1 Move

Set $G(s^{**}) \leftarrow G(s^*)$, $s^{**} \leftarrow s^*$ and update T_list ;

IF $G(s^{**}) < C_{best}$, set $C_{best} \leftarrow G(s^{**})$, $S_{best} \leftarrow s^{**}$, $M_ctr \leftarrow M_ctr$
 $+1$, and $N_ctr \leftarrow N_ctr + 1$;

ELSE set $M_ctr \leftarrow M_ctr + 1$, $N_ctr \leftarrow N_ctr + 1$;

ENDIF

Step 2.2 Diversification

Diversify the search using one or more of the following strategies:

- (1) Divide the maximum number of moves, M_{max} , into equal sized phases of N_{max} moves and:

IF $N_ctr \geq N_{max}$, set $T_list = \{\emptyset\}$, $N_ctr \leftarrow 0$, and diversify the search path, using one of the following scenarios:

- (a) Restart from S_{best} , the best sequence found so far.
(b) Restart from a randomly selected sequence.

- (2) Change the search parameters and repeat the search.

ELSE continue;

ENDIF

ENDWHILE

Stop

To evaluate $N(s)$ at each iteration, it is necessary to calculate the minimum cost $G(s)$ for each sequence in $N(s)$ which involves solving a set of linear programming models. The linear programming model and its solution procedure are described in the following.

4.4.1 Single sequence cost minimization model

When the problem involves a fixed sequence say s , Model 4-1 reduces to the following linear programming problem:

Model 4-3

$$\text{Min} \sum_{l=1}^J (a_{\mu(s,l)} T_{\mu(s,l)} + b_{\mu(s,l)} E_{\mu(s,l)} + \alpha_{\mu(s,l)} X_{\mu(s,l)} + \beta_{\mu(s,l)} Y_{\mu(s,l)}) \quad (4-17)$$

Subject to:

$$P_{\mu(s,l)} + X_{\mu(s,l)} \geq P_{\mu(s,l)}^N \quad \forall l \quad (4-18)$$

$$P_{\mu(s,l)} - Y_{\mu(s,l)} \leq P_{\mu(s,l)}^N \quad \forall l \quad (4-19)$$

$$P_{\mu(s,l)}^L \leq P_{\mu(s,l)} \leq P_{\mu(s,l)}^U \quad \forall l \quad (4-20)$$

$$\sum_{ll=1}^l (t_{\mu(s,ll-1),\mu(s,ll)} + P_{\mu(s,ll)}) - d_{\mu(s,l)} \leq T_{\mu(s,l)} \quad \forall l \quad (4-21)$$

$$d_{\mu(s,l)} - \sum_{ll=1}^l (t_{\mu(s,ll-1),\mu(s,ll)} + P_{\mu(s,ll)}) \leq E_{\mu(s,l)} \quad \forall l \quad (4-22)$$

$$E_{\mu(s,l)} \geq 0 \quad \forall l \quad (4-23)$$

$$T_{\mu(s,l)} \geq 0 \quad \forall l \quad (4-24)$$

This model is solvable using commercial software such as LINDO. It takes about 2 seconds to solve Model 4-3 for a 30-job problem using LINDO. However, since Model 4-3 is nested in Algorithm 4-1 and each move involves solving 435 such problems for a 30-job problem and $J(J-1)/2$ such problems in general, it is practically not acceptable to use commercial software to solve Model 4-3. To reduce the computational burden, an efficient solution algorithm is developed which can solve Model 4-3 in much shorter computational time.

4.4.2 Alternative solution procedure for Model 4-3

Here, an efficient heuristic algorithm for solving Model 4-3 is developed. This algorithm starts with a permutation in which all jobs are assumed to be processed with minimum allowable processing times. Therefore, the further compression of processing time of a job is impossible and only extension of processing time can be made. As a result, the *saving rate*, i.e., the total potential saving in iteration n due to one unit time increment for the job in position ll of sequence s , can be given by the following expression:

$$F_{\mu(s,ll)}^n = B_{\mu(s,ll)} - A_{\mu(s,ll)} + H_{\mu(s,ll)} \quad (4-25)$$

Where

$$B_{\mu(s,ll)} = \sum_{l=ll}^J b_l \quad \{l \mid E_l^n > 0\} \quad (4-26)$$

$$A_{\mu(s,ll)} = \sum_{l=ll}^J a_l \quad \{l \mid T_l^n \geq 0\} \quad (4-27)$$

and

$$H_{\mu(s,ll)} = \begin{cases} \alpha_{\mu(s,ll)} & \text{if } \rho_{\mu(s,ll)}^n > P_{\mu(s,ll)}^U - P_{\mu(s,ll)}^N \\ -\beta_{\mu(s,ll)} & \text{if } \rho_{\mu(s,ll)}^n \leq P_{\mu(s,ll)}^U - P_{\mu(s,ll)}^N \end{cases} \quad (4-28)$$

In Equation (4-25), the first two terms respectively take care of the earliness and tardiness costs caused by one unit time increment. $B_{\mu(s,ll)}$ is the sum of the reduced earliness penalty on all the early jobs and $A_{\mu(s,ll)}$ is the total increased tardiness penalty due to one unit time increment. The sign of $B_{\mu(s,ll)}$ is positive since any time increase for the early jobs will reduce the earliness penalty which means a positive saving. In contrast, for the tardy jobs, any time increment will increase the tardiness penalty and therefore the second term has a negative sign. The last term in Equation (4-25) reflects the effect of one unit time increment on the compression and extension costs. As shown in Equation (4-28), if the increasable processing time is greater than the difference of $P_{\mu(s,ll)}^U$ and $P_{\mu(s,ll)}^N$, i.e., the current processing time of job $\mu(s,ll)$ is less than $P_{\mu(s,ll)}^N$, one unit of time increase will release the compression and thus the effect is a saving of $\alpha_{\mu(s,ll)}$. Otherwise, if the amount of increasable time is less than the difference of $P_{\mu(s,ll)}^U$ and $P_{\mu(s,ll)}^N$, namely, the current processing time of job $\mu(s,ll)$ is greater than $P_{\mu(s,ll)}^N$, further slowing down the process by one time unit will reduce the saving by $\beta_{\mu(s,ll)}$.

The above saving rate can be used as a measure of the potential improvement achievable by changing the processing time of each job in the permutation. The heuristic procedure for solving Model 4-3 is based on the saving rate measure and the algorithm

is summarized below.

Algorithm 4-2

- Step 1* Set $n \leftarrow 1$ and compute the tardiness or earliness of each job assuming all jobs are processed with the minimum possible time.
- Step 2* Compute the saving rate, $F_{\mu(s,l)}^n$, for all jobs with $\rho_{\mu(s,l)}^n > 0$. If $F_{\mu(s,l)}^n \leq 0$ or $\rho_{\mu(s,l)}^n = 0$ for all jobs, further cost reduction cannot be achieved and the computation is terminated. Otherwise, go to next step.
- Step 3* Find $F_{\mu(s,q)}^n = \max\{F_{\mu(s,l)}^n \mid \forall l\}$. Increase processing time of $\mu(s,q)$ by

$$\Delta P_{\mu(s,q)}^n = \min \{ \delta_1, \delta_2 \} \quad (4-29)$$

Where

$$\delta_1 = \min \{ E_{\mu(s,l)}^n > 0 \mid l \geq q \} \quad (4-30)$$

and

$$\delta_2 = \begin{cases} \rho_{\mu(s,q)}^n + P_{\mu(s,q)}^N - P_{\mu(s,q)}^U & \text{if } \rho_{\mu(s,q)}^n > P_{\mu(s,q)}^U - P_{\mu(s,q)}^N \\ \rho_{\mu(s,q)}^n & \text{if } \rho_{\mu(s,q)}^n \leq P_{\mu(s,q)}^U - P_{\mu(s,q)}^N \end{cases} \quad (4-31)$$

- Step 4* Set $n \leftarrow n + 1$. Update $\rho_{\mu(s,q)}^n$, $E_{\mu(s,l)}^n$, $T_{\mu(s,l)}^n$, for $l \geq q$, and go to step 2.

Step 3 in the above algorithm is based on the logic that the time increase for the job with the highest saving rate is likely to be most profitable. Thus the time increase

should be made until such high profitability has been exhausted. The range of such high profitable time is established in Step 3 and proven below.

Theorem 1 (Selection of time increment)

For a job $\mu(s,q)$ with increasable processing time in iteration n , i.e., $\rho^n_{\mu(s,q)} > 0$,

- (a) if $F^n_{\mu(s,q)}$ is positive, it will remain constant in the range of $[P^N_{\mu(s,q)}, P^N_{\mu(s,q)} + \Delta P^n_{\mu(s,q)}]$ and the saving will be a monotonic linear increasing function with slope $F^n_{\mu(s,q)}$ in that range.
- (b) If processing time is further increased above the range of $[P^N_{\mu(s,q)}, P^N_{\mu(s,q)} + \Delta P^n_{\mu(s,q)}]$, $F^n_{\mu(s,q)}$ will decrease though it may remain positive.

Proof of Theorem 1

- (a) Referring to Equations (4-25) to (4-31), increasing processing time in the range of $[P^N_{\mu(s,q)}, P^N_{\mu(s,q)} + \Delta P^n_{\mu(s,q)}]$ will not change the values of $A_{\mu(s,q)}$, $B_{\mu(s,q)}$, and $G_{\mu(s,q)}$. Therefore, the saving will be a monotonic linear increasing function in the range with slope $F^n_{\mu(s,q)}$.
- (b) Referring to Step 3 of algorithm 4-2, the magnitude of $F^n_{\mu(s,q)}$ is determined by one of the following cases:

Case 1

If $\Delta P^n_{\mu(s,q)} = \delta_l$, i.e., $\Delta P^n_{\mu(s,q)} = E^n_{\mu(s,q')} = \min \{E^n_{\mu(s,l)} > 0 \mid l \geq q\}$ (see Equations (4-29) and (4-30)), any increment of processing time of job $\mu(s,q)$ greater than

$E_{\mu(s,q)}^n$ will make the job in position q' tardy and therefore reduce $F_{\mu(s,q)}^n$ by at least $a_{\mu(s,q')} + b_{\mu(s,q')}$. This is because the original possible saving $b_{\mu(s,q')}$ has been lost and a new penalty cost, $a_{\mu(s,q')}$ is imposed due to this job's status change from early to tardy. In addition, the status change of the job in position q' may also cause status changes of other succeeding jobs in the sequence and thus additional reduction in $F_{\mu(s,q)}$ is possible.

Case 2

If $\Delta P_{\mu(s,q)}^n = \delta_2$ (see Equations (4-29) to (4-31)), there exist the following two possibilities:

- (i) $\rho_{\mu(s,q)}^n > P_{\mu(s,q)}^U - P_{\mu(s,q)}^N$. If so, any increase of processing time of job $\mu(s,q)$ more than δ_2 will cause a transition from saving due to the released compression to penalty caused by extension and thus reduce $F_{\mu(s,q)}$ by at least $\alpha_{\mu(s,q)} + \beta_{\mu(s,q)}$. Since δ_2 is less than δ_1 , increasing the processing time of job $\mu(s,q)$ by δ_2 will not affect the earliness or tardiness of other jobs. However, if further increasing the processing time of job $\mu(s,q)$, some succeeding jobs in the sequence may become tardy and thus additional reduction in $F_{\mu(s,q)}$ may occur as established in case 1.
- (ii) $\rho_{\mu(s,q)}^n \leq P_{\mu(s,q)}^U - P_{\mu(s,q)}^N$. In this case, since $\Delta P_{\mu(s,q)}^n + P_{\mu(s,q)} = \delta_2 + P_{\mu(s,q)} = P_{\mu(s,q)}^U$ (see Equation (4-29)), i.e., the maximum allowed processing time has been reached, it is therefore impossible to increase processing time of job $\mu(s,q)$ by more than $\Delta P_{\mu(s,q)}^n$.

Theorem 1 is thus proved.

As proven in Theorem 1, the saving rate level of the current job becomes lower beyond certain range. Thereafter, even if the saving rate is still positive, we will not further increase the processing time in the current iteration. The reason is that some other jobs may yield a higher saving rate than that of the current job. Therefore, a new iteration using a newly selected highest saving rate will be carried out. The termination criterion is based on the following theorem.

Theorem 2 (Termination criterion)

In Algorithm 4-2, further improvement can not be achieved in the current and future iterations if the maximum saving rate, $F_{\mu(s,q)}^n = \max \{F_{\mu(s,l)}^n \mid l=1, \dots, J\} \leq 0$.

Proof of Theorem 2

For a job $\mu(s,q)$, if its increasable processing time $\rho_{\mu(s,q)}^n > 0$ and its processing time is increased by $\Delta P_{\mu(s,q)}^n = \min \{\delta_1, \delta_2\}$, then for any other job $\mu(s,l)$, we have the followings:

$$\text{Case 1} \quad F_{\mu(s,l)}^{n+1} = F_{\mu(s,l)}^n \text{ if } \Delta P_{\mu(s,q)}^n = \delta_2, \text{ or } \Delta P_{\mu(s,q)}^n = \delta_1 \quad (\delta_1 = E_{\mu(s,q')}^n = \min \{E_{\mu(s,l)}^n > 0 \mid l \geq q\} \text{ and } l > q').$$

$$\text{Case 2} \quad F_{\mu(s,l)}^{n+1} = F_{\mu(s,l)}^n - (b_{\mu(s,q)} + a_{\mu(s,q)}) \text{ if } \Delta P_{\mu(s,q)}^n = \delta_1 \quad (\delta_1 = E_{\mu(s,q')}^n = \min \{E_{\mu(s,l)}^n > 0 \mid l \geq q\} \text{ and } l \leq q').$$

It follows that if the processing time of job $\mu(s,q)$ with maximum saving rate $F_{\mu(s,q)}^n$ in

iteration n is increased by $\Delta P_{\mu(s,q)}^n$, and job $\mu(s,q')$ results in maximum saving rate $F_{\mu(s,q')}^{n+1}$ in iteration $n+1$, then $F_{\mu(s,q)}^n \geq F_{\mu(s,q')}^{n+1}$ always holds. This leads to:

$$F_{\mu(s,q)}^n \geq F_{\mu(s,q')}^m \quad (m > n) \quad (A1)$$

Now, suppose the solution obtained using algorithm 4-2 can be further improved in at least one future iteration, say, iteration m ($m > n$). Then there must exist an $F_{\mu(s,q')}^m = \max \{F_{\mu(s,l)}^n | l=1, \dots, J\} > 0$, which means $F_{\mu(s,q')}^m > F_{\mu(s,q)}^n$. This, however, contradicts (A1) and thus completes the proof of Theorem 2.

Algorithm 4-2 has been coded in C and verified using 1000 randomly generated problems of different sizes on a 486 PC. The results obtained using the algorithm are compared with the optimal solutions obtained using LINDO package. As summarized in table 4-1, Algorithm 4-2 provided exact solutions for all the 1000 problems while the computational times are considerably reduced.

Table 4-1 Comparison between Algorithm 4-2 and LINDO

Problem size No. of jobs	No. of problems solved	Accuracy of solutions	Ave. computational time (sec.)	
			LINDO	Algorithm 4-2
10	200	100%	0.62	0.012
20	200	100%	1.10	0.018
30	200	100%	1.83	0.031
40	200	100%	2.55	0.049
50	200	100%	3.46	0.062

Now, Algorithm 4-2 can be nested in Algorithm 4-1 for solving Model 4-1. The combined search algorithm has also been coded in C.

4.5 Tabu search algorithm for minimization of weighted flow time

The objective of this section is to demonstrate that the proposed tabu search scheme in the previous section can be modified to solve the problem of minimizing weighted flow time with controllable processing times and sequence dependent setups. The main modification needed here is in the cost function and the procedure required to calculate it. The process for calculating the cost of each sequence is established below and is based on the terms and definitions stated in Section 4.4.

Similar to Algorithm 4-1, to evaluate the neighbourhood at each iteration, it is necessary to obtain the minimum cost for each member of $N(s)$. This again requires solving a linear programming model for each sequence. The linear programming model is given below:

Model 4-4

$$\text{Min } \{G(s)\} = \text{Min } \left\{ \sum_{l=1}^J u_{\mu(s,l)} \sum_{ll=1}^l (t_{\mu(s,ll-1)\mu(s,ll)} + P_{\mu(s,ll)}) + \sum_{l=1}^J \alpha_{\mu(s,l)} X_{\mu(s,l)} \right\} \quad (4-32)$$

Subject to:

$$P_{\mu(s,l)} + X_{\mu(s,l)} \geq P_{\mu(s,l)}^N \quad \forall l \quad (4-33)$$

$$P_{\mu(s,l)} \geq P_{\mu(s,l)}^L \quad \forall l \quad (4-34)$$

Now the procedure for calculating minimum cost $G(s)$ for each sequence in $N(s)$ can be established as follows.

As reported by Vickson (1980), in any given sequence with minimum processing cost, a job is either processed in normal processing time or compressed to its minimum allowable processing time. This property, allows us to limit our focus on finding the jobs to be compressed. To start with, the *saving rate* for the job in position ll of sequence s is defined as:

$$F_{\mu(s,ll)} = \sum_{l=ll}^J u_{\mu(s,l)} - \alpha_{\mu(s,ll)} \quad (4-35)$$

In this case, the first term of the above expression presents the saving achieved by one unit reduction of processing time of the job in position ll . The cost associated with such a process time reduction is reflected by the second term of Equation (4-35) and its sign. It is obvious that if there is no penalty to the reduced processing times, the minimum cost would occur provided that all jobs are processed with their minimum processing times. This saving rate can be used to find the jobs whose processing time compressions have a positive effect on the overall saving. As a result, the processing times of such jobs can be reduced to their minimum allowable times. Hence, Model 4-4 can be solved in at most J iterations using the following algorithm.

Algorithm 4-3

DO $l = 1$ to J

compute $F_{\mu(s,l)}$

```

IF       $F_{\mu(s,l)} > 0$ , set  $P_{\mu(s,l)} = P_{\mu(s,l)}^L$ 
ELSE   set  $P_{\mu(s,l)} = P_{\mu(s,l)}^N$ 
ENDIF

ENDDO

```

The above algorithm can be used to determine the minimum cost for each sequence in $N(s)$. This allows the tabu search to move towards an optimum or near optimum solution among all possible sequences.

4.6 Computational results and discussions

In the following sections, a 30-job problem is solved for Models 4-1 and 4-2. The effects of some search parameters are also discussed. All the computations are run on the same 486 PC mentioned earlier.

4.6.1 Example problem for Model 4-1

In this section, the proposed algorithm is applied to sequence 30 jobs on a machining centre based on the objectives stated in Model 4-1. In the stated problem, each job has its own due date. The deviation from the due date will result in earliness or tardiness penalty. The processing time of each job is controllable within a certain range. The information about the processing times, due dates, compression and extension costs, as well as earliness and tardiness penalty costs is listed in Table 4-2. The sequence-dependent setup times are the same as those given in Table 3-2.

Table 4-2 Processing times, due dates, and cost data for the example problem

j	P_j^N	P_j^U	P_j^L	d_j	a_j	b_j	α_j	β_j
	(min)	(min)	(min)	(min)	(\$/min)	(\$/min)	(\$/min)	(\$/min)
1	3	4	3	72	5.0	5.0	4.0	1.0
2	4	5	3	360	4.0	3.0	5.5	1.6
3	8	14	8	300	4.0	3.0	4.5	1.2
4	8	20	8	30	1.5	1.5	1.8	1.4
5	10	10	10	270	6.5	1.5	4.0	1.0
6	10	15	8	330	4.0	4.0	2.6	0.3
7	10	18	10	432	6.0	3.0	1.5	2.5
8	10	15	8	390	4.5	2.0	2.0	2.0
9	10	12	5	180	5.0	3.0	2.7	1.5
10	11	15	9	210	4.5	1.5	1.7	1.4
11	12	12	9	228	1.5	1.0	3.5	2.5
12	12	12	10	420	2.0	0.5	0.9	0.3
13	14	20	10	210	4.0	3.0	0.5	0.5
14	14	14	5	465	3.5	0.5	3.6	2.3
15	15	15	15	115	1.0	1.0	1.0	1.3
16	15	20	12	150	8.5	0.5	1.5	0.5
17	15	19	7	210	2.5	1.0	0.6	0.3
18	15	25	11	282	3.0	0.5	3.0	0.2
19	15	25	8	480	8.0	1.0	1.0	0.5
20	18	20	14	405	4.0	1.5	0.6	0.3
21	20	25	15	30	5.5	2.0	2.3	1.3
22	20	28	14	120	6.0	2.0	1.0	0.6
23	20	30	12	240	5.5	2.0	3.5	1.5
24	20	30	11	480	4.5	1.0	1.8	1.0
25	24	25	12	360	7.0	4.0	1.8	0.4
26	25	40	14	345	5.0	2.0	2.2	0.5
27	28	35	15	60	5.0	1.0	2.0	0.5
28	30	50	20	90	5.5	1.5	1.6	0.2
29	30	37	18	240	3.5	2.5	1.5	2.0
30	35	40	22	144	5.0	1.5	0.8	0.5

Note: The total processing cost=\$5327 if jobs are sequenced in the order given above.

To investigate the effects of diversification strategies, tabu list sizes, and starting sequences, the computations are carried out using 20 randomly generated starting sequences based on the following search scheme:

Diversification strategy	Phase length <i>Nmax</i>	Tabu list size <i>T_size</i>
Restart from <i>S_best</i>	20	5,10
	50	5,10,20,30
Restart from a random sequence	20	5,10
	50	5,10,20,30
No diversification	100	5,10,20,30

As presented above, three diversification strategies are used. The search process is terminated after 100 moves for all the above scenarios. With the first strategy, we clear the tabu list after each search phase, i.e., a pre-specified number of moves, *Nmax*, and resume the search from the best solution found so far. The similar procedure is applied to the second diversification strategy. The only difference is that the restarting point is selected randomly. For the third strategy, the entire search process is performed in a single phase.

Clearly, the tabu list sizes should be smaller than the associated search phase since otherwise all moves will become tabu. For example, the longest tabu list for a 20-move phase is 10, and the longest tabu list is specified as 30 for a 50-move phase, etc.

The solution obtained, using the best combination of the diversification strategy, tabu list size and starting sequence is tabulated in Table 4-3. Both the sequencing and

processing time decisions can be simultaneously made using the information in Table 4-3.

Table 4-3 Solution to the Model 4-1 example problem

Seq. ^(a)	dP (min)	TE (min)	P* (min)	FT (min)
21	0	6	20	24
4	0	-4	8	34
27	0	-2	28	62
1	1	5	4	67
28	-8	0	22	90
22	2	6	22	114
30	-13	8	22	136
16	-3	0	12	150
15	0	-52	15	167
9	0	2	10	178
17	-8	22	7	188
10	-2	10	9	200
13	-4	0	10	210
29	-12	10	18	230
23	-8	-3	12	243
11	0	-28	12	256
5	0	1	10	269
18	0	-4	15	286
3	6	0	14	300
12	0	105	12	315
6	3	0	10	330
26	-11	0	14	345
25	-10	0	14	360
2	0	-5	4	365
8	5	7	15	383
20	2	0	20	405
14	0	45	14	420
7	0	0	10	432
24	8	17	28	463
19	0	0	15	480

Note: ^(a) : Total cost = \$504 using this sequence
dP : Deviation from normal processing time
TE : Tardiness or earliness of jobs
P* : Final processing time
FT : Flow time

4.6.2 Example problem for Model 4-2

The problem of minimization the total weighted flow time with controllable processing time and sequence dependent setups was formulated in Section 4.3.2. The following example illustrates the application of tabu search algorithm to solve this problem. It should be mentioned that in the proposed algorithm the cost associated with each sequence, $G(s)$, is based on the objective function and constraints of Model 4-4 which is calculated using Algorithm 4-3. The job processing times and their compression costs are the same as those in the previous example and listed in Table 4-2. The sequence dependent setup times and the unit flow cost for each job are also given in Tables 3-2 and 3-8 respectively.

For this example, different tabu list sizes and initial solutions were tested and for each trial, the computation was carried out in a single phase with a 15 minute search time. Table 4-4 illustrates the results for the best tabu list size and initial solution.

Table 4-4 Solution for the Model 4-2 example problem

Seq. ^(a)	dP (min)	P* (min)	FT (min)	FC (\$)
14	- 9	5	6	6
19	- 7	8	15	15
1	0	3	19	19
11	- 3	9	28	31
23	- 8	12	41	49
2	- 1	3	45	18
21	- 5	15	60	54
8	- 2	8	69	69
3	0	8	78	55
12	- 2	10	90	72
24	- 9	11	103	93
10	- 2	9	113	56
5	0	10	124	74
29	-12	18	144	144
28	-10	20	165	198
13	- 4	10	175	88
26	-11	14	190	95
18	- 4	11	201	121
16	- 3	12	214	107
4	0	8	223	67
17	- 8	7	230	92
20	- 4	14	245	123
25	-12	12	257	129
7	0	10	268	107
30	-13	22	291	146
6	0	10	301	60
9	0	10	313	63
15	0	15	330	66
27	0	28	359	108
22	0	20	382	76

Notes ^(a) : Total cost = \$2654 using this sequence
dP : Deviation from normal processing time
P* : Final processing time
FT : Flow time
FC : Flow cost

The total processing cost for the above sequence is \$2654. This includes \$2402 job flow cost and a total of \$252 processing time compression cost. As shown in Table 4-4, despite added compression costs, the processing times of most jobs have been reduced to their minimum allowable times. However, this is justified by the overall savings achieved by reducing the job flow time. Consequently, the makespan (382 minutes, the time required to process all jobs) for this sequence is significantly lower than that obtained in the previous example, i.e., 480 minutes.

As demonstrated by the above examples, the job permutations vary considerably depending upon the performance measure used to evaluate each sequence. The models developed in this chapter provide a guideline for managers and the planning decisions can be made in accordance with the system requirements and objectives.

4.6.3 Discussions

The discussions provided below are based on the results of the example problem for JIT sequencing (example problem for Model 4-1). Obviously, similar discussions may be made for the results obtained for Model 4-2.

To gain further insight into JIT sequencing problem and the proposed tabu search approach, the computation results obtained, using the three best starting sequences, are summarized in Table 4-5, though all different starting sequences lead to similar solutions. The results in Table 4-5 are based on the best tabu list size. With reference to Tables 4-3 and 4-5, discussions are made as follows:

a) JIT sequencing

Table 4-3 shows that JIT sequencing has been achieved for 10 jobs, i.e., jobs 28, 16, 13, 3, 6, 26, 25, 20, 7, and 19. It is also shown that 24 out of the 30 jobs have a deviation from their due dates less than or equal to 10 minutes.

b) Cost reduction

Table 4-5 clearly indicates that the total cost associated with time compression, extension, and the penalty to the deviation from the due dates has been substantially reduced for all the search scenarios. For instance, with the first diversification strategy and the first random starting sequence, the total cost has been reduced from \$5327 to \$504. Even if diversification is not applied, the total cost can be reduced to \$548 using the same starting sequence.

Table 4-5 Final costs for different starting sequences and diversification strategies

Diversification Strategy												
			Restart from S_{best}				Restart from a random sequence				No diversification	
Phase size			50				20				100	
Starting Seq.	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
$TC_s(\$)$	5327	7678	6333	5327	7678	6333	5327	7678	6333	5327	7678	6333
$TC_r(\$)$	504	504	606	504	504	606	548	632	768	709	879	1107
γ (%)	1057	1523	1045	1057	1523	1045	972	1214	825	751	873	572
			5327 7678 6333				5327 7678 6333				5327 7678 6333	
			504 504 606				548 632 768				548 632 606	
			1057 1523 1045				972 1214 825				972 1214 1045	

Note: TC_s : Total initial cost TC_r : Total final cost γ : Improvement over initial sequence

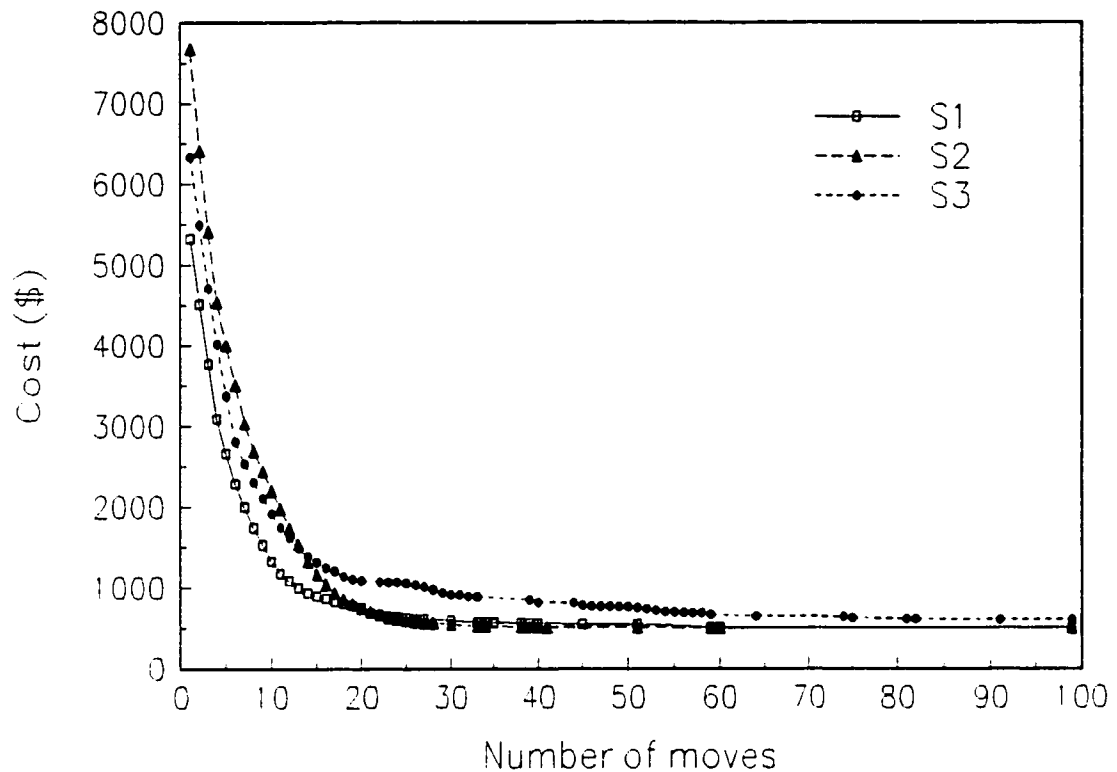


Figure 4-1 Convergence curves for the Model 4-1 example problem with different starting sequences

c) Solution convergence

The convergence curves for the first diversification strategy (Table 4-5) are plotted in Figure 4-1. These curves reveal that most of the cost reduction is obtained in the first 25 moves (almost 6.0 minutes of search time). No additional improvement is observed after 60 moves (approximately 16 minutes). The computations have been extended to 24 hours and the same was observed.

d) Effect of search diversification

The computational results (Table 4-5) indicate that search path diversification has some impact on the performance of the algorithm. The first diversification strategy, i.e., restarting from the best solution, S_{best} , seems to be superior to the other two strategies. A possible reason is that the search barriers are reduced since the tabu list is cleared after each search phase and thus the search can proceed more easily towards a better solution. The second diversification strategy, i.e., resuming the search from a randomly selected sequence after each phase, appears to be the worst of the three. This is even more evident when the phase size becomes smaller, e.g., 20 moves. The reason could be that part of the previous search effort is more frequently discarded if the phase size is smaller and thus the chance of resuming from a good solution is lower. This phenomenon is illustrated in Figure 4-2, which corresponds to the starting sequence S1. It can be seen in Figure 4-2 that the random diversification results in a piece-wise convergence curve and every restarting point is much worse than the discarded solution.

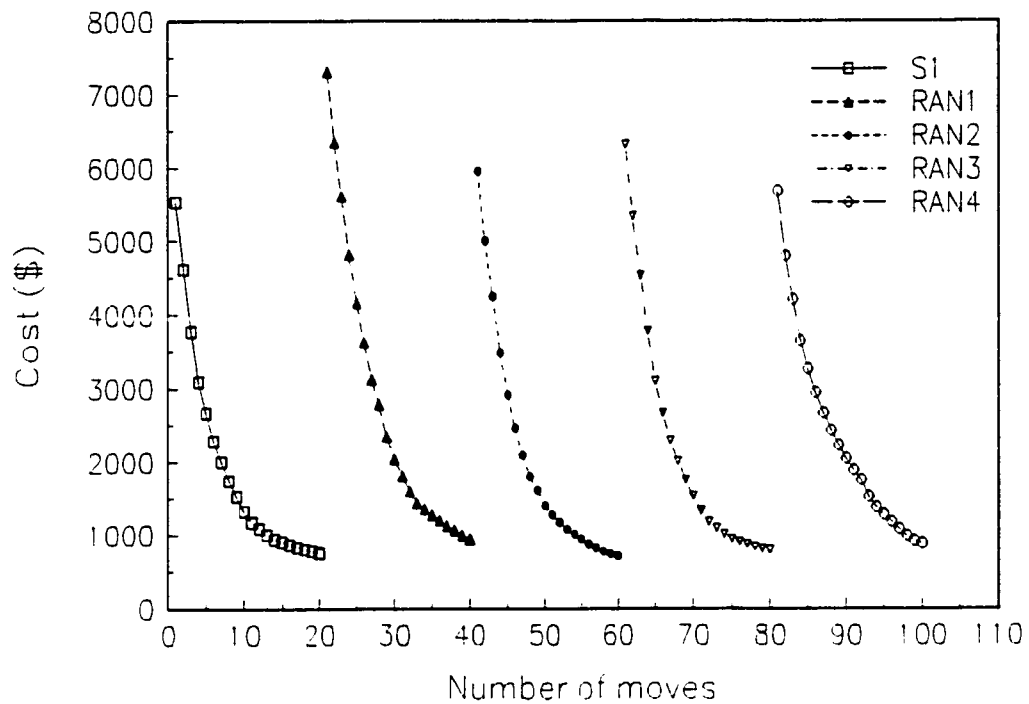


Figure 4-2 Piece-wise convergence curve when search restarts from a random sequence

e) Effect of tabu list size

The computational results based on different tabu list sizes for the three best starting sequences S1, S2, and S3 are summarized in Table 4-6. All the computations were conducted without path diversification. Table 4-6 shows that a longer tabu list usually results in slightly better solutions. This is in agreement with the notion of adopting tabu list.

Table 4-6 The effect of tabu list size on search performance

Starting Sequence	Final cost (\$) using tabu list size			
	5	10	20	30
S1	608	548	548	548
S2	641	641	632	632
S3	667	606	606	606

4.7 Refinement of the tabu search algorithm

4.7.1 Neighbourhood generation and move selection

Conventionally, a move is made after evaluating the entire neighbourhood. However, when the problem size is large, the required computational time could be unacceptably long. To overcome this difficulty, partial search schemes have been introduced recently. For instance, Crauwels *et. al* (1996) used the first non-tabu move that results in a superior solution. William (1994) and Della Croce (1995) recommended the selection of the best non-tabu move from a fixed number of randomly or sequentially generated solutions.

A more efficient and sophisticated strategy is probably to evaluate part of neighbours that most likely contain the best solution. Along this line, Andeso-Diaz (1992) proposes the use of a restricted neighbourhood, RN, to improve the computational performance in solving a weighted tardiness problem. This approach is based on the observation that the convergence process generally presents an exponential behaviour. It is further observed that, in the first few iterations, the largest cost reduction (objective

improvement) is often obtained by exchanging the jobs located far apart in the current sequence. Thus, larger transposition ranges tend to be used in the first few iterations. As the search progresses, the transposition ranges become smaller. In view of this, Andeso-Diaz (1992) suggests that the entire neighbourhood be evaluated only in the first few iterations whereas the transposition range in the later iterations be consecutively reduced to a pre-specified number of neighbouring jobs.

Nevertheless, in the above strategies the effect of the starting point on the search performance is overlooked. In addition, the search process is not guided by any objective value. In Andeso-Diaz's method, for instance, all the parameters used in determining the range of transitions are pre-specified and are maintained constant throughout the entire search. This approach cannot fully utilize the information readily available in the search process such as the objective value improvement rate. If the search commences from a near-optimum solution, then pairwise interchanging of the adjacent jobs may be good and fast enough to obtain the final solution, possibly the optimum one. In the same manner, it would be unnecessary to generate those solutions by interchanging neighbouring jobs if, most probably, they are not good candidates for the next move (as it could be the case in the first iterations). It may be beneficial to change the transposition range in accordance with the cost reduction rate during the search. That is, the transposition range will be reduced only if the cost reduction rate falls below an expected value, otherwise the current range of transpositions may be maintained regardless of the number of iterations that have been completed so far. To this end, an adaptive approach will be introduced to determine the range of transpositions in response to the search dynamics.

In this study, Model 4-1 represents a general form of the weighted tardiness problem. The same pattern in cost reduction can be observed as the pairwise job interchanges take place. This means interchanging the jobs located far away in the sequence would result in higher cost difference (disruption level) and vice versa. This property is used to establish the following procedure for determining the transposition range at each iteration. To develop a refined tabu search algorithm, an adaptive neighbourhood generation method is developed in the next section.

4.7.2 Adaptive neighbourhood generation (ANG)

The proposed strategy is similar to the restricted neighbourhood RN in which a limited range of transpositions is permitted for each iteration. However, the allowable transposition range is selected according to the cost reduction rate achieved in the previous iterations. The main idea of the presented ANG method is to maintain a large transposition range when large objective improvement is observed and reduce this range if a small objective improvement or no improvement is observed. In contrast with the other tabu search methods, the ANG features the following two strategies: (a) objective-guided adaptive adjustment of transposition ranges; and (b) double-ended transposition range reduction.

To begin with, the total number of iterations, M_{max} , is specified and divided into equal intervals each with R iterations. For an interval, say r , the upper and lower boundaries of transposition range will take integer values of $RA(r)$ and $ra(r)$ which are respectively defined as follows:

$$RA(r) = \begin{cases} 1 & \text{if } Z \leq 1 \\ Z & \text{if } 1 < Z < J-1 \\ J-1 & \text{if } Z \geq J-1 \end{cases} \quad (4-36)$$

$$ra(r) = \begin{cases} 1 & \text{if } z \leq 1 \\ z & \text{if } 1 < z < RA(r) \\ RA(r) & \text{if } z \geq RA(r) \end{cases} \quad (4-37)$$

Where

$$Z = \left\lceil RA(r-1) - Y_{RA} J \left(1 - \frac{\lambda_{r-1}}{\lambda_0 e^{-\Psi r}} \right) \right\rceil$$

$$z = \left\lceil ra(r-1) - Y_{ra} J \left(1 - \frac{\lambda_{r-1}}{\lambda_0 e^{-\Psi r}} \right) \right\rceil$$

λ_r maximum improvement obtained between the two successive iterations in the interval r (%)

λ_0 aspired improvement for the first interval (%)

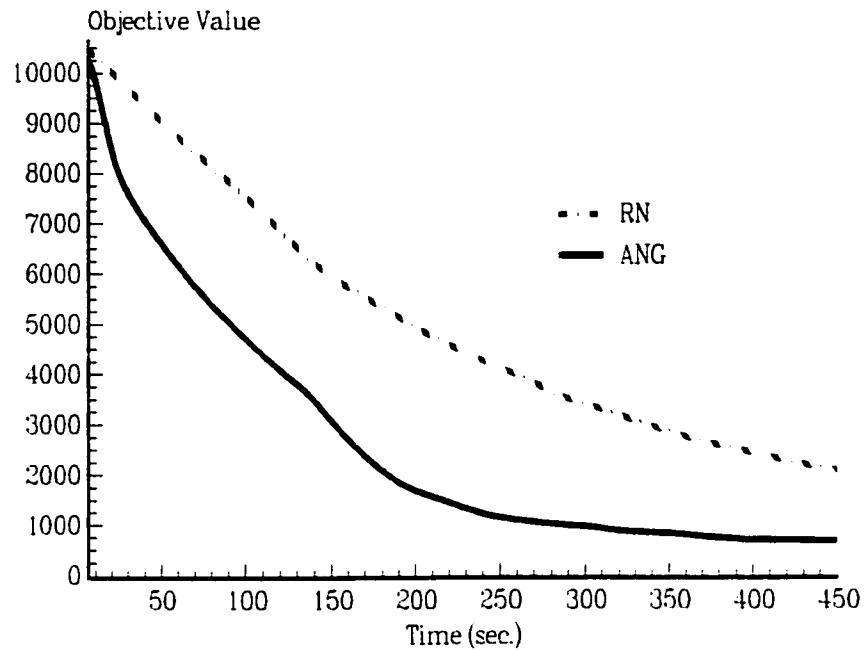
ψ, Ψ damping factors of the aspired improvement

Y_{RA} a 0-1 integer variable, $Y_{RA} = 1$, if $\lambda_{r-1} < \lambda_0 e^{-\Psi r}$; 0, otherwise

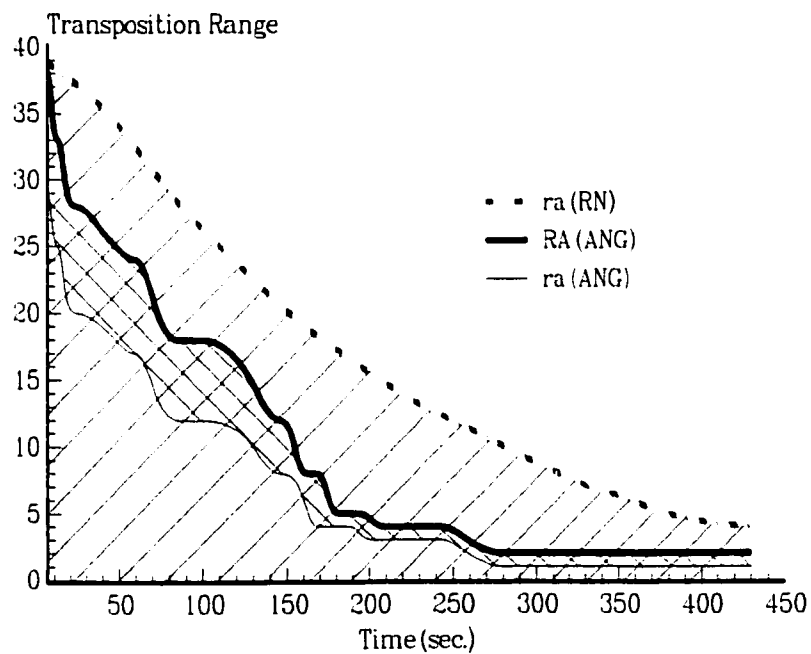
Y_{ra} a 0-1 integer variable, $Y_{ra} = 1$, if $\lambda_{r-1} < \lambda_0 e^{-\psi r}$; 0, otherwise

$RA(r)$ and $ra(r)$ respectively define the upper and lower boundaries of transposition range for interval r and they are adjusted adaptively based on an objective improvement indicator, i.e., the ratio $\lambda_{r,t}/\lambda_0$. The purpose of dividing the entire search process into intervals is to avoid the sudden drop of transposition range due to the possible non-improving moves. As the objective function value in the later iterations usually does not improve as fast as it does in the first few iterations, the aspiration level should be reduced accordingly since otherwise the range will be reduced too rapidly. For this purpose, the damping factors ψ and Ψ are used to restrain the reduction process of the transition range. Using the above expressions, the neighbourhood size can be calculated as $|N_a(s)| = J [J-ra(r)] - 1/2[J-ra(r)] [ra(r)+RA(r)]$ which is in the range of $[J-1, |N(s)|]$ (Note: $|N(s)| = J(J-1)/2$) and thus generally $|N_a(s)| \leq |N(s)|$.

The proposed ANG method may be better exemplified using Figure 4-3. For comparison, the typical patterns of the objective values (Figure 4-3 (a)) and the transition ranges (Figure 4-3 (b)) obtained using both the ANG and RN methods are plotted.



(a)



(b)

Figure 4-3 Transposition range and objective value

In Figure 4-3 (b) the area between the two solid curves is the transposition range corresponding to the ANG search process with RA as the upper boundary (end) and ra the lower boundary (end). The entire area under the dash-dot curve represents the transition range obtained using RN method. As displayed in this figure, the ANG range is significantly narrower than the RN range. It can also be seen that the ANG transposition range in Figure 4-3 (b) is adaptively adjusted in response to the objective improvement rate (Figure 4-3 (a)) while the RN transposition range has no correspondence to this objective value.

4.7.3 The improved algorithm

Having developed the new neighbourhood generation mechanism, we can now modify Algorithm 4-1 to accommodate these changes. The new algorithm is given as follows.

Algorithm 4-4

Step 1 Initialization

- (a) Set $T_list = \{\emptyset\}$, $S_best = \{\emptyset\}$, M_ctr (a counter) = 0, $r=1$, $\lambda_best = -\infty$, and $Cbest = \infty$
- (b) Read T_size , $Mmax$, R , λ_o , ψ , Ψ , $ra(1)$, $RA(1)$, and $P_j^L, P_j^N, P_j^U, \alpha_j, \beta_j, a_j, b_j, d_j$ for $j=1, \dots, J$
- (c) Construct a starting sequence s^{**} and compute $G(s^{**})$

Step 2 Search

Step 2.1 Generate and evaluate neighbouring solutions

WHILE $M_{ctr} < M_{max}$ DO

 set $G(s^*) = \infty$;

 DO $jj = 1$ to $J - ra(r)$

 DO $kk = ra(r)$ to $RA(r)$

 generate a new neighbour s for s^* by interchanging the part in position jj with the part in position kk .

 IF $s \in T_list$, discard s , and continue;

 ELSE compute $G(s)$

 IF $G(s) < G(s^*)$, set $G(s^*) \leftarrow G(s)$, and $s^* \leftarrow s$;

 ELSE discard s , and continue;

 ENDIF

 ENDIF

 ENDDO

ENDDO

Step 2.2 Move

IF $\lambda_best < (G(s^*) / G(s^*)) - 1$, set $\lambda_best \leftarrow (G(s^*) / G(s^*)) - 1$;

ELSE continue;

ENDIF

Set $G(s^*) \leftarrow G(s^*)$, $s^* \leftarrow s^*$ and update T_list ;

IF $G(s^*) < C_{best}$, set $C_{best} \leftarrow G(s^*)$, $S_best \leftarrow s^*$, and $M_{ctr} \leftarrow$

$M_ctr + 1;$

ELSE set $M_ctr \leftarrow M_ctr + 1;$

ENDIF

Step 2.3 Calculate range of transpositions

IF $r < (M_ctr / R)$

$r \leftarrow r+1, \lambda_{r-1} \leftarrow \lambda_best$, calculate new $ra(r)$ and $RA(r)$, $\lambda_best = -\infty$;

ELSE continue;

ENDIF

ENDWHILE

Stop

Step 3 Diversification

Diversify the search using one or more of the following strategies:

- (1) Divide the maximum number of moves, $Mmax$, into equal sized phases of $Nmax$ moves and after each phase:
 - (a) restart from the best solution found so far, S_best .
 - (b) restart from a randomly selected sequence.
- (2) Change the search parameters: T_size , $Nmax$, λ_0 , ψ , Ψ , $ra(1)$, $RA(1)$, R and repeat the search.

Step 3 in the above algorithm is used to enhance the search performance and to find the best combination of search parameters.

4.7.4 Comparison of neighbourhood generation methods and discussions

The developed ANG method has been applied to solve JIT sequencing problem. The results are compared with those given by four other commonly used mechanisms. The computational tests are conducted using the following neighbour generation and move selection schemes:

Pairwise Interchange (PI): This is the most basic approach. The transposition range in this case includes the entire permutation, i.e., from 1 to $J-1$. A move is made to the best non-tabu neighbour in each iteration.

Adjacent Pairwise Interchange (API) (Della Croce, 1995) : With this method, only the adjacent jobs are switched. The size of a neighbourhood is $J-1$. The move is selected in the same manner as in the PI method.

First Improving Neighbour (FIN) (Crauwels and Potts, 1996): In this case, the neighbourhood is generated using the PI method. However, the first non-tabu move which improves the current value of the objective function is accepted and the neighbourhood generation is truncated once such a move is found.

Restricted Neighbourhood (RN): This method was introduced by Adenso-Diaz (1992). For an iteration, r , only those neighbours with a transposition range lower than $ra(r)$ are evaluated and a move is then made to the best non-tabu neighbour in the neighbourhood. The function $ra(r)$ is given by:

$$ra(r) = \left[J (minra/J)^{(r-ret)/estab} \right]_{minra}^{J-1} \quad (4-38)$$

Where *minra* is the minimum range of transpositions, *ret* the number of initial iterations in which the entire neighbourhood is evaluated, and *estab* the iteration number from which any transpositions bigger than *minra* are not allowed. The following values are found to be best for our computations: *minra*=4; *rat*=1; *estab*=15.

Adaptive Neighbourhood (ANG): This is the proposed method in this study and the details have been elaborated in Section 4.7.2. The application of this procedure to solve JIT scheduling problem can also be found in Kolahan and Liang (1998). The following ranges of the settings are used for the tests: $\lambda_o = 10\%$, $\psi = 0.01$ to 0.02 , $\Psi = 0.004$ to 0.008 , $R = 2$ to 4 , $RA(1) = J-1$, and $ra(1) = 1/2J$ to $2/3J$.

A total of 100 problems were randomly generated with 25 problems for each of the four problem sizes: $J = 30, 40, 50$, and 60 jobs. For comparison purpose, the algorithm was run 10 minutes for the 30-job and 40-job problems and 15 minutes for the problems with 50 and 60 jobs. The same starting points were used for all the problems. The phase size, where diversification was applied, was set to 20 moves. It should be noted that only the results obtained, using the best combinations of the search parameters (tabu list size, diversification strategies, etc.) are compared. The average cost reductions over initial solutions are listed in Table 4-7.

Table 4-7 Comparison of different neighbourhood generation methods

Problem set ^a	Number of jobs <i>J</i>	Average cost reduction (%) ^c				
		PI	API	FIN	RN	AN
1	30	90.6 ^b	77.8	64.5	91.4	93.1
2	40	69.5	59.0	42.3	84.7	92.0
3	50	51.4	41.2	29.6	62.3	83.2
4	60	25.3	16.3	19.2	30.2	62.9

a) Each set contains 25 randomly generated problems

b) Average of 25 randomly generated problems

c) Cost reduction = (initial cost - final cost) / initial cost

As illustrated in Table 4-7, PI, RN, and ANG methods yield very good results for medium sized problems while API and FIN lag behind. For example, for the 40-job problems, the PI, RN, and ANG strategies can achieve cost reduction of 69.5%, 84.7%, and 93.1% respectively, but the cost reductions are only 59% and 42.3% respectively for API and FIN approaches. As the problem size grows, the superiority of the ANG technique becomes evident. For instance, after 15 minutes of search for 60-job problems, the ANG method reduces the average processing cost by almost 63% while the second best method, RN, leads to only about 30% cost reduction on the average.

The low search efficiency of the PI, API, FIN, and RN methods is probably caused by their neighbourhood generation or the move selection policies. With the PI method, the neighbourhood size and hence the time required to make a move increases rapidly with the increased problem size. Though the neighbourhood size can be limited

for the API, FIN, and RN methods, a large number of moves may be needed to reposition jobs in the permutation.

The convergence curves for a typical 40-job problem corresponding to different neighbourhood generation and move selection mechanisms are plotted in Figure 4-4. These curves show that the ANG method has the best improvement rate and it can achieve most of the cost reduction in the first 5 minutes.

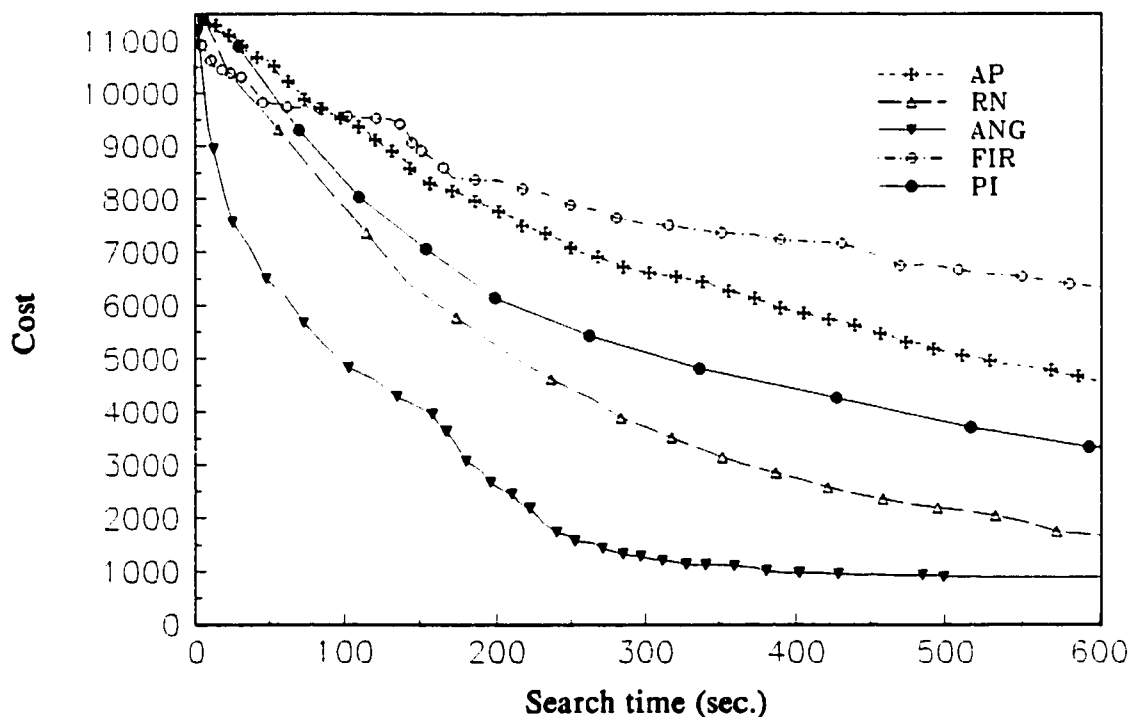


Figure 4-4 Convergence curves for different neighbourhoods

The average costs for the test problems obtained using the ANG method and different diversification strategies are listed in Table 4-8. Referring to Table 4-8, it can be seen that the first diversification strategy, i.e., restarting from the best solution, *S_{best}*, marginally outperforms the third strategy, i.e., no diversification at all. The second strategy, i.e., resuming the search from a randomly selected sequence, appears to be the worst of the three. This result illustrates that search diversification has similar effects on both the ANG method and the regular tabu search method.

Table 4-8 Average costs for the ANG method with different diversification strategies (tabu list size=10, phase size=20)

Problem set	# of jobs ^{a)}	Diversification strategy		
		Restart form best sequence	Restart form random sequence	No diversification
1	30	617 ^{b)}	759	624
2	40	874	2203	891
3	50	2045	3820	2052
4	60 ^{c)}	7885	7885	7885

a) Each set contains 25 randomly generated problems

b) Average of 25 randomly generated problems

c) Search did not complete first phase within the time limit

Nevertheless, it should be mentioned that RN and ANG can be applied only to those problems that exhibit a recognizable pattern in the disruption level (cost difference) with respect to the positions of jobs in the sequence. If such a tendency cannot be found, it may be advantageous to evaluate the entire neighbourhood or to perform other partial searches.

Chapter 5

PLANNING FOR HOLE-MAKING OPERATIONS

5.1 Background

5.1.1 Scope and purpose

Job sequencing at the machine level was the main issue addressed in the previous chapters. The problems were formulated and solved to minimize the total processing cost while considering different performance measures and constraints at the machine level planning. Sequencing problems, however, are not limited to the machine level planning. In many machining operations, such as hole making, operation sequencing and tool provision at the part level pose a great challenge in terms of planning effectiveness and final production cost. Part level planning could greatly affect machine level planning, especially if the time involved in processing a job is a significant portion of the total planning horizon.

In view of this, operation sequencing with the objective of minimizing processing cost at the part level is considered in the present chapter. In particular, the planning problems in hole making are analyzed. The problem formulation and the development of the solution procedure are carried out. The work cited agrees with the pattern established in this research as it again involves sequencing problem with the objective of minimizing total processing cost. The details of the problem and its solution procedure are elaborated in the following sections.

5.1.2 Issues in hole making operations

Hole making operations such as drilling, reaming, and tapping account for a large portion of machining processes for many industrial parts such as dies, jigs, fixtures and moulds. Nowadays CNC machines are widely used in the industry to process such components. In order to utilize the flexibility of these machines and minimize the production cost, it is important that a good process plan be generated within reasonably short period of time. In doing so, close attention should be paid to all parameters contributing to the total production cost.

One survey showed that tool and part movements take on average 70% of the total time in a manufacturing process (Merchant 1985). Due to the point-to-point machining feature, the cutting time in hole making can be significantly smaller than in other machining processes. A considerable amount of processing time is spent in switching tools and moving the table from one drilling location to another. In addition, to drill a given hole to its final size, different sets of tools with different cutting speeds may be selected, which directly influences tooling and machining costs. Hence, to improve machining efficiency and reduce production costs in drilling operations, four issues should be addressed: a) minimization of non-cutting tool travels; b) minimization of tool switches; c) selection of a proper combination of tools to drill a hole to its final size; and d) optimization of machining speed.

These issues are closely related. For instance, a typical plastic injection mould could have over 100 holes of different diameters, depths, tolerance and surface specifications, representing various tool requirements and a large number of tool

switches. In such cases, a tool may be required by several holes and several tools of different diameters may be employed to drill a hole to its final size. To reduce tool travel, it may be desirable not to move the machine table until a hole is completed, using several tools of different diameters. This however will lead to excessive tool switches. By the same token, though tool switches can be reduced by completing all operations on all the holes that require the current tool, the travel time will be increased. Furthermore, the amount of tool movement and the number of tool switches will depend on which set of tools is to be selected to drill each hole to its final size. The machining cost and tool cost are affected by the selection of tool combination for each hole and the machining speeds. For these reasons, the above four issues should be addressed simultaneously. The objective is to minimize the production cost which consists of machining cost, tool cost, non-cutting travel cost, and tool switch cost. Although the focus of this section is on a general class of drilling operations, the solution procedure developed here can be applied for most of the machining processes involving multiple cutting operations.

5.2 Problem statement

As mentioned earlier, this section aims to provide an efficient solution method to determine the best sequence of operations and associated machining speeds in a general class of drilling operations so that the total processing cost is minimized. Generally, a part may have many holes of various diameters, surface finishes, tolerances, and possibly different depths. Depending on the hole diameter, tool geometry and surface quality specifications, a hole may or may not be completed using a single tool. If the diameter

of hole is relatively large, a pilot hole may have to be drilled using a tool of smaller diameter and then a larger tool enlarges it to its final size, with possibly a subsequent reaming or tapping when necessary. A hole can be made using alternative *sets (or combinations)* of tools. Each set of tools can include a pilot tool, a final tool (a tool with the same diameter as that of the final hole) and one or more intermediate tools. Figure 5-1 shows a schematic representation of different set of tools which may be used to drill three holes on a part.

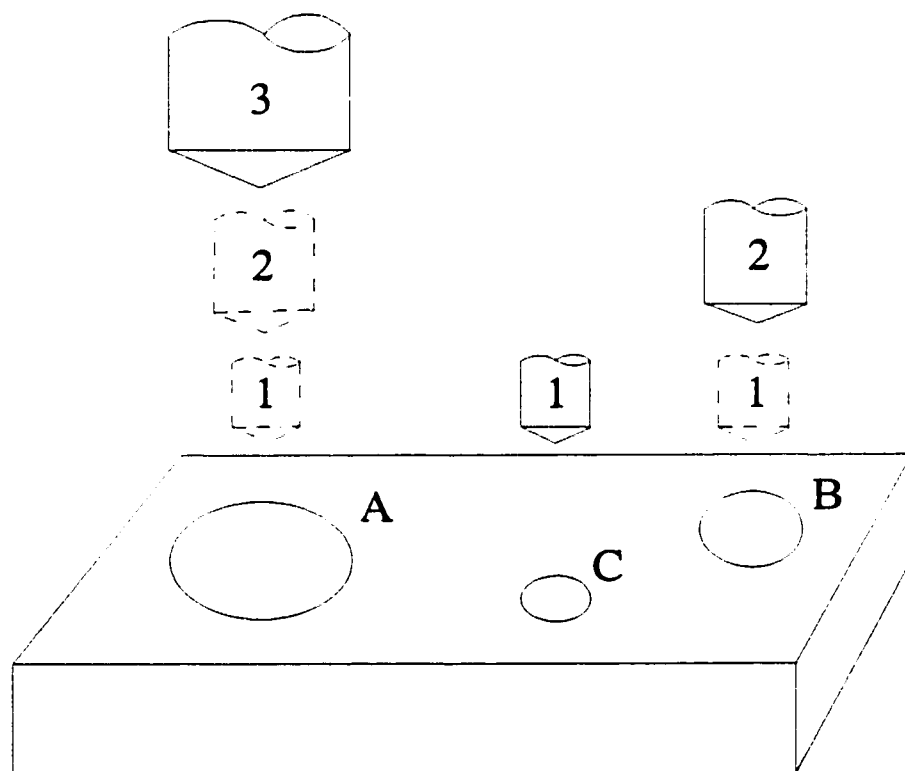


Figure 5-1 A schematic representation of alternative sets of tools for hole making

For each hole in Figure 5-1 the largest tool, displayed by solid lines, has to operate to drill the hole to its final size. Some pilot or intermediate tools, shown by dashed lines, may also be used. For instance, there are four different sets of tools for hole A: {1,2,3}, {2,3}, {1,3}, and {3}. Therefore, the selection of the set of tools for each hole will affect the optimum cutting speeds, required number of tool switches, and tool travel distances.

The problem under consideration is similar to the Travelling Salesman Problem (TSP) in which each node (operation) in a tour (sequence of all operations) must be visited only once. However, unlike the TSP, the cost of visiting each node is determined by its position in the sequence of operations (constructed tour), i.e., certain tools may not be used for some holes even though such operations are feasible (such nodes are visited at no cost). This adds more complexity to the problem as the cost involved in performing a cutting operation in a drilling site is affected by the historical background pertaining to that particular site. That is, the selection of cutting speed and the schedule of tool switches depend upon the depth of cut and hence the current status of the concerned hole which, in turn, is determined by whether or not other tools have already been used on that hole.

Now, the problem is to select a set of operations along with the optimum cutting speed and sequence those operations in such a way that the total processing cost is minimized. The cost components include:

a) *Tool travel cost*: This is the cost of moving the tool to the current drilling position from its previous location. Tool travel cost is proportional to the distance required for the spindle to move between two consecutive drilling locations, p_{jk} . The cost

per unit time is usually lower for non-cutting motions than for the actual cutting processes due to the lower power consumption.

b) Tool switch cost: This cost occurs whenever a different tool is used for the next operation. If for operation ij tool type i is not available on the spindle, then the required tool must be loaded on the spindle prior to performing operation ij . In CNC machining usually each tool is mounted on the tool magazine with its own collet. For any tool switch, the tool magazine has to rotate so that the tool changing device can reach the required tool. The tool switch time depends on the location of the required tool on the magazine. If for operation ij the location of replacing tool i on the tool magazine is far from the current tool, i' , then the magazine has to rotate a longer distance before the tool switch can take place. This causes a longer tool switch time, $t_{ii'}$, and hence a higher tool switch cost.

c) Tool and machining costs: The tool cost consists of the new tool cost and the cost of machine down time required to replace the tool. The operating cost and the machine overhead cost are the major components of machining cost. Both tool and machining costs are affected by machining parameters such as depth of cut, feed rate, and cutting speed. The total tool and machining costs when tool type i is used on hole j can be written as:

$$\hat{C}_{ij} = \frac{\Upsilon_{ij}}{\bar{T}_{ij}} Q_i + C \Upsilon_{ij} \quad (5-1)$$

Where the machining time, Υ_{ij} , is calculated by:

$$\Upsilon_{ij} = \frac{\pi D_i L_j}{1000 V_{ij} f_i} \quad (5-2)$$

and the general tool life expression, $\bar{T}_{ij}$, is as follows: (Zhao, 1992)

$$\bar{T}_{ij} = \left(\frac{A_1 D_i^{A_2}}{V_{ij} e_{ij}^{A_3} f_i^{A_4}} \right)^{A_5} \quad (5-3)$$

The above expression is an empirical formula in which $\bar{T}_{ij}$ is the life of tool type i performing a cutting operation on hole j with the speed of V_{ij} , D_i is the tool diameter, and f_i the recommended feed rate for tool type i which is determined by tool and part materials, required surface finish, and tool geometry. If an existing hole is enlarged by drilling, reaming or tapping, the depth of cut, e_{ij} , is the difference between the tool radius and the current hole radius. The values of A_1 , A_2 , A_3 , A_4 and A_5 will depend on the type of operation (drilling, reaming, tapping, etc).

There is a set of machining parameters with which the total tooling and machining costs is minimized. In drilling operations the depth of cut is fixed. It is also a common practice to keep feed rate constant. Consequently, the optimum cutting speed, V_{ij}^* , can be found by substituting Equations (5-2) and (5-3) into Equation (5-1) and solving the following differential equation:

$$\frac{d\hat{C}_{ij}}{dV_{ij}} = 0 \quad (5-4)$$

The cutting speed given by Equation (5-4) minimizes the sum of machining and tool costs for a given operation (single tool-hole combination). However, it cannot be used

to find total processing cost without knowing which set of operations is to be selected to complete each hole. Therefore, the cutting speeds obtained by solving Equation (5-4) are used only as input data to the search algorithm.

Since the problem in hand involves a single part, the planning decision has to be made frequently and quickly. For this purpose, a tabu search approach is developed to solve the combined tool travel scheduling, tool switch scheduling, tool selection, and machining speed specification problem for hole-making. The details are given in following section.

5.3 Problem formulation

To minimize the production cost, the following model can be formulated:

Model 5-1

$MinG(s) =$

$$Min \sum_{i \in \bar{I}_j} \sum_{i' \in \bar{I}_j} \sum_{i'' \in I} \sum_{l=1}^J \sum_{j=1}^J \sum_{k=1}^J x_{ii'i''ljk} \left[\bar{a} \left(\frac{p_{lj} + p_{jk}}{2} \right) + \bar{t}_{ii''j} c + \frac{\gamma_{ii'j}^*}{\bar{T}_{ii'j}} Q_i + \gamma_{ii'j}^* C \right] \quad (5-5)$$

subject to

$$\sum_{i' \in \bar{I}_j} \sum_{i'' \in I} \sum_{l=1}^J \sum_{k=1}^J x_{ii'i''ljk} = 1 \quad \forall j \quad (5-6)$$

$$x_{ii'i''ljk} + x_{ii'i''kjl} \leq 1 \quad \forall \{l, j, k, i, i', i''\} \quad l \neq j, k \neq j, i \in \bar{I}_j, i' \in \bar{I}_j, i'' \in I \quad (5-7)$$

The objective is to minimize the total cost of processing a job with J holes. The 0-1 decision variables, $x_{ii'i''ljk}$, simultaneously determine the sequence of holes to be

processed as well as the sequence of tools to be used to process each hole. It is noted that to find out the tool path at each instance at least three points, i.e., starting point, end point and a point in between, should be known. That is the reason why the decision variable includes three indices for three holes l , j , and k with j denoting current position of the spindle. Since the distance between two adjacent drilling sites will be counted twice, one for the current path and the other for the next path, the total tool travel distance in the objective function is divided by two to represent the actual tool travel distance. The indices i, i' , and i'' in the 0-1 decision variable are used to determine the proper tool switch order during the operation. Although the tool switch order for the current hole j can be found by the first and third indices, i and i'' , it is important to know which tool is used last to process hole j so that the proper cutting speed and tool life for the current operation can be determined. This is done using i' .

Constraint sets (5-6) guarantee that each hole is drilled to its final size. The last set of constraints means that backward movement of spindle is not allowed unless a tool switch is needed. This prevents redundant spindle movement which may occur when the tool travel cost is negligible.

This model has a large number of 0-1 decision variables. Unfortunately, for medium or large problems, solving this model requires an excessive amount of computational time. To provide an efficient solution procedure, a tabu search approach is presented.

5.4 Tabu search algorithm for hole making problem

The following definitions are provided to facilitate the development of the proposed algorithm.

Possible operation: A possible operation is defined as the operation performed by tool type i on hole j in which a) $i \in \bar{I}_j$ and b) the tool diameter, D_i , is not greater than the final size of the hole to be drilled, $\bar{D}_j$. For a problem involving J holes (drilling sites) and I tool types, the maximum number of feasible tool-hole combinations or *possible operations* is given by:

$$\bar{U} = \sum_{j=1}^J N_j \quad (5-8)$$

Required operation: Operation ij is called a required operation if a) it is a possible operation, and b) no tool with the diameter greater than that of tool type i has been already used for hole j . Hence, for a given feasible operation sequence, s , an operation could be either a possible or a required operation.

For the problem under consideration, a solution is defined as a sequence of all possible operations in which some operations, depending on their positions in the sequence, are the required ones. The starting sequence should include all feasible tool-hole combinations. This ensures that all possible operations are available for search, though some of such operations may be redundant in the final solution, i.e., some tools may not operate on certain holes. This is illustrated in the following example. Consider the part shown in Figure 5-1 in which three holes, A, B, and C are to be drilled to their

final sizes $\bar{D}_A$, $\bar{D}_B$, and $\bar{D}_C$ respectively. Suppose $\bar{I}_A = \{1,2,3\}$, $\bar{I}_B = \{1,2\}$, and $\bar{I}_C = \{1\}$ where $D_3 = \bar{D}_A$, $D_2 = \bar{D}_B$, $D_1 = \bar{D}_C$ and $D_3 > D_2 > D_1$. Then one of the possible starting solutions for the search is $s_0 = \{1C, 1B, 2B, 1A, 2A, 3A\}$. During the search process, some operations may become redundant due to the changed sequence. For instance, sequence s_0 may be changed to $s_n = \{1C, 2B, 1B, 1A, 3A, 2A\}$ after n iterations. In sequence s_n operations 1B and 2A are unnecessary since holes B and C have already been enlarged to the sizes greater than the diameters of tool types 1 and 2 respectively. Therefore only the costs associated to performing operations 1C, 2B, 1A, and 3A contribute to the total processing cost. These are the sets of required operations which will be executed on the machine to complete holes A, B, and C to their final sizes.

The neighbourhood generation mechanism and the rest of search parameters are similar to those explained in preceding chapters. The complete description of the proposed tabu search algorithm is presented below.

Algorithm 5-1

Step 1. Initialize the search

- (a) Read the input data $\bar{D}_j$, p_{ji} , D_i , Q_i , t_{ij} , for $j=1, \dots, J$,
 $i=1, \dots, I$, and $\bar{I}_j, C, \bar{a}, c$
- (b) Read the search parameters, T_size , $Mmax$, $Nmax$
- (c) Calculate total number of possible operations, $\bar{U}$, and set $T_list = \{\emptyset\}$,
 $S_best = \{\emptyset\}$, $M_ctr = 0$, $N_ctr = 0$, and $C_best = M_big$
- (d) Find s^{**} , a feasible starting sequence including all possible operations; i.e.

a sequence with U "required" operations. Compute $G(s^{**})$

Step 2 Search

WHILE $M_ctr < Mmax$ DO

set $G(s^*) = M_big$;

DO $mm = 1$ to $\tilde{U}-1$

DO $nn = mm+1$ to $\tilde{U}$

generate a new neighbour s for s^{**} by swapping the operation in position mm with the operation in position nn .

IF $s \in T_list$, discard s , and continue;

ELSE compute the sum of processing costs for all "required" operations in s , $G(s)$.

IF $G(s) < G(s^*)$, set $G(s^*) \leftarrow G(s)$ and $s^* \leftarrow s$;

ELSE discard s , and continue;

ENDIF

ENDIF

ENDDO

ENDDO

Step 2.1 Move

SET $G(s^{**}) \leftarrow G(s^*)$, $s^{**} \leftarrow s^*$ and update T_list ;

IF $G(s^{**}) < Cbest$, set $Cbest \leftarrow G(s^{**})$, $Sbest \leftarrow s^{**}$,

$M_ctr \leftarrow M_ctr + 1$, and $N_ctr \leftarrow N_ctr + 1$;

```

ELSE set  $M_{ctr} \leftarrow M_{ctr} + 1$ ,  $N_{ctr} \leftarrow N_{ctr} + 1$ ;
ENDIF

```

Step 2.2 Diversify

```

IF  $N_{ctr} \geq N_{max}$ , set  $T_{list} = \{\emptyset\}$ ,  $N_{ctr} \leftarrow 0$ , and diversify the search
path using one of the following strategies:

```

- (1) Restart from S_{best} , the best sequence found so far.
- (2) Restart from a randomly selected sequence.

```

ELSE continue;

```

```

ENDIF

```

```

ENDWHILE

```

Stop

5.5 Case study

The proposed algorithm was applied to determine the set of tools, sequence of operations, and cutting speeds for the upper base of a plastic injection mould shown in Figure 5-2. This figure includes the data about the distances between the holes, type of operations required, and depth of each hole.

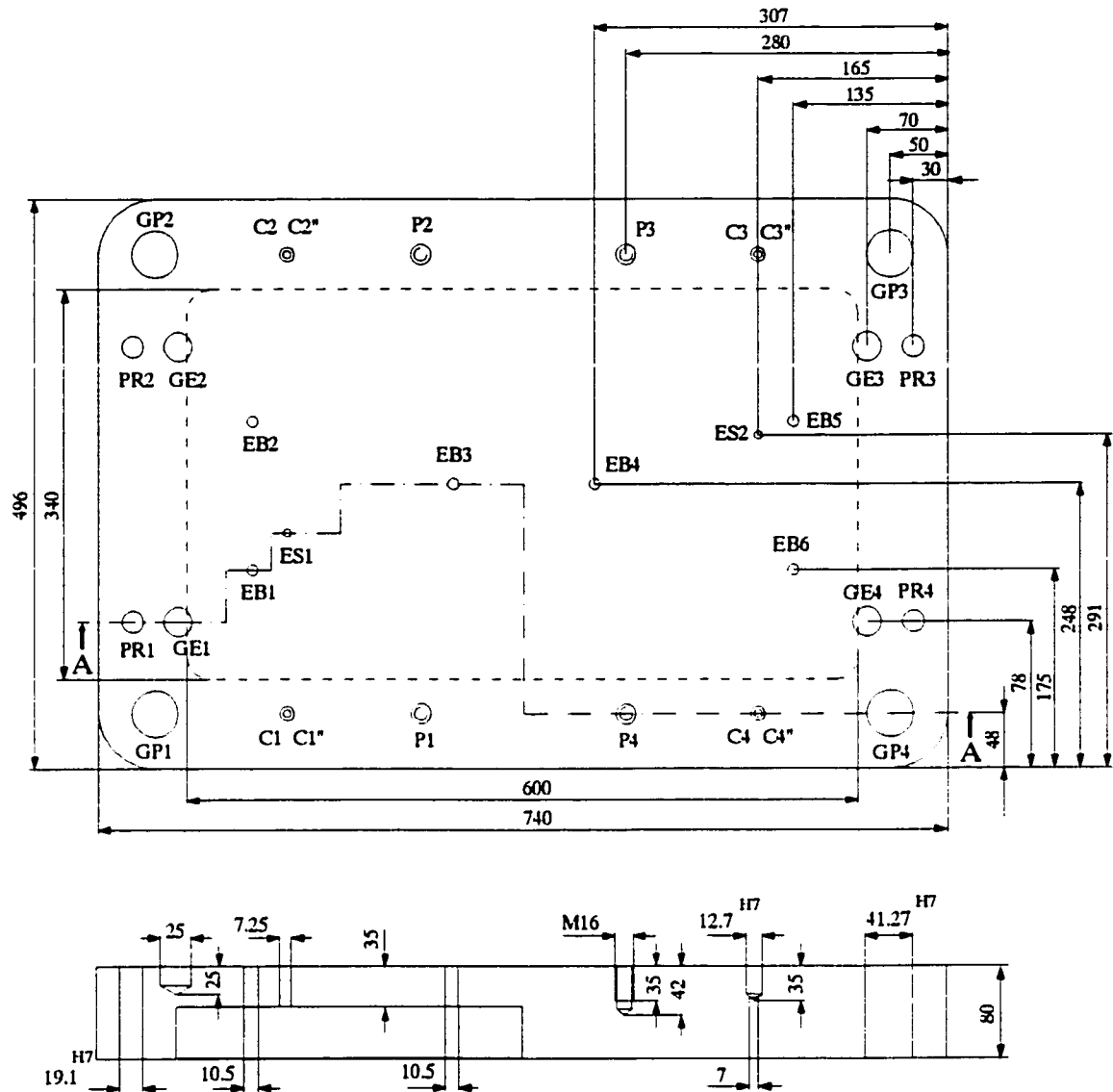


Figure 5-2 Upper holder of the plastic injection mould (courtesy of Komtech Plastics Corp.)

Three types of operations, drilling, reaming, and tapping are required to complete this part. The tool life expressions for these operations are as follows (Zhao, 1992):

$$\bar{T}_{ij} = \left(\frac{8 D_i^{0.4}}{V_{ij} f_i^{0.7}} \right)^5 \quad \text{for drilling a new hole} \quad (5-9)$$

$$\bar{T}_{ij} = \left(\frac{18.4 D_i^{0.4}}{V_{ij} e_{ij}^{0.2} f_i^{0.5}} \right)^5 \quad \text{for enlarging a hole by drilling} \quad (5-10)$$

$$\bar{T}_{ij} = \left(\frac{12.1 D_i^{0.3}}{V_{ij} e_{ij}^{0.2} f_i^{0.65}} \right)^{2.5} \quad \text{for enlarging a hole by reaming or tapping} \quad (5-11)$$

The following optimum cutting speeds can be obtained respectively by solving differential Equation (5-4) with above tool life equations:

$$V_{ij}^* = 6 \sqrt[5]{\frac{C D_i^2}{Q_i f_i^{3.5}}} \quad (5-12)$$

$$V_{ij}^* = 13.9 \sqrt[5]{\frac{C D_i^2}{Q_i e_{ij} f_i^{2.5}}} \quad (5-13)$$

$$V_{ij}^* = 10.3 \sqrt[2.5]{\frac{C D_i^{0.75}}{Q_i e_{ij}^{0.5} f_i^{1.65}}} \quad (5-14)$$

These speeds are used to calculate the tool and machining costs for each operation in the search process. The information about tools is given in Table 5-1. Table 5-2 lists

the tool switch times which are asymmetric and dependant on the tool locations in the tool magazine.

Table 5-1 Tool diameter, cost data, and specified feed rate

Tool type i	drills								reams			tap
	1	2	3	4	5	6	7	8	9	10	11	12
f_i (mm/rev.)	0.12	0.10	0.12	0.15	0.20	0.20	0.18	0.15	0.50	0.80	0.80	1.50
D_i (mm)	7.0	7.25	10.5	12.5	13.0	19.0	25.0	41.0	12.7	19.1	41.2	16.0
Q_i (\$)	10	12	15	15	14	20	26	50	55	70	85	45

Table 5-2 Tool switch times (min.)

Successor tool	Predecessor tool											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.6	0.2	0.4	0.4	0.9	0.6	1.0	0.8	1.4	0.4	1.0
2	0.6	0.0	0.8	1.2	0.4	0.8	1.2	0.5	0.6	0.6	1.2	1.4
3	0.2	0.8	0.0	0.6	1.4	1.2	1.4	1.0	0.4	1.0	1.9	1.2
4	0.4	1.2	0.6	0.0	0.4	0.7	1.0	1.6	0.8	1.4	0.4	0.6
5	0.4	0.4	1.3	0.5	0.0	0.8	0.6	1.0	0.8	1.8	1.6	1.5
6	0.5	0.5	1.2	0.2	0.8	0.0	0.2	1.5	1.2	1.3	0.4	1.9
7	0.6	1.2	1.4	1.0	0.6	0.4	0.0	1.0	0.8	0.8	1.2	0.6
8	1.0	0.6	0.6	0.4	0.4	0.7	0.6	0.0	0.8	1.5	0.4	0.5
9	0.2	0.9	1.1	0.6	1.5	1.2	1.4	1.0	0.0	1.2	2.0	1.2
10	0.4	1.2	0.7	1.5	0.5	0.3	1.0	0.4	0.7	0.0	0.4	0.6
11	0.4	1.2	2.0	0.4	1.2	0.8	0.6	1.2	0.8	1.0	0.0	0.2
12	1.0	0.7	0.3	0.4	0.4	0.8	0.6	1.4	1.0	1.5	0.5	0.0

Since each hole can be made using alternative tool sets, a feasible set of tool-hole combinations should be known to construct the initial sequence of possible operations. Such a set of feasible operations used in this example is tabulated in Table 5-3.

Table 5-3 Possible tool-hole combinations used in the example problem

j	GP1-GP4	GE1-GE4	PR1-PR4	C1-C4	C'1-C'4	P1-P4	EB1-EB6	ES1-ES2
$\bar{I}_j$	1-6-7-8-11	3-6-7	3-6-10	1-4-9	1	1-5-12	3	2

The remaining process parameters assumed for this problem are as follows: $C=\$1/\text{min}$, $\bar{a}=\$0.0008/\text{mm}$, and $c=\$1/\text{min}$.

The data given in Table 5-3 lead to 80 ($\bar{U} = \sum_j N_j$) possible operations. The size of neighbourhood is $\bar{U}(\bar{U}-1)/2=3160$ for this problem. These neighbouring solutions must be evaluated before a move can be made.

A set of computations was carried out on 10 different starting solutions, i.e., 10 different initial sequences. For each initial sequence, the search was terminated after 100 moves. To investigate the effects of diversification strategies and tabu list size on search performance, the following search strategies were used.

Diversification strategy	Phase length (<i>N_{max}</i>)	Tabu list size (<i>T_{size}</i>)
(1) Restart from <i>S_{best}</i>	20	10,15
	50	10,20,40
(2) Restart from a random sequence	20	10,15
	50	10,20,40
(3) No diversification	100	20,40,50

The results obtained using the best combination of starting sequence, tabu list size and diversification strategy are summarized in Table 5-4.

As shown in Table 5-4, the final solution consists of 56 required operations (reduced from 80 possible operations given in initial solution). The total processing cost for the final solution is \$64.8 including \$45.2 machining and tool costs, \$11.0 non-productive travelling cost, and \$8.6 tool switch cost. This result indicates a 47% cost reduction over the initial solution. The operation sequence, tool switch sequence, cutting speed, and tool sets can all be determined simultaneously based on the output shown in Table 5-4. For example, tool type 6 is used first to drill 12 holes (4 GPs, 4 GEs, and 4 PRs) with the cutting speed of 34 m/min. The tool is then switched to tool type 8 to enlarge 4 GP holes to 41 mm. in diameter. The same holes are then reamed to their final sizes (41.2 mm.) using tool type 11 and so on.

Table 5-4 Operations sequence and their corresponding cutting speeds

<i>ij</i>	6-GP4	6-PR4	6-GE4	6-GE3	6-PR3	6-GP3	6-GP2	6-PR2	6-GE2	6-GE1	6-PR1	6-GP1
<i>V</i>	34	34	34	34	34	34	34	34	34	34	34	34
<i>ij</i>	8-GP1	8-GP4	8-GP3	8-GP2	11-GP2	11-GP1	11-GP4	11-GP3	10-PR3	10-PR2	10-PR1	4-C1
<i>V</i>	45	45	45	45	9	9	9	9	10	10	10	36
<i>ij</i>	1-C1'	1-C4'	3-EB6	1-C3'	1-C2'	3-EB2	3-EB1	3-EB3	3-EB4	3-EB5	2-ES2	2-ES1
<i>V</i>	37	37	39	37	37	39	39	39	39	39	40	40
<i>ij</i>	7-GE1	7-GE2	7-GE3	7-GE4	10-PR4	4-C4	4-C3	4-C2	5-P2	5-P3	5-P4	5-P1
<i>V</i>	50	50	50	50	10	36	36	36	30	30	30	30
<i>ij</i>	12-P1	12-P2	12-P3	12-P4	9-C4	9-C3	9-C2	9-C1				
<i>V</i>	4	4	4	4	11	11	11	11				

5.6 Discussions

5.6.1 Solution

The computational results revealed the following findings that could be useful for day to day shop floor planning decisions.

a) There is a trade off between tool travel cost and tool switch cost. The proposed approach minimizes neither the tool switch cost nor the tool travel cost. In fact, the tool switch cost in the final solution is 18% higher than the one for the same problem when the objective function does not include non-productive travelling cost, i.e., $\bar{a}=0$. Also the tool travel distance is almost 400% longer than the case when tool switch cost is not considered, i.e., $c=0$. It is also found that the relative weights of tool travelling cost

and tool switch cost result in notable changes in the final operation sequence. This is illustrated in Table 5-5 which corresponds to the best sequence found for the example problem when the tool switch times in Table 5-2 are reduced by 50%. This sequence has a total processing cost of \$60.2 from which \$45.2 is the tool cost and machining cost, \$10.1 tool switch cost, and \$4.9 tool travel cost. A comparison between the results in Table 5-4 and Table 5-5 indicates that, when tool switch cost is reduced, the search attempts to reduce the tool travel cost even though number of tool switches may increase.

Table 5-5 Operations sequence and their cutting speeds for the example problem when tool switch costs are reduced by 50%

<i>ij</i>	3-EB3	3-EB4	3-EB6	3-EB5	4-C3	6-GP3	6-PR3	6-GE3	7-GE3	1-C3'	8-GP3	2-ES2
<i>V</i>	39	39	39	39	36	34	34	34	50	37	45	40
<i>ij</i>	5-P2	5-P3	12-P3	12-P2	1-C2'	4-C2	6-GE2	7-GE2	6-PR2	6-GP2	8-GP2	11-GP2
<i>V</i>	31	31	4	4	37	36	34	50	34	34	45	9
<i>ij</i>	10-PR2	9-C2	3-EB2	3-EB1	6-GE1	7-GE1	6-PR1	6-GP1	8-GP1	11-GP1	10-PR1	4-C1
<i>V</i>	10	11	39	39	34	50	34	34	45	9	10	36
<i>ij</i>	1-C1'	9-C1	2-ES1	5-P4	5-P1	12-P1	12-P4	1-C4'	4-C4	6-GE4	6-PR4	6-GP4
<i>V</i>	37	11	40	30	30	4	4	37	36	34	34	34
<i>ij</i>	9-C4	8-GP4	7-GE4	11-GP4	10-PR4	10-PR3	11-GP3	9-C3				
<i>V</i>	11	45	10	9	50	50	10	11				

b) Tools selection and spare provision can be simultaneously determined based on computational results. With the reference to Table 5-3, although several tools may be available to drill a hole to its final size, only a few of them will be actually used (see

Table 5-4). For instance, there is no need to use tool types 1 and 7 on holes GP1-GP4 though they are provided in the initial solution. Furthermore, the work load of some tools may exceed the life of those tools and therefore extra copies are needed. This is the case for tool type 6 which has been used as the pilot drill for 12 holes.

In some cases it may be desirable to balance the work load among the tools. One way of doing this is to prevent the cutting time of each tool from exceeding its economic life. For this purpose, the following constraint has to be added to Model 5-1:

$$\sum_{i' \in \bar{I}_j} \sum_{l=1}^J \sum_{j=1}^J \sum_{k=1}^J x_{ii'/ljk} \Upsilon_{ij}^* \leq \bar{T}_i \quad \forall i \quad (5-15)$$

5.6.2 Search performance

1) *Accuracy*: The accuracy of the solutions obtained using the proposed search method was investigated by comparing with optimal solutions. In doing so, five sets of 10-operation problems were generated and solved by enumerating all possible ($10! = 3,628,800$ each) solutions. These problems were also solved using the proposed algorithm. The search was terminated after 1.0 minute for every problem. The comparison showed that the average error is less than 1.7%. This is sufficiently accurate for practical planning purpose.

2) *Effects of diversification strategies and tabu list size*: Throughout the computational experiments two aspects of diversification policies, namely number of phases and starting sequences, were examined. In addition, each strategy was tested with different tabu list sizes to investigate the effect of tabu list size. It should be mentioned that the total

number of moves was fixed for all problems for fair comparison. It is found that a tabu list with a size less than or equal to 15 may lead to cycling while tabu lists with sizes 20 to 40 perform equally well in terms of solution quality. For the majority starting solutions, diversification strategies 1 and 3 yield similar results whereas strategy 2 appears to be the least efficient one. In summary, the search is more sensitive to the initial solution than the diversification strategy. The above work has been partially reported in Kolahan and Liang (1996).

5.7 Improvement of the tabu search algorithm

Generally, computational time is an important issue in solving the day to day shop floor planning problems. The study carried out in this chapter involves solving a number of planning problems for a single part and hence it becomes even more important to obtain a good solution in a short period of search time.

In the proposed tabu search algorithm, the neighbourhood size is the product of two parameters: the number of holes and the number of available tools. As the value of either parameter increases, the number of tool-hole combinations, and consequently the neighbourhood size, grows very rapidly. Consequently, the search time for large size problems may be too long for real applications. In this section, it is attempted to improve the search performance by reducing the computational time without sacrificing the solution quality. This can be achieved through adapting different neighbourhood generation mechanisms and move selection policies. The details are given in the following.

As mentioned earlier, in the basic tabu search all candidate moves are generated by the pairwise interchange mechanism and then a move is made toward the best non-tabu solution in the neighbourhood. For a drilling process with $\bar{U}$ possible operations, the lower and upper limits of transposition range are 1 and $\bar{U} - 1$ respectively. This approach is sometimes called steepest descent. The adjacent pairwise interchange, or fastest descent method (Della Croce, 1995), is another neighbourhood generation scheme in which only the adjacent jobs are switched to generate the candidate moves. Since only the adjacent jobs are exchanged, the transposition range for this method is equal to 1 resulting in a neighbourhood of size $\bar{U} - 1$ candidate moves. Although adjacent pairwise interchange does not guarantee a best solution at each iteration, its neighbourhood size is much smaller than that of pairwise interchange and hence the search can progress much faster in the solution space. In the following, two approaches will be introduced which combines the above policies in order to improve the search performance.

Partial Neighbourhood (PN): This approach is proposed to achieve a compromise between the quality of candidate moves given by pairwise interchange and the search speed resulted from adjacent pairwise interchange mechanism. The objective is to find a transposition range that provides the best combination of these parameters. Therefore, the upper boundary of transposition range, RA , can be set between 1 and $\bar{U} - 1$ which is fixed for the entire search process. The best range is then selected based on the computational results for a given search time. Usually, the best transposition range (and hence the neighbourhood size) is between those of pairwise and adjacent pairwise interchange mechanisms.

Dynamic Neighbourhood (DN): This method aims to take advantage of both adjacent pairwise interchange and simple pairwise interchange mechanisms by dynamically changing the transposition range in different stages of the search process. The idea is to guide the search as fast as possible toward the unsearched areas where the optimum solution may be located and then find such a solution by complete evaluation of the entire neighbourhood. To achieve this, the adjacent pairwise interchange is used to speed up the search process at the early stages. As the search progresses, the neighbourhood size is enlarged by increasing the transposition range which, in turn, improves the solution quality at each iteration. Finally, in the later iterations, the pairwise interchange mechanism is employed to generate the neighbourhood. This will maximize the possibility of finding the best solution in the current neighbourhood. To this end, we proposed the following linear function to determine the transposition range at each iteration.

$$RA(n) = \begin{cases} 1 & \text{if } Z \leq 1 \\ Z & \text{if } 1 < Z < \bar{U} - 1 \\ \bar{U} - 1 & \text{if } Z \geq \bar{U} - 1 \end{cases} \quad (5-16)$$

Where

$$Z = A + Bn \quad (5-17)$$

In the above expression, $RA(n)$ is the upper limit of transposition range at the n th iteration, and A and B are the constant values determined experimentally. It should be mentioned that the above methods are based on the size of generated neighbourhood (not the location of the exchanged operations) which can be controlled by changing

either the upper or the lower boundary. Therefore, in Equation (5-16) only the upper bound of transposition range is determined and its lower bound can be set to 1 throughout the search.

For comparison, the computations were repeated for 10 minutes of search time for each of the three neighbourhood generation and move selection policies; namely, pairwise interchange (PI), partial neighbourhood (PN), and dynamic neighbourhood (DN). The following values are found to be best suited for our computations: $RA = \bar{U} / 2$ for PN and $A=4$, $B=1.5$ for DN. Figure 5-3 illustrates the convergence curves for different neighbourhood generation and move selection mechanisms.

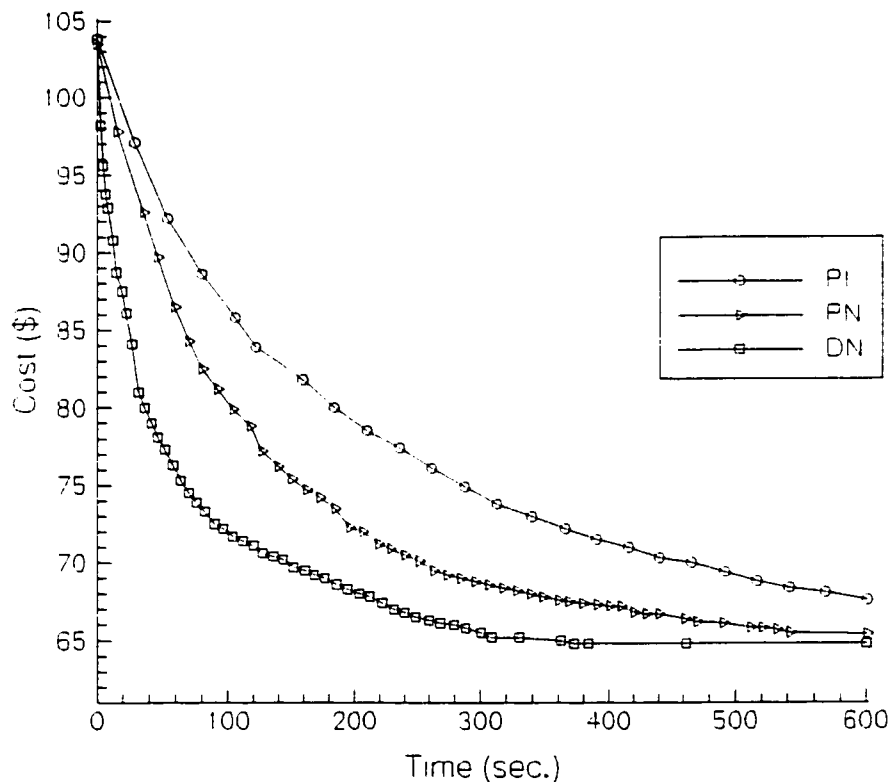


Figure 5-3 Convergence curves for different neighbourhood mechanisms

As illustrated in Figure 5-3, all the three curves tend to converge to the same solution in the long run. However, the DN method converges much faster towards the final solution and most of the cost reduction is achieved in the first 5 minutes of search time. This has a significant impact on the search performance specially for the large size problems or when there is a limited search time available. The main advantage of the proposed PN and DN methods is that they can greatly reduce the computational time with a little tuning.

Chapter 6

SUMMARY AND FUTURE WORK

6.1 Summary and conclusions

Over the past decades, the complexity involved in the decision making process in automated manufacturing systems has increased dramatically. The performance of these systems depends, in part, upon the techniques used to plan and control the flow of tools and parts in the system. Therefore, the development of easily adaptable and reliable scheduling methods becomes increasingly important. Recently, a number of intelligent search methods have been developed and have produced reasonable solutions to a variety of scheduling problems. In particular, tabu search is reported to be a robust solution technique that can provide consistently good solutions to this type of problems.

In this study, tabu search has been applied to a number of new single machine scheduling problems. First, mathematical models are developed to simultaneously address different elements contributing to production cost. Then, tabu search is employed to solve each problem and its specific implementation issues are explained in detail. Comprehensive computational tests are conducted to investigate the impact of the different parameter settings. Comparison of the results under each setting is also presented to illustrate the relative importance of search parameters on its performance and to provide a general guideline for selection of these parameters.

Nevertheless, no direct comparison is provided between the performance of tabu search and those of its counterparts in solving the problems addressed in this study, nor is it necessary. This is because, with the fine parameters tuning, it is possible to produce comparable results using any of the neighbourhood search methods. Therefore, the efficiency of these methods should be measured by the effort needed to tune them rather than the computational time required to reach the final solution for a given trial. Having said that, conclusions which can be drawn from the work carried out in this dissertation can be summarized as follows.

- a) For scheduling problems in automated machining centres, mathematical models have been developed to incorporate different performance measures and objectives. The performance measures considered in developing these models are different from those in literature and the resultant models give a more realistic presentation of the machine-level planning problems.
- b) It is shown that tabu search can provide consistently good solutions in reasonable computational times. The computational results also show that tabu search is insensitive to the input data and requires very little tuning. Tabu list size and diversification strategies seem to be the only parameters that have some effects on solution quality. As shown by the results, these parameters can be easily determined. For instance, tabu list size can be safely set at 20 moves for all the problems. Similarly, the results indicate that the regular tabu heuristic with no diversification provides reasonably good search paths, although the diversification strategy with search restarting from the best solution, S_{best} , is superior to the

other two tested strategies. The effect of starting solution, in term of solution quality, is insignificant and search would finally converge to the same solutions. In term of computational time, for most of the cases, the majority of cost improvement is obtained within the first 5 minutes of search and any additional improvement is negligible thereafter. For the class of early/tardy problems, the search time is greatly improved by using the proposed ANG mechanism. The computational experiments proved that the proposed neighbourhood structure outperforms the existing neighbourhood generation mechanisms in both search time and solution quality. Finally, for hole making operations, two partial neighbourhood search schemes were proposed. It is shown that, particularly for the large size problems, these methods can considerably reduce the computational efforts.

- c) The hardware/software requirements in implementing the proposed algorithms are readily available. A 486 PC and a C compiler will suffice for computations. Furthermore, the input data may be read directly from the manufacturing database if available. However, comprehensive data are needed if the models are to be implemented for real life applications. Fortunately, with the advent of computer integrated manufacturing systems and CAD/CAM interfaces, most of the data can be acquired in the automated manufacturing environment. For example, the machining time and setup time data are accessible from the process plans of associated parts, and tool and raw material cost data are available in purchasing department. Though, the tool reliability data for different operating conditions

may not be readily available, they can be estimated based on the shop floor data. In brief, even though the actual implementation of the proposed procedure may require large amount of data from the process plan and shop floor, we believe that it is feasible to use the proposed approach as a tool for shop floor planning.

6.2 Recommendations for future studies

Several areas of research directly associated with the work carried out in this study were identified which require further investigation.

- 1) *Single-machine planning in different contexts.* The single machine scheduling problem can be modeled to incorporate other performance measures and constraints. This could include preemptive scheduling, scheduling with restricted job release dates, and scheduling with precedence constraints between the jobs. The algorithms developed in this thesis may be used, with some modifications, to solve these problems.
- 2) *Combined search schemes.* Tabu search was used as the sole solution technique in this research and, whenever possible, its performance was improved solely by adjusting the search parameters and modifying its neighbourhood generation and move selection mechanism. It is possible to develop hybrid search methods to further improve the solution quality and reduce the computational effort. A combination of the modern search techniques such as genetic algorithm and simulated annealing with the tabu search is a promising approach in solving combinatorial scheduling problems.

- 3) *Application to multi-machine planning problems.* Though the main purpose of this research was to solve planning problems for the single machine systems, many multi-machine planning problems share similar properties. The tabu search approach proposed in this study may be extended to address planning problems in the multi-machine manufacturing environments.

REFERENCES

- Abdul-Razaq, T.S., and Potts, C.N., 1988, Dynamic programming state-space relaxation for single-machine scheduling. *Journal of Operational Research Society*, **39** (2), 141-152.
- Adenso-Diaz, B., 1992, Restricted neighbourhood in the tabu search for the flowshop problem. *European Journal of Operational Research*, **62**, 27-37.
- Alidaee, B., and Ahmadian, A., 1993, Two parallel machine sequencing problems involving controllable job processing times. *European Journal of Operational Research*, **70**, 335-341.
- Avci, S., and Akturk, M.S., 1996, Tool magazine arrangement and operations sequencing on CNC machines. *Computers and Operations Research*, **23** (11), 1069-1081.
- Bagchi, U., 1989, Simultaneous minimization of mean and variation flow time and waiting time in single machine systems. *Operations Research*, **37**, 118-125.
- Bard, J., and Feo, T.A., 1989, The cutting path and tool selection problem in computer aided process planning. *Journal of Manufacturing Systems*, **8**, 17-26.
- Bard, J.F., 1988, A heuristic for minimizing the number of tool switches on a flexible machine. *IIE Transactions* , **20**, 382-391.
- Barnes, J.W., and Chambers, J.B., 1995, Solving the job shop scheduling problem with tabu search. *IIE Transactions*, **27**, 257-263.

Barnes, J.W., and Laguna, M., 1991, A review and synthesis of tabu search applications in production scheduling problems. *Technical Report Series, Graduate Program in Operations Research*, The University of Texas at Austin, ORP 91-05.

Bector, C.R., Gupta, Y.P., and Gupta, M.C., 1991, Optimal scheduling of jobs about a common due date on a single machine. *International Journal of System Science*, **22** (12), 2541-2552.

Bellmore, M., and Nemhauser, G., 1968, The travelling salesman problem: A Survey. *Operations Research*, **16**, 538-558.

Billatos, S.B., and Kendall, L.A., 1991, A general optimization model for multi-tool manufacturing system. *Journal of Engineering for Industry*, **113** (1), 10-16.

Brandimarte, P., 1993, Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, **41**, 157-183.

Carric, A.S., and Perera, D.T.S., 1986, Work scheduling in FMS under tool availability constraints. *International Journal of Production Research*, **24**, 1299-1308.

Chandra, P., Li, S., and Stan, M., 1993, Job and tool sequencing in automated manufacturing environment. *International Journal of Production Research*, **31** (12), 2911-2925.

Chauney, F., Loulou, R., Wagneur, E., 1987, Sequencing punch operations in an FMS: a three-dimensional space filling curve approach. *INFOR*, **25**, 26-45.

Chen, C.L., and Buflin, R.L., 1993, Complexity of single machine multicriteria

scheduling problems. *European Journal of Operational Research*, **70**, 115-121.

Chen, C.L., Usher, J.M., and Palanimuthu, N., 1998, A tabu search based heuristic for a flexible flow line with minimum flow time criterion. *International Journal of Industrial Engineering*, **5** (2), 157-168.

Cheng, T.C.E., Oguz, C., and Qi, X.D., 1996, Due-date assignment and single machine scheduling with compressible processing times. *International Journal of Production Economics*, **43**, 29-35.

Cheng, T.C.E., Chen, Z., Li, C., and Lin, B.M., 1998, Scheduling to minimize the total compression and late costs. *Naval Research Logistics*, **45**, 67-82.

Commare, U.L., Diega Nord, L.A., and Passannanti, A., 1983, Optimum tool replacement policies with cost for unforeseen tool failure. *Journal of Machine Tool Decision Research*, **23** (4), 237-243.

Cook, N.H., 1973, Tool wear and tool life. *Journal of Engineering for Industry*, **11**, 931-938.

Crama, Y., 1997, Combinatorial optimization models for production scheduling in automated manufacturing systems. *European Journal of Operational Research*, **99**, 136-153.

Crauwels, H.A.J., Potts, C.N., Van Wassenhove, L.N., 1996, Local search heuristics for single-machine scheduling with batching to minimize the number of late jobs. *European Journal of Operational Research*, **90**, 200-213.

De, P., Ghosh, J.B., and Wells, C.E., 1993, On the general solution for a class of early/tardy problems. *Computers and Operations Research*, **20** (2), 141-149.

Della Croce, F., 1995, Generalized pairwise interchanges and machine scheduling. *European Journal of Operational Research*, **83**, 310-319.

Dileepan, P., and Sen, T., 1988, Bicriterion static scheduling research for a single machine. *Omega*, **16**, 53-59.

Elsyed, E.A., Lee, M.K., Kim, S., and Scherer, E., 1993, Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals. *International Journal of Production Research*, **31**, 727-738.

Emmons, H., 1975, One machine sequencing to minimize mean flow time with minimum tardy. *Naval Research Logistic*, **22** (3), 585-592.

French, S., 1982, Sequencing and scheduling: An introduction to the mathematics of the job-shop. *Ellis Horwood Limited*, Chichester, USA.

Fry, T.D., Armstrong, R.D., and Rosen, L.D., 1990, Single machine scheduling to minimize mean absolute lateness: A heuristic solution. *Computers and Operations Research*, **17**, 105-112.

Fry, T.D., Armstrong, R.D., and Lewis, H., 1988, A framework for single machine multiple objective sequencing research. *Omega*, **17**, 595-607.

Ghosh, S., Melnyk, S.A., and Ragatz, G.L., 1992, Tooling constraints and shop floor scheduling: evaluating the impact of sequence dependency. *International Journal of*

Production Research, **30** (6), 1237-1253.

Glover, F., 1989, Tabu search - Part I. *ORSA Journal of computing*, **1** (3), 190-206.

Glover, F., 1990 a, Tabu search - Part II. *ORSA Journal of computing*, **2** (1), 4-32.

Glover, F., 1990 b, Tabu search: a tutorial. *Interfaces*, **20**, 74-94.

Glover, F., 1986, Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, **13**, 533-549.

Golden, B., Bodin, L., Doyle, T., and Stewart, W., 1980, Approximate travelling salesman algorithms. *Operations Research*, **28**, 694-712.

Gordon, V. S., and Strusevich, V. A., 1999, Earliness penalties on a single machine subject to precedence constraints: SLK due date assignment. *Computers and Operations Research*, **26** (2), 157-177.

Gray, A.E., Seidmann, A., and Steckel, K.E., 1993, A synthesis of decision model for tool management in automated manufacturing. *Management Science*, **39** (5), 549-567.

Gupta, M.C., Gupta, Y.P., and Bector, C.R., 1990, Minimizing the flow time variance in single machine systems. *Journal of operational Research Society*, **41**, 767-779.

Gupta, M.C., Gupta, Y.P., and Kumar, A., 1993, Minimizing flow time variance in a single machine system using genetic algorithms. *European Journal of Operational Research*, **70**, 289-303.

Gupta, S.K., and Kyparisis, J., 1987, Single machine scheduling research. *Omega*, **15**, 207-227.

Hao, Q., Yang, Z., Wang, D., and Li, Z., 1996, Common due-date determination and sequencing using Tabu search. *Computers & Operations Research*, **23** (5), 409-417.

Hertz, A., and de Werra, D., 1991, The tabu search metaheuristic: how we used it. *Annual Mathematics in Artificial Intelligence*, **1**, 111-121.

Hitomi, K., Nakamura, N., and Inous, S., 1979, Reliability analysis of cutting tool. *Journal of Engineering for Industry*, **101** (1), 185-190.

Hoogeveen, J.A., 1992, Single machine bicriteria scheduling. *Ph.D thesis*, CWI University, Amsterdam, The Netherlands.

Hubscher, R., and Glover, F., 1994, Applying tabu search with influential diversification to multiprocessor scheduling. *Computers and Operations Research*, **21** (8), 877-884.

Iakovou, E., Chi, M., and Christos, K., 1996, Optimal solutions for the machining economics problem with stochastically distributed tool lives. *European Journal of Operational Research*, **92**, 63-68.

Jackson, J.R., 1955, Scheduling a production line to minimize maximum tardiness. *Management Sciences Research Project*, Report **43**, UCLA, Los Angeles, CA.

James, R.J.W., and Buchanan, J.T., 1997, Neighbourhood scheme with a compressed solution space for the early/tardy scheduling problem. *European Journal of Operational Research*, **102**, 513-527.

Jeang, A. and Yang, K., 1992, Optimal tool replacement with nondecreasing tool wear. *International Journal of Production Research*, **30**, 299-314.

Jianqiang, M., and Moi Keow, L., 1997, Economical optimization of tool replacement intervals. *Integrated Manufacturing Systems*, **8** (1), 59-62.

Kellerer, H., Tautenhahn, T., and Woeginger, G.J., 1996, Approximability and nonapproximability results for minimizing total flow time on a single machine. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 418-426.

Khoshnevis, B., and Tan, W., 1995, Automated process planning for hole making. *Manufacturing Review*, **8** (2), 106-113.

Knox, J., and Glover, F., 1989, Comparative testing of travelling salesman heuristics derived from tabu search, genetic algorithms, and simulated annealing. Working paper, California State Polytechnic University.

Koksalan, M., Azizoglu, M., and Kondakci, S.K., 1998, Minimizing flow time and maximum earliness on a single machine. *IIE Transactions*, **30**, 192-200.

Kolahan, F., Liang, M., **1998**, An adaptive TS approach to JIT sequencing with variable processing times and sequence-dependent setups. *European Journal of Operational Research*, **109**, 142-159.

Kolahan, F., and Liang, M., **1996**, A tabu search approach to optimization of drilling operation. *Proceedings of the 19th International Conference on Computers and Industrial Engineering*, Miami, Florida, 371-374.

Kolahan, F., and Liang, M., 1994, Analysis of part sequencing and tool replacement for an automated machining centre. *Proceedings of 1994 Industry, Engineering and Management Systems Conference (IEMS '94)*, Cocoa Beach, Florida, 260-265.

Kolahan, F., Liang, M., and Zuo, M., 1995, Solving the combined part sequencing and tool replacement-problem for an automated machining center - a tabu search approach. *International Journal of Computers and Industrial Engineering*. **28** (4), 731-743.

Kondakci, S., and Bekiroglu, T., 1997, Scheduling with bicriteria: total flow time and number of tardy jobs. *International Journal of Production Economics*, **53** , 91-99.

Kondakci, S., Azizoglu, M., and Koksalan, M., 1996, Note: Bicriteria scheduling for minimizing flow time and maximum tardiness. *Naval Research Logistics*, **43**, 929-936.

Koulamas, C.P., 1991, Simultaneous determination of optimal machining conditions and tool replacement policies in constrained machining economics problems by geometric programming. *International Journal of Production Research*, **29** (12), 2407-2421.

Kyparisis, J., and Douligeris, C., 1993, Single machine scheduling and selection to minimize total flow time with minimum number tardy. *Journal of Operational Research Society*, **44** (8), 835-838.

Lam, S. and Cai, X., 1998, Minimizing earliness and tardiness of job completions about a fuzzy due date. *IEEE International Conference on Fuzzy Systems*, **2**, Piscataway, NJ, USA. 869-872.

Lee, I., 1991, Single machine scheduling with controllable processing times: a parametric study. *International Journal of Production Economics* **22** (2), 105-110.

Lee, C.Y., and Kim, S.J., 1995, Parallel genetic algorithm for the earliness-tardiness job scheduling problem with general penalty weights. *Computers and Industrial Engineering*, **28** (2), 231-243.

Lee, C.Y., and Choi, J.Y., 1995, A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights. *Computers and Operations Research*, **22** (8), 857-869.

Lenstra, J.K., Rinnoy, A.H.G., and Brucker, P., 1977, Complexity of machine scheduling problems. *Annals of Discrete Mathematics* **1**, 343-362.

Liaw, C., 1999, Tabu search algorithm for the open shop scheduling problem. *Computers and Operations Research*, **26** (2), 109-126.

Lin, Chen-Sin, 1993, Single machine scheduling to minimize the weighted sum of earliness and tardiness with maximum allowable tardiness. *Proceeding 2nd Industrial Engineering Research Conference*, USA. 301-307.

Liu, P.H., Makis, V., and Jardine, A.K.S., 1997, Joint scheduling of the optimal tool replacement times and optimal operation sequencing in a Flexible Manufacturing System. *Proceeding 6th Industrial Engineering Research Conference*, USA. 334-339.

Luong, L.H.S., and Spedding, T., 1995, An integrated system for process planning and cost estimation in hole making. *International Journal of Manufacturing Technology*, **10**, 411-415.

Lyu, J., Gunasekaran, A., and Ding, J.H., 1996, Simulated annealing algorithm for solving the single machine early/tardy problem. *International Journal of systems science*,

27 (7), 605-610.

Maccarini, G. C., Zavanella, L., and Bugini, A., 1991, Production cost and tool reliabilities: The machining cycle influence in flexible plants. *International Journal of Machine Tools and Manufacturing*, 31 (3), 415-424.

Makis, V., 1996, Optimal tool replacement with asymmetric quadratic loss. *IIE Transactions*, 28, 463-466.

Malek, M., Guruswamy, M., Owens, H., and Pandya, M., 1989, Serial and parallel search techniques for the travelling salesman problem. *In Annals of Operation Research: Linkages with Artificial Intelligence*, 21, 59-84.

Merchant, R.L., 1985, World trends and prospects in manufacturing technology. *International Journal for Vehicle Design*, 6, 121-138.

Merten, A.G., and Muller, M.E., 1972, Variance minimization in single machine sequencing problem. *Management Science*, 18, 518-528.

Mittal, J.M., and Lewis, R.L., 1989, A micro process planning system based on integer programming for prismatic parts produced on horizontal machining centers. *Annals of Operations Research*, 17, 273-290.

Nayanzin, N., 1989, Organizing optimum group flows of parts and tools in an FMS. *Soviet Engineering Research*, 9 (12), 82-85.

Nowicki, E., and Zdrzalka, S., 1996, Single machine scheduling with major and minor setup times: A tabu search approach. *Journal of Operational Research Society*, 47 (8),

1054-1064.

Nowicki, E., and Zdrzalka, S., 1990, A survey of results for sequencing problems with controllable processing times. *Discrete Applied Mathematics*, **25**, 271-287.

Oguz, C., and Dincer, C., 1994, Single machine earliness-tardiness scheduling problems using the equal-slack rule. *Journal of Operational Research Society*, **45** (5), 589-594.

Ouenniche, J. and Boctor, F., 1998, Sequencing, lot sizing and scheduling of several products in job shops: the common cycle approach. *International Journal of Production Research*, **36** (4), 1125-1140.

Ow, P.S., and Morton, T.E., 1988, The single machine early/tardy problem. *Management Science*, **35**, 177-191.

Palei, S.M., and Zubarev, P.M., 1990, Optimum tool-change in an FMM to ensure planned output of good parts. *Soviet Engineering Research*, **61** (2), 7-9.

Panwalker, S.S., and Rajagopalan, R., 1992, Single machine sequencing with controllable processing times, *European Journal of Operational Research*, **59**, 298-302.

Punnen, A.P., and Aneja, Y.P., 1993, Categorized assignment scheduling: a tabu search approach. *Journal of Operational Research Society*, **44** (7), 673-679.

Raghavchari, M., 1988, Scheduling problems with non-regular penalty functions - a review. *Operations Research*, **25**, 144-164.

Ramalingam, S., Peng, Y., and Waston, J., 1987, Tool life distributions, Part 3:

Mechanism of single injury tool failure and tool life distribution in interrupted cutting. *ASME, Journal of Engineering for Industry*, **100**, 193-200.

Ramalington, S. and Watson, J.D., 1977, Tool life distribution. Part 1: Single-injury tool life model. *Journal of Engineering for Industry*, **99** (3), 519-522.

Reeves, C. R., 1993, Improving the efficiency of tabu search for machine sequencing problems. *Journal of the Operational Research Society*, **44** (4), 375-382.

Rinnooy Kan, A.H.G., 1976, Machine complexity problems: classification complexity and computations. *Nijhoff, The Hague*.

Roychoudhury, B., and Muth, J.F., 1995, Tool path optimization procedures for machine tools. *International Journal for Computers and industrial Engineering*, **28**, 367-377.

Sarin, S.C., Erel, E., Steiner, G., 1991, Sequencing jobs on a single machine with a common due date and stochastic processing times. *European Journal of Operational Research*, **55**, 188-198.

Shewchuk, J.P., and Chang, T.C., 1995, Resource-constrained job scheduling with recyclable resources. *European Journal of Operational Research*, **81**, 364-375.

Sinclair, M., 1993, Comparison of the performance of modern heuristics for combinatorial optimization on real data. *Computers and operations research*, **20** (7), 687-695.

Skorin-Kapov, J. and Vakharia, A J., 1993, Scheduling a flow-line manufacturing cell: A tabu search approach. *International Journal of Production Research*, **31** (7),

1721-1734.

Smith, W.E., 1956, Various optimizers for single-stage production. *Naval Research Logistic*, 1, 59-66.

Srivastava, B., and Chen, W., 1993, Part type selection problem in flexible manufacturing systems: Tabu search algorithm. *Annals of Operations Research*, 41, 279-298.

Ssemakula, M.E., and Rangachar R.M., 1989, The prospects of process sequence optimization in CAPP systems. *Computers and Industrial Engineering*, 16, 161-170.

Taiber, J.G., 1996, Optimization of process sequencing considering prismatic workpieces. *Advances in Engineering Software*, 25, 41-59.

Taillard, E., 1990, Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47, 65-74.

Tang, C.S., and Denardo, E.V., 1988, Models arising from a flexible manufacturing machine, part I: minimization of the number of tool switches. *Operations Research*, 36, 767-777.

Taylor, F.W., 1907, On the art of cutting metal. *ASME Transactions*, 28, 102-107.

Tsiushuang, C., Xiangtong, Q., and Fengsheng, T., 1997, Single machine scheduling to minimize weighted earliness subject to maximum tardiness. *Computers and Operations Research*, 24 (6), 147-152.

Tsubakitani, S., and Evans, J.R., 1998, Optimizing the tabu list size for the travelling salesman problem. *Computers and Operations Research*, **25** (2), 91-97.

Usher, M., Fernandes, J., Dhage, S., and Sharma, G., 1997, Object-oriented implementation of tool selection for dynamic process planning. *Proceedings of IIE Industrial Engineering Research Conference*, Norcross, GA, USA, 895-900.

Veeramani, D., Upton, D.M., and Barash, M.M., 1992, Cutting-tool management in computer-integrated manufacturing. *International Journal of FMS.*, **3** (4), 237-265.

Veral, E., and Mohan, R., 1996, Effects of sequencing on job flow times. *Proceedings-Annual Meeting of the Decision Sciences Institute*, Atlanta, GA, USA, 1289-1295.

Vickson, R.G., 1980, Two single machine sequencing problems involving controllable job processing times. *AIIE Transactions*, **12**, 258-262.

Walas, R.A., and Askin, R.G., 1984, An algorithm for NC turret punch press tool location and hit sequencing. *IIE Transactions*, **6**, 280-287.

Wang, L., and Wang, M., 1997, Hybrid algorithm for earliness-tardiness scheduling problem with sequence dependent setup time. *Proceedings of the IEEE Conference on Decision and Control*, **2**, Piscataway, NJ, USA, 1219-1222.

Wen, U., and Yeh, C., 1997, Tabu search methods for the flow shop sequencing problem. *Journal of the Chinese Institute of Engineers*, **20** (4), 465-470.

Weng, M. X., and Ventura, A., 1994, A note on Single machine scheduling for minimizing total cost with identical, asymmetrical earliness and tardiness penalties.

International Journal of Production Research, **32**, 2725-2729.

William, H.K.B., 1994, Experimental investigation of intelligent search methods for job scheduling. *Ph.D Dissertation*, Faculty of Operations Management, Georgia Institute of Technology.

Yagiura, M., and Ibaraki, T., Metaheuristics as robust and simple optimization tools. *Proceedings of the IEEE Conference on Evolutionary Computation*, Piscataway, NJ, USA. 541-546.

Yano, C.A., and Kim, Y.D., 1991, Algorithm for a class of single machine weighted tardiness and earliness problems. *European Journal of Operational Research*, **52**, 167-178

Zhang, F., Li, S., Foulds, L.R., and Tang, G., 1996, Single machine scheduling with controllable processing times and compression costs. *Proceeding 5th Industrial Engineering Research Conference*, Norcross, GA, USA. 423-428.

Zhao, R., 1992, Handbook for Machinists. *Shanghai Science and Technology Press*, China.

Zheng, W., Nagasawa, H., and Nishiyama, N., 1993, Single machine scheduling for minimizing total cost with identical, asymmetric earliness and tardiness penalties. *International Journal of Production Research*, **31**, 1611-1620.

Zhou, C., Chandra J., and Wysk, R., 1990, Optimal cutting tool replacement based on tool wear states. *International Journal of Production Research*, **28**, 1357-1367.