Multi-objective dynamic scheduling of serial and batch manufacturing processes

Gupta, Amit Kumar

2006

Gupta, A. K. (2006). Multi-objective dynamic scheduling of serial and batch manufacturing processes. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/6552>

<https://doi.org/10.32657/10356/6552>

Nanyang Technological University

Downloaded on 26 Apr 2023 20:47:51 SGT

**MULTI-OBJECTIVE DYNAMIC SCHEDULING
OF SERIAL AND BATCH
MANUFACTURING PROCESSES**

AMIT KUMAR GUPTA

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2006

MULTI-OBJECTIVE DYNAMIC SCHEDULING OF SERIAL AND BATCH MANUFACTURING PROCESSES

Amit Kumar Gupta

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2006

Abstract

Today's manufacturing industry is mostly driven by dynamic market forces, which poses unique planning and scheduling challenges, especially in the semiconductor manufacturing. The real-life scheduling problem requires the decision maker to simultaneously consider a number of objectives before arriving at any decision. These scheduling objectives are often conflicting in nature. The trade-off involved in considering conflicting objectives provides useful insights to the decision maker.

The objective of this research is to characterize the control curves for the multi-objective dynamic scheduling (MODS) of serial and batch manufacturing processes. The pairs of conflicting objectives considered in this thesis are (1) mean cycle time and maximum tardiness, (2) mean cycle time and cycle time variance, and (3) earliness and tardiness related objectives. The motivation for using these conflicting objectives is to simultaneously incorporate the manufacturer's as well as the customer's concerns in scheduling. The interest of the decision maker lies not in just achieving a single optimal solution, but in defining a trade-off curve so that the decision maker can operate the scheduling along the trade-off curve according to the dynamic needs of the manufacturing shop.

To address the needs of conflicting scheduling objectives, the concept of Pareto control has been adopted and applied for MODS environment. For scheduling dynamically arriving jobs on a single machine, at every decision instance in the simulation clock, a Pareto optimal job is selected for processing by using compromise program-

ming. Selection of these Pareto jobs at each decision instance leads to a near-Pareto optimal schedule of the jobs in terms of overall objectives of scheduling. By changing weights of individual objectives at Pareto job selection level, the characterization curves for Pareto control have been generated and a trade-off among the conflicting objectives is established.

In computer integrated manufacturing environment, such as found in semiconductor industry, the real-time information about the incoming lots in near-future is usually available. Exploiting this look-ahead information, a look ahead batching (LAB) method is proposed and developed to minimize the earliness-tardiness (ET) performance. The results of LAB method for ET performance show significant improvements over the existing batch scheduling heuristics such as MBS, DBH and NACH. Furthermore, the ET performance is considered as the simultaneous minimization of earliness and tardiness related objectives, which are two individual conflicting objectives. Thus, the LAB method is extended to incorporate the concept of Pareto control between these conflicting objectives.

To present the relevance of the research findings in industrial application, a case study on the photolithography cluster in semiconductor wafer fabrication has been developed for the concept of Pareto control in serial processes. The results obtained from Pareto control approach are superior to the simulated results of actual operating heuristics in the factory. The characterization of Pareto control curves gives the ability to control the scheduling objectives within a specified range. By selecting the appropriate parameters, the decision maker can control the machine scheduling within the specific range of the scheduling objectives. Thus, the Pareto control curves can be used as a tool for the shop floor control management.

Acknowledgements

Towards the successful completion of this thesis, I express sincere thanks to my thesis advisor, Dr. A. I. Sivakumar, for freely encouraging my ideas, providing ample resources and guidance, and helping me develop independent research capabilities. Thanks are due to Nanyang Technological University (NTU) and Singapore Millennium Foundation (SMF) for providing the necessary financial support.

I express my heartfelt gratitude and eternal indebtedness to my teachers Dr. and Mrs. P. V. Krishnan (IIT Delhi, India) for showing me the right vision and for giving me the proper motivation and dedication for this study. I have tremendously benefitted by Dr. Krishnan's teachings and books, "Wake Up! Intellectuals" and "Science and Nescience" in terms of constructive vision, personal habits, dedication to work and sense of responsibility in life.

I am also indebted to Dr. Ankush Mittal for showing by his example how to do active research for the right cause with definite goals and plans, to Dr. Akshay Naidu for providing a wonderful friendly association, to Dr. Sumit Gupta for teaching me MatLab to carry out necessary programming and to Dr. Ramesh and Dr. Manoranjan for proof reading the papers and thesis. Special thanks are due to all my friends, Sujoy, Anjan, Shashi, Pankaj, Manish, Sumeet and Vipin, for always helping me at good and bad time during PhD studies.

Thanks are also due to Dr. Viswanath for his friendly research guidance and encouragement, to the research group namely, Dr. Mathirajan, Qi Chao, Li Kunpeng

and Ang Tee Hwee, and to the lab technician, Ms. Leong Yuit Lin, for her generous support in offering necessary facilities at Center for Supply Chain Management.

Sincere thanks are due to Lieu Sip Khean, Madan Mohan Chakravarthi and Tan Hiap Keong from the Industrial Engineering Department of Chartered Semiconductor Manufacturing Ltd., Singapore, for their full support in building the real-life case study, which has added to the practical relevance of the thesis work.

Lastly, it wouldn't be complete without deeply thanking my loving parents for fully supporting my higher studies and also my uncle and other relatives who sincerely wished for my well being in all phases of life.

Amit Kumar Gupta
NTU, Singapore.

DEDICATED

WITH

A SIMPLE AND GRATEFUL HEART

TO

MY LOVING PRECEPTORS,

THEIR MATCHLESS EXAMPLES

AND THEIR GLORIOUS VISION

Contents

Abstract	i
Acknowledgements	iii
Contents	vi
List of Figures	x
List of Tables	xiv
List of Acronyms	xvi
List of Symbols	xvii
List of Publications	xix
1 Introduction	1
1.1 Scheduling Framework and Notation	1
1.2 Pareto Optimal Solution in Multi-Objective Scheduling	3
1.3 Research Motivation	7
1.4 Research Objectives	10
1.5 Thesis Organization	12
2 Literature Review and Methodology	13

2.1	Multi-Objective Scheduling	15
2.1.1	Minimizing mean flowtime and maximum tardiness	16
2.1.2	Minimizing mean and variation of flowtime	19
2.1.3	Minimizing earliness-tardiness (ET) performance	22
2.2	Scheduling of a Batch Processing Machine	25
2.2.1	Burn-in oven model of batching	25
2.2.2	Furnace model of batching	26
2.3	Methodology: Pareto Control in Multi-Objective Dynamic Scheduling .	29
2.3.1	Determination of Pareto optimal job	33
2.3.2	Convex combination approaches	34
2.4	Conclusion	41
3	Pareto Control in Serial Processing	42
3.1	Introduction	42
3.2	Model, Assumptions and Constraints	43
3.3	Scheduling Objectives	45
3.4	Computational Experiments for Pareto Control in Serial Processing . .	46
3.5	Scheduling Criterion Selection Using Single Objective	51
3.6	Results of Pareto Control in Serial Processing	57
3.6.1	Minimizing mean cycle time ($\bar{F}$) and maximum tardiness (T_{max})	57
3.6.2	Minimizing mean cycle time ($\bar{F}$) and cycle time variance (σ_F^2) .	67
3.7	Effects of Parameter Variation on Pareto Control in Serial Processing .	73
3.7.1	Effect of due date tightness	73
3.7.2	Effect of non-linear weights	75
3.7.3	Effect of number of replications	76
3.8	Conclusion	83

4	Pareto Control in Batch Processing	85
4.1	Introduction	85
4.2	Model, Assumptions and Constraints	87
4.3	Batch Scheduling Methods	89
4.4	Computational Experiments for Pareto Control in Batch Processing . .	92
4.5	Results of Pareto Control in Batch Processing	95
4.6	Effects of Parameter Variation on Pareto Control in Batch Processing .	100
4.7	Conclusion	105
5	Single Objective Look Ahead Batch Processing	107
5.1	Introduction	107
5.2	Existing LAB Methods for Cycle Time Performance	109
5.3	Proposed LAB method for ET Performance	112
5.4	Model, Assumptions and Parameters	117
5.5	Results of Single Objective Look Ahead Batch Processing	121
5.6	Conclusion	125
6	Pareto Control in Look Ahead Batch Processing	126
6.1	Introduction	126
6.2	Earliness-Tardiness (ET) Performance	127
6.3	Pareto Control using LAB	128
6.4	Computational Experiments for Pareto Control using LAB	131
6.5	Results of Pareto Control in Look Ahead Batch Processing	136
6.6	Effects of Parameter Variation on Pareto Control using LAB	141
6.6.1	Effect of traffic intensity	141
6.6.2	Effect of due date tightness	143

6.6.3	Effect of due date window	146
6.7	Conclusion	146
7	Industrial Application: A Case Study on Photolithography Cluster	148
7.1	Relevance of Research Findings in Industrial Application	148
7.2	An Overview of Semiconductor Manufacturing	151
7.3	Case Study Model, Data and Experimentation	152
7.4	Case Study Results and Discussion	155
7.4.1	Mean cycle time and Maximum tardiness	156
7.4.2	Mean cycle time and Cycle time variance	156
7.5	Case Study with Three Objectives	159
8	Conclusions and Recommendations	163
8.1	Originality of Research	163
8.2	Research Contributions	164
8.3	Conclusions and Findings	165
8.4	Limitations of Present Research	168
8.5	Recommendations for Future Work	170
	References	172
	Appendices	182
A	Diversity of Scheduling	182
B	Duncan's Multiple Range Test	187
C	Output Data Tables	190

List of Figures

1.1	Pareto optimal solution	5
1.2	Graph showing all pairs of $(\sum_{i=1}^4 F_i, T_{max})$ for possible sequences	6
2.1	Literature review framework for multi-objective dynamic scheduling of manufacturing processes	14
2.2	Pareto control in multi-objective dynamic scheduling	32
2.3	Graph showing all jobs in Table 1.2 with points (d_i, p_i)	34
2.4	Utopia and Nadir points in compromise programming	37
3.1	Schematic diagram of the model with serial processing machine	44
3.2	Steady state for cumulative average cycle time at 0.75 utilization level .	49
3.3	Flow chart for simulation experiments on serial processing machine . .	50
3.4	Pareto control for Mean Cycle Time and Maximum Tardiness ($\rho = 0.75$)	59
3.5	Pareto control for Mean Cycle Time and Maximum Tardiness ($\rho = 0.85$)	60
3.6	End points of CP for Mean Cycle Time and Maximum Tardiness ($\rho = 0.75$)	64
3.7	Pareto control for Mean Cycle Time and Cycle Time Variance ($\rho = 0.75$)	69
3.8	Pareto control for Mean Cycle Time and Cycle Time Variance ($\rho = 0.85$)	70
3.9	End points of CP for Mean Cycle Time and Cycle Time Variance ($\rho = 0.75$)	72
3.10	Effect of tight vs. loose due-dates on Pareto control ($\rho = 0.85$)	74
3.11	Relative weight patterns	76
3.12	Effect of non-linear weights on Pareto control ($\rho = 0.85$)	77

3.13	Effect of varying the number of replications ($\rho = 0.75$)	79
3.14	Effect of varying the number of replications ($\rho = 0.85$)	81
4.1	Schematic diagram of the model with batch processing machine	87
4.2	Natural Batch Scheduling (NBS) rule	90
4.3	Neuts's Minimum Batch Size (MBS) rule	91
4.4	Steady state of cumulative average cycle time with MBS 1 scheduling strategy and CP method with unity power at 0.60 utilization level . . .	94
4.5	Mean Cycle Time vs. Maximum Tardiness for dynamic scheduling of a batch processing machine ($\rho = 0.5$, MBS = 1)	96
4.6	Mean Cycle Time vs. Maximum Tardiness for dynamic scheduling of a batch processing machine ($\rho = 0.6$, MBS = 1)	97
4.7	Mean Cycle Time vs. Maximum Tardiness for dynamic scheduling of a batch processing machine ($\rho = 0.7$, MBS = 1)	98
4.8	Effect of ρ variation on range of Pareto control between Mean Cycle Time and Maximum Tardiness	101
4.9	Effect of MBS value variation on Pareto control between Mean Cycle Time and Maximum Tardiness for dynamic scheduling of a batch processing machine for CP1 at $\rho = 0.5$	102
4.10	Effect of MBS value variation on Pareto control between Mean Cycle Time and Maximum Tardiness for dynamic scheduling of a batch processing machine for CP1 at $\rho = 0.6$	103
4.11	Effect of MBS value variation on Pareto control between Mean Cycle Time and Maximum Tardiness for dynamic scheduling of a batch processing machine for CP1 at $\rho = 0.7$	104
5.1	Glassey and Weng's Dynamic Batching Heuristic (DBH) algorithm . .	109
5.2	Cumulative number of lots in queue through time	110

5.3	Flowchart of simulation logic of Fowler et al.'s Next Arrival Control Heuristic (NACH) algorithm	112
5.4	Flow chart of the simulation logic of LAB algorithm for ET performance	114
5.5	An example of constructing look ahead batching (LAB) scenarios . . .	116
5.6	Schematic diagram of the experimental model	117
5.7	Steady state simulation results of cumulative average cycle time in dy- namic batch scheduling ($\rho = 0.7$).	120
5.8	Column graphs for % reduction in (a) average sum of absolute earliness and tardiness $\overline{(E + T)}$ and (b) average square sum of earliness and tardiness $\overline{E^2 + T^2}$	124
6.1	Due date window concept	127
6.2	Look Ahead Batching (LAB) scenarios	129
6.3	Flow chart of the simulation experiment for the dynamic scheduling of a batch processing machine	134
6.4	Steady state simulation results of cumulative average cycle time in dy- namic batch scheduling at two levels of traffic intensities: (a) $\rho = 0.5$ and (b) $\rho = 0.7$	135
6.5	Performance of LAB method over MBS, DBH and NACH methods for schedule control on two objectives (a) average earliness <i>vs.</i> average tar- diness and (b) number of early jobs <i>vs.</i> number of tardy jobs. (Experi- mental settings : $\rho = 0.7$, DD1, DDW1)	138
6.6	Performance of LAB method over MBS, DBH and NACH methods for two ET performance measures (a) $\overline{(E + T)}$ and (b) $\overline{E^2 + T^2}$. (Ex- perimental settings : $\rho = 0.7$, DD1, DDW1)	140

6.7	Effect of varying traffic intensity on schedule control between (a) Average earliness <i>vs.</i> Average tardiness and (b) Number of early jobs <i>vs.</i> Number of tardy jobs (Experimental settings: DD1, DDW1)	142
6.8	Effect of varying due date tightness on schedule control between (a) Average earliness <i>vs.</i> Average tardiness and (b) Number of early jobs <i>vs.</i> Number of tardy jobs (Experimental settings: $\rho = 0.7$, DDW1) . . .	144
6.9	Effect of varying due date window on schedule control between (a) Average earliness <i>vs.</i> Average tardiness and (b) Number of early jobs <i>vs.</i> Number of tardy jobs (Experimental settings: $\rho = 0.7$, DD1)	145
7.1	A simplified semiconductor manufacturing flow	151
7.2	Flow of processing steps in a photolithography cluster	153
7.3	Case study: Pareto curves for mean cycle time and maximum tardiness	157
7.4	Case study: Pareto curves for mean cycle time and cycle time variance	158
7.5	Case study: two different projection views of the Pareto surface for the three objective Pareto control case	161
7.6	Case study: orthographic projection views of the Pareto surface for the three objective Pareto control case	162
A.1	Diversity of scheduling	183
B.1	Results of ANOVA test for RDI values of Mean Cycle Time, Cycle Time Variance and Max. Tardiness at $\rho = 0.75$ from SPSS 10 output window	188
B.2	Results of ANOVA test for RDI values of Mean Cycle Time, Cycle Time Variance and Max. Tardiness at $\rho = 0.85$ from SPSS 10 output window	189

List of Tables

1.1	Typical scheduling objectives	3
1.2	A static bi-objective scheduling problem (French, 1982)	5
1.3	Pareto optimal solutions for the static bi-objective scheduling problem .	6
3.1	Factors for simulation experiments	47
3.2	Mean cycle time performance of various priority indices at $\rho = 0.75$. .	52
3.3	RDI of various priority indices for Mean Cycle Time performance ($\rho =$ 0.75)	53
3.4	Steady state results of RDI and Duncan's test for Mean Cycle Time ($\bar{F}$)	55
3.5	Steady state results of RDI and Duncan's test for Max. Tardiness (T_{max})	55
3.6	Steady state results of RDI and Duncan's test for Cycle Time Variance (σ_F^2)	56
3.7	Mean cycle time and maximum tardiness values for $w = 0.45, 0.50$ and 0.55 points on CP1 curve in Figure 3.4	62
3.8	Paired t -test for mean cycle time and maximum tardiness values between the two pairs of consecutive points on CP1 curve in Figure 3.4	63
4.1	Factors for simulation experiments	92
5.1	Factors for computational experiments	119
5.2	Steady state results of average sum of absolute earliness and tardiness .	122
5.3	Steady state results of average square sum of earliness and tardiness . .	123

6.1	Factors for computational experiments	132
7.1	Sample presentation of the factory data	154
C.1	Results of Pareto control in serial processing for Mean Cycle Time and Maximum Tardiness ($\rho = 0.75$)	190
C.2	Results of Pareto control in serial processing for Mean Cycle Time and Maximum Tardiness ($\rho = 0.85$)	191
C.3	Results of Pareto control in serial processing for Mean Cycle Time and Cycle Time Variance ($\rho = 0.75$)	192
C.4	Results of Pareto control in serial processing for Mean Cycle Time and Cycle Time Variance ($\rho = 0.85$)	193

List of Acronyms

CIM	:	Computer Integrated Manufacturing
CP	:	Compromise Programming
CSS	:	Conjunctive Simulated Scheduling
DBH	:	Dynamic Batching Heuristic
EDD	:	Earliest Due Date
ET	:	Earliness-Tardiness
FCFS	:	First-Come-First-Serve
IC	:	Integrated Circuits
JIT	:	Just-in-Time
LAB	:	Look Ahead Batching
MBS	:	Minimum Batch Size
MODS	:	Multi-Objective Dynamic Scheduling
MOO	:	Multi-Objective Optimization
NACH	:	Next Arrival Control Heuristic
NBS	:	Natural Batch Scheduling
RDI	:	Relative Deviation Index
SPT	:	Shortest Processing Time
WA	:	Weighted Average
WIP	:	Work-In-Process

List of Symbols

n	Number of jobs
i	Index for the jobs
j	Index for the scheduling objectives and their criteria
J_i	i^{th} job
p_i	Processing time of J_i
r_i	Release date of J_i
d_i	Due date of J_i
w_i	Weight associated with J_i : $\sum_{i=1}^n w_i = 1$
t	Decision instance in simulation clock
W_i	Waiting time of J_i ($= t - r_i$)
C_i	Completion time of J_i
F_i	Cycle time (flowtime) of J_i ($= C_i - r_i$)
L_i	Lateness of J_i ($= C_i - d_i$)
E_i	Earliness of J_i ($= \max(0, -L_i)$)
T_i	Tardiness of J_i ($= \max(0, L_i)$)
U_{Ei}	Early job indicator ($= 1$, if $C_j < d_j$; else 0)
U_{Ti}	Tardy job indicator ($= 1$, if $C_j > d_j$; else 0)
$\bar{C}$	Mean completion time
$\bar{C}^w$	Mean weighted completion time
$\bar{F}$	Mean cycle time
$\bar{F}_i$	Mean cycle time of jobs in the system before J_i
σ_F^2	Cycle time variance
T_{max}	Maximum tardiness

$\bar{T}$	Mean tardiness
$\bar{E}$	Mean earliness
N_T	Number of tardy jobs
N_E	Number of early jobs
k	Due-date allowance factor
λ	Inter-arrival rate
μ	Service rate
ρ	Traffic intensity or utilization level ($= \lambda/\mu$)
F_j	Scheduling objective function
f_j	Scheduling criterion function
w_j	Weight associated with the scheduling criterion (f_j)
r	Exponent value in compromise programming method
$F(i)$	WA function value for i^{th} job
$F_r(i)$	CP function value for i^{th} job
c	Batch capacity
T	Batch processing time
t_0	The time epoch that the batching machine is idle with positive queue length.
t_i	The arriving epoch of the next i^{th} lot after t_0
q	The number of lots in queue at epoch t_0 ($q > 0$)
L	The look-ahead number (assuming that the next L arrival epochs are known with certainty at t_0 .)

List of Publications

Publications in International Journals

1. Sivakumar, A.I. and Gupta, A.K. (2006), "On-line multi-objective Pareto optimal dynamic scheduling of semiconductor back-end using conjunctive simulated scheduling", *IEEE Transactions on Electronics Packaging Manufacturing*, Publisher: IEEE, Vol. 29, No. 2, 2006, pp 99-109.
2. Gupta, A.K. and Sivakumar, A.I. (2006), "Controlling delivery performance in semiconductor manufacturing using look ahead batching", Accepted in a special issue of *International Journal of Production Research*.
3. Gupta, A.K. and Sivakumar, A.I. (2006), "Optimization of due date objectives in scheduling semiconductor batch manufacturing", *International Journal of Machine Tools and Manufacture*, Publisher: Elsevier Science, Vol. 46, No. 12-13, 2006, pp 1671-1679.
4. Gupta, A.K. and Sivakumar, A.I. (2006), "Job shop scheduling techniques in semiconductor manufacturing", *International Journal of Advanced Manufacturing Technology*, Publisher: Springer-Verlag London Ltd., Vol. 27, Nos. 11-12, 2006, pp 1163-1169.
5. Gupta, A.K. and Sivakumar, A.I. (2005), "Multi-objective scheduling of two job families on a single machine", *OMEGA: The International Journal of Management Science*, Publisher: Elsevier, Vol. 33(5), October 2005, 399-405.
6. Gupta, A.K. and Sivakumar, A.I. (2005), "Conjunctive simulated scheduling", *International Journal of Advanced Manufacturing Technology*, Publisher: Springer-Verlag London Ltd., Vol. 26, Nos. 11-12, 2005, pp 1409-1413.
7. Gupta, A.K. and Sivakumar, A.I. (2005), "Single machine scheduling with multiple objectives in semiconductor manufacturing", *International Journal of Advanced Manufacturing Technology*, Publisher: Springer-Verlag London Ltd., Vol. 26, Nos. 9-10, 2005, pp 950-958.
8. Ganesan, V.K., Gupta, A.K. and Sivakumar, A.I. (2004), "Cycle time variance minimization in dynamic scheduling of single machine systems", *Computers and Industrial Engineering* (submitted, Ref: 04/096).

Publications in International Conferences

9. Gupta, A.K. and Sivakumar, A.I. (2006), "Pareto control in multi-objective dynamic scheduling of a stepper machine in semiconductor wafer fabrication," 2006 Winter Simulation Conference (WSC 2006), Dec. 3-6, 2006, Monterey, CA, USA.
10. Gupta, A.K. and Sivakumar, A.I. (2005), "On-time Delivery Pareto Controllability for Batch Processing in Semiconductor Manufacturing", Proc. of the *3rd International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM 2005)*, Singapore, 6-7 Oct, 2005, 13-20.
11. Gupta, A.K., Sivakumar, A.I., and Ganesan, V.K. (2004), "Look ahead batching to minimize earliness/tardiness related objectives in batch processes", Proc. of the *IEEE Conference on Robotics, Automation and Mechatronics (RAM-2004)*, Singapore, 1-3 Dec. 2004, 1101-1106.
12. Ganesan V.K., Gupta, A.K., and Sivakumar, A.I. (2004), "Bi-objective schedule control of batch processes in semiconductor manufacturing", Proc. of the *IEEE Conference on Robotics, Automation and Mechatronics (RAM-2004)*, Singapore, 1-3 Dec. 2004, 1077-1082.
13. Gupta, A.K., Ganesan V.K., and Sivakumar, A.I. (2004), "Hot lot management: minimizing cycle time in batch processes", Proc. of the *IEEE International Engineering Management Conference (IEMC-2004)*, Singapore, 18-21 Oct. 2004, 1217-1221.
14. Ganesan V.K. Gupta, A.K., and Sivakumar, A.I. (2004), "Bi-objective Pareto-schedule generation and control in semiconductor supply chain", Proc. of the *IEEE International Engineering Management Conference (IEMC-2004)*, Singapore, 18-21 Oct. 2004, 1212-1216.
15. Gupta, A.K., Sivakumar, A.I. and Ganesan V.K. (2004), "Cycle time reduction in batch processes using look-ahead batch scheduling", Proc. of the *33rd International Conference on Computers and Industrial Engineering (ICCIE 2004)*, Jeju, Korea, CIE 253. **[Received BEST PAPER AWARD AND PRIZE]**
16. Gupta, A.K. and Sivakumar, A.I. (2002), "Simulation based multi-objective schedule optimization in semiconductor manufacturing", Proc. of the *2002 Winter Simulation Conference (WSC 2002)*, San Diego, California, USA 1862-1870.
17. Gupta, A.K., Sivakumar, A.I., and Sarawgi, S. (2002), "A shop floor control module with simulation based proactive decision support", Proc. of the *2002 Winter Simulation Conference (WSC 2002)*, San Diego, California, USA 1897-1902.
18. Gupta, A.K. and Sivakumar, A.I. (2002), "Approaches to multi-objective scheduling optimization in semiconductor back-end", Proc. of the *International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM 2002)*, April 10-12, 2002, Tempe, Arizona, USA, 223-228.

Chapter 1

Introduction

Today's manufacturing industry is mostly driven by dynamic market forces, which poses unique planning and scheduling challenges, especially in the semiconductor manufacturing. To meet the changing market needs, scheduling of manufacturing systems is considered one of the prime options. The real-life scheduling problem requires the decision maker to simultaneously consider a number of objectives before arriving at any decision. These scheduling objectives are often conflicting in nature and their relative importance varies with market needs. In such situations, the trade-off involved in considering conflicting objectives provides useful insights to the decision maker. The interest of the decision maker lies not in just achieving a single optimal solution, but in defining a trade-off curve so that the decision maker can operate the scheduling along the trade-off curve according to the dynamic needs of the manufacturing shop. Thus, considering problems with multiple objectives is more relevant in the context of real-life manufacturing scheduling, especially in the semiconductor industry.

1.1 Scheduling Framework and Notation

The vast diversity of the scheduling field has been described in detail in Appendix A. Here, a brief definition of scheduling is being presented as:

“Scheduling concerns the allocation of limited resources to tasks over time. It is a decision-making process that has as a goal the optimization of one or more objectives.”

– Pinedo (1995)

The standard representation for scheduling problems (Graham et al., 1979) is $\Psi_1|\Psi_2|\Psi_3$, where Ψ_1 indicates the scheduling environment, Ψ_2 describes the job and family characteristics and any restrictive requirements, and Ψ_3 defines the objective function to be minimized. For this, the basic notations of job characteristics have already been presented in the list of symbols. Let $\Psi_1 = \alpha m$ (Potts and Kovalyov, 2000), where m is the number of machines, $\alpha \in \{P, Q, R, F, FF, J, FJ, O\}$ respectively for classical machines in identical, uniform and unrelated parallel machine, flow shop, flexible flow shop, job shop, flexible job shop and open shop environments. In the case of a single machine, α is omitted, and $m = 1$ denotes a single machine. Field Ψ_2 specifies the processing restrictions and constraints such as release dates (r_i), sequence dependent setup times (s_{ij}), batch processing (*batch*), preemption (*prmp*), precedence constraints (*prec*), breakdowns (*brkdown*), recirculation (*recrc*), etc. (Pinedo, 1995).

The scheduling objectives are considered in the last field Ψ_3 . Multi-objective scheduling problems are represented by having multiple entries in Ψ_3 (T’kindt and Billaut, 2002). The need of scheduling with multiple objectives is further discussed in Section 1.3. In dynamic scheduling of a single machine, the jobs go on arriving dynamically at the machine for processing, but the simulation statistics are collected only in the steady state situation for a fixed number of jobs. If n is the number of jobs considered in steady-state simulation statistics, then the scheduling objectives are defined based on these n jobs. Typical objectives in dynamic scheduling are presented in Table 1.1.

Throughout the thesis, the term ‘scheduling objective’ is used in an overall sense, while the term ‘scheduling criterion’ is used in a myopic sense, *i.e.*, at each decision instance in simulation clock. For example, in single machine static scheduling prob-

Table 1.1: Typical scheduling objectives

$\bar{C} = \sum_{i=1}^n C_i/n$	mean completion time
$\bar{C}^w = \frac{1}{n} \sum_{i=1}^n (w_i \cdot C_i)$	mean weighted completion time
$\bar{F} = \sum_{i=1}^n F_i/n$	mean cycle time
$\sigma_F^2 = \sum_{i=1}^n (F_i - \bar{F})^2/n$	cycle time variance
$T_{max} = \max(T_1, T_2, \dots, T_n)$	maximum tardiness
$\bar{T} = \sum_{i=1}^n T_i/n$	mean tardiness
$\bar{E} = \sum_{i=1}^n E_i/n$	mean earliness
$N_T = \sum_{i=1}^n U_{Ti}$	number of tardy jobs
$N_E = \sum_{i=1}^n U_{Ei}$	number of early jobs

lem, the scheduling objective of minimizing mean flowtime ($\bar{F}$) is achieved by using the scheduling criterion of shortest processing time (SPT), $\min(p_i)$, at each selection instance. Similarly, the objective of minimizing maximum tardiness (T_{max}) is obtained by applying earliest due date (EDD), $\min(d_i)$, scheduling criterion.

1.2 Pareto Optimal Solution in Multi-Objective Scheduling

The multi-objective scheduling problem is a class of multi-objective optimization (MOO) problem. A MOO problem can be denoted as (Sawaragi et al., 1985; Yu, 1985):

$$\min F(x) \triangleq (f_1(x), f_2(x), \dots, f_P(x)), \forall x \in X \quad (1.1)$$

where each $f_j(x), j = 1, \dots, P$ is a scalar objective function. An optimal solution is the one, which attains the minimum value of all the objectives simultaneously. The

solution x^* is optimal to the problem defined if and only if $x^* \in S$ and

$$f_j(x^*) \leq f_j(x), \forall j \text{ and } \forall x \in S \quad (1.2)$$

where S is the feasible region. Unlike the single objective optimization problem, the definition of an optimal solution for a MOO problem is not straightforward because a solution, which simultaneously minimizes all the objectives, rarely occurs. As in most of the real-life MOO problems, the objective functions are found conflicting in nature, there may not be a particular optimal solution to a MOO problem. Therefore, a more general definition of optimality is required, which is called as Pareto optima as introduced by an Italian economist, Vilfredo Pareto, in 1896.

For a minimization type of MOO problem, as denoted in Equation 1.1, a Pareto optimal solution is one in which no decrease can be obtained in any of the objectives without causing a simultaneous increase in at least one of the other objectives. A Pareto optimal solution is also called as efficient, non-dominated, or non-inferior solution and is defined mathematically as (Tabucanon, 1988):

Definition:

The solution x^ is Pareto Optimal to the problem defined if and only if there does not exist any $x \in S$ such that $f_j(x) \leq f_j(x^*) \forall j$ and $f_j(x) < f_j(x^*)$ for at least one j .*

Suppose that there are two objective functions, $f_1(x)$ and $f_2(x)$, where $x \in X$. A single objective problem can be formulated as $\min Z \triangleq (f_1(x), f_2(x))$. For a point $x_1 \in X$, $Z_1 \triangleq (f_1(x_1), f_2(x_1))$ can be calculated and plotted on the Cartesian coordinates as shown in Figure 1.1. For simultaneous minimization of both f_1 and f_2 , point Z_1 resulting from x_1 is certainly not the choice as point Z_2 is better than point Z_1 in terms of both f_1 and f_2 .

Pareto Optimal or efficient solutions are defined as the boundary line of solutions that are better solutions than the others in the operating region. An operating solution $x_0 \in X$ is a Pareto optimal solution if no other operating point $x \in X$ exists such that

Figure 1.1: Pareto optimal solution

$f(x) \leq f(x_0)$ and this implies that there won't be any other operating solution x_a such that $f_1(x_a) \leq f_1(x_0)$ and $f_2(x_a) \leq f_2(x_0)$. In this example, point Z_0 can be considered as a Pareto optimal point since there is no solution in the shaded area.

To clarify the concept of Pareto optimal solutions in the field of static scheduling, an example of a bi-objective scheduling problem with four jobs on a single machine is presented in Table 1.2, which is first considered by VanWassenhove and Gelders (1980) and later reproduced by French (1982). In this static problem, the two conflicting objectives are minimizing total cycle time and minimizing maximum tardiness. Job data includes the processing times and the due dates for four jobs.

Table 1.2: A static bi-objective scheduling problem (French, 1982)

Objective function: $\min Z \triangleq (\sum_{i=1}^4 F_i, T_{max})$					
Job data:	Job	J_1	J_2	J_3	J_4
	Processing time (p_i)	2	4	3	1
	Due date (d_i)	1	2	4	6

For this static scheduling problem, $f_1(x) = \sum_{i=1}^4 F_i$, $f_2(x) = T_{max}$ and the total number of feasible solutions are $4!$ ($=24$) sequences, which are presented in graphical

form on the two objectives in Figure 1.2. There are only 3 sequences which satisfy the Pareto optimality conditions and hence these are called as Pareto Optimal solutions or Pareto sequences, which are presented in Table 1.3.

Figure 1.2: Graph showing all pairs of $(\sum_{i=1}^4 F_i, T_{max})$ for possible sequences. Encircled points present Pareto optimal solutions/sequences with respect to $\min(\sum_{i=1}^4 F_i, T_{max})$.

Table 1.3: Pareto optimal solutions for the static bi-objective scheduling problem

Pareto optimal solutions/sequences		
Sequence :	$\sum_{i=1}^4 F_i$	T_{max}
$(J_4, J_1, J_3, J_2) :$	20	8
$(J_4, J_1, J_2, J_3) :$	21	6
$(J_1, J_2, J_3, J_4) :$	27	5

However, in case of multi-objective dynamic scheduling (MODS) problems, the concept of Pareto optimality is not yet clearly defined due to stochastic nature of job arrivals. In spite of the steady growth of research work in the area of multi-objective scheduling, most of the published work only presents results on finding Pareto-optimal

solutions for static scheduling problems, in which it is assumed that all jobs are available for scheduling at the start of the planning period and their processing information is known.

1.3 Research Motivation

The present research on Multi-Objective Dynamic Scheduling (MODS) is motivated primarily to deal with multiple conflicting objectives, dynamic scheduling, presence of serial and batch processes and availability of look-ahead information of jobs in computer integrated manufacturing (CIM) environment. These are described in detail in the following subsections.

Need of scheduling with multiple objectives

A major portion of the past scheduling research deals mainly with single objective scheduling. Though, a vast amount of literature exists on the optimization of various single objective functions such as Makespan time, average cycle time, machine utilization and tardiness. Only a little effort has been made in the direction of simultaneously considering multiple objectives while scheduling, which may very well be contradicting each other in nature (T'kindt and Billaut, 2002). In the field of semiconductor scheduling, the contradicting needs of the multiple objectives were taken into consideration only in a few research papers (Yang and Chang, 1998; Dabbas et al., 2001; Sivakumar, 2001). Therefore, the primary focus of this research is to develop scheduling methodology to deal with the contradicting multiple objectives, which can be applied to the serial as well as batch manufacturing processes. Pinedo and Chao (1999) also proposed “scheduling with multiple objectives”, as a new direction of research in the field of scheduling as it may often be desirable to see the trade-offs between the different objectives of scheduling.

Pareto trade-off between conflicting scheduling objectives

In manufacturing scheduling, following objectives are of high importance from the perspective of customer and producer:

- Minimizing mean cycle time in order to reduce the work-in-process (WIP) inventory and the cost of production.
- Minimizing variance of cycle time for providing more reliable predictions of completion times to the customers.
- Minimizing tardiness for meeting due dates and reducing the penalty cost arisen out of unfulfilled demand.
- Minimizing earliness for reducing the inventory holding cost arisen out of early finished products.

These objectives constitute the following pairs of conflicting scheduling objectives:

- Minimizing mean cycle time and minimizing maximum tardiness (VanWassenhove and Gelders, 1980; Koksalan, 1999)
- Minimizing mean cycle time and variance of cycle time (Bagchi, 1989)
- Minimizing both earliness and tardiness related objectives, *i.e.*, ET performance in Just-in-Time (JIT) manufacturing environment (Koulamas, 1996; Ventura and Radhakrishnan, 2003)

Thus, the pairs of conflicting objectives studied in this thesis are mean cycle time and maximum tardiness, mean cycle time and cycle time variance, average earliness and average tardiness and number of early jobs and number of tardy jobs. For each such pair, the interest lies not in just achieving a single optimal solution, but in defining a trade-off curve so that the decision maker can operate the scheduling along the trade-off curve according to the dynamic needs of the manufacturing shop. Thus, there is a need to develop a Pareto control between these conflicting objectives.

Requirements of dynamic scheduling

From more than six decades, efforts have been invested by many researchers around the world towards the development of polynomial algorithms for solving complex scheduling problems. Exact algorithms are developed even for multi-objective scheduling problems as reviewed by Nagar et al. (1995) and T'kindt and Billaut (2001). However, most of these efforts are dedicated only to static scheduling problems, where all the jobs are available from the beginning. Only some effort has been directed towards multi-objective scheduling problems in dynamic environment such as Dabbas et al. (2001); Dabbas and Fowler (2003); Sivakumar (2001); *etc.* There is requirement of research efforts to develop scheduling methods for MODS environment in both serial as well as batch processes.

As most of the scheduling problems are intrinsically NP-Hard (Pinedo, 1995), no simple rules or algorithms yield optimal solutions in a limited amount of computer time. Even after formulating as mathematical programs, solving these NP-Hard problems to optimality may require an enormous amount of computer time at best and may not be achievable in most cases. Therefore, in the absence of such large computing power, one is usually satisfied with an “acceptable” feasible solution that presumably is not far from optimal called as “Near-Optimal Solutions” (Pinedo and Chao, 1999). One way to obtain these near-optimal solutions of the NP-hard optimization problem is to develop a detailed and deterministic simulation model of the entire system. This simulation-based scheduling method simply ignores the NP-hard issue and efficiently generates near-optimal solutions for dynamic scheduling problems (Hopp and Spearman, 2000).

Need of exploiting look-ahead information

With the recent influence of CIM environment, the highly sophisticated computerized shop floor control systems can provide reasonably accurate predictions of arrival times

and due dates of future jobs to a station. This real-time information about the incoming jobs in the near-future is seldom exploited in decision making. However, this look-ahead information can make a large difference in the scheduling performance, especially for batch processing workstations such as oxidation and diffusion furnaces in semiconductor wafer fabs. This can be highly useful in scheduling a batch process in supporting the decision of whether or not to start processing a batch.

In the existing literature, the future arrival information has been exploited by several researchers in order to minimize cycle time in batch processes such as Glassey and Weng (1991); Fowler et al. (1992); Robinson et al. (1995) and Fowler et al. (2000). There is also a need to exploit the due date information of the future arrivals for improving the ET performance (earliness and tardiness related objectives) in scheduling the batch processes. Further, this information can also be utilized to develop a trade-off between earliness and tardiness related objectives in JIT manufacturing environment.

1.4 Research Objectives

The main objective of this thesis is to explore the field of multi-objective dynamic scheduling in manufacturing environment. The primary focus is to characterize and analyze the nature of trade-off curves between the conflicting objectives in dynamic scheduling of a single machine with serial or batch processing. The secondary focus is on studying the effects of using look ahead information for improving ET performance in batch processing and further to develop a trade-off between earliness and tardiness related objectives. To this end the specific research objectives of the thesis can be summarized as follows:

1. To adopt and apply the novel concept of Pareto control for multi-objective dynamic scheduling (MODS) problems.
2. To investigate the trade-off curve between the conflicting objectives in dynamic

scheduling of a serial, as well as batch, processing machine using the developed MODS concept.

3. To propose and validate a single objective look ahead batching method for minimizing the ET performance in scheduling a batch processing machine.
4. To investigate the trade-off curve for the Pareto control between earliness and tardiness related objectives in look ahead batching using the developed MODS concept.

The third research objective focuses on the development of a look ahead batching method for minimizing the ET performance. This is a preliminary requirement for applying the MODS concept in look ahead batch processing environment, which constitutes the fourth research objective.

The prime motivation of this research has arisen from scheduling problems in semiconductor manufacturing. Nonetheless, the developed algorithms and methods are generic in nature and therefore applicable in other industrial production scheduling environments as well. The single machine models considered in this thesis are generic to manufacturing processes; however the case study has been focused on the example of photolithographic process from the wafer fabrication. The primary focus is to develop scheduling methods for simultaneously optimizing the conflicting objectives and to control the achieved schedule on a trade-off curve. The proposed method is capable of providing the control parameters of scheduling to run the shop in a particular fashion, which results in the desired range of the objective functions. Thus, it acts as a throttle in the hands of the shop floor manager to control scheduling in favor of any objective according to the time-demands and circumstance-needs of the industry.

1.5 Thesis Organization

In this introductory chapter, the need for research in the field of multi-objective dynamic scheduling has been discussed and the research objectives have been defined. The scheduling literature relevant to this research is reviewed in Chapter 2 in two parallel directions: multi-objective scheduling and the scheduling of a batch processing machine. This chapter also presents the methodology of adopting the Pareto control concept for MODS problems, which covers the first research objective.

The concept of Pareto control is applied in dynamic scheduling of serial as well as batch processing machines respectively in Chapter 3 and Chapter 4. The trade-off curves for Pareto control are presented and analyzed for simultaneous minimization of two conflicting scheduling objectives. These chapters cover the second research objective.

Exploiting the available real-time information of incoming jobs, a look ahead batching (LAB) method is proposed in Chapter 5. LAB method is computationally experimented to minimize the ET performance in the dynamic batch scheduling, which covers the third research objective. In Chapter 6, the LAB method is extended to incorporate the concept of Pareto control between the conflicting objectives of earliness and tardiness in the dynamic scheduling of a batch processing machine. This chapter covers the last research objective.

The relevance of research findings in industrial application and a real-life case study on the photolithography process in semiconductor wafer fabrication for the concept of Pareto control in serial processing are presented in Chapter 7. Finally, the essential conclusions and contributions of this research to the field of MODS are presented in Chapter 8 along with the possible future research directions.

Chapter 2

Literature Review and Methodology

This chapter reviews the scheduling literature relevant to this research on multi-objective dynamic scheduling in serial and batch manufacturing processes. Further, it presents the adopted Pareto control concept for MODS problems. Scheduling literature is very vast and can be categorized in many ways as presented in Appendix A. For ease of literature review, a framework has been developed, which is presented in Figure 2.1. In this framework, scheduling literature is vertically categorized first based on the application type, then in to single and multiple objectives and then in to static and dynamic environment. In the horizontal direction, the scheduling literature is first categorized based on shop configuration and then based on type of job processing.

In the literature review framework, presented in Figure 2.1, a total of eight blocks are presented corresponding to single machine scheduling for manufacturing environment. Each block is filled with the existing key research papers. In the figure, two blocks are highlighted, which present the focus area of this research.

For building a research base, the existing literature is reviewed in the following sections in two parallel directions: multi-objective scheduling with serial processing and single objective scheduling with batch processing.

Figure 2.1: Literature review framework for multi-objective dynamic scheduling of manufacturing processes

2.1 Multi-Objective Scheduling

In this section, a review of multi-objective scheduling literature relevant to this thesis has been presented. The most important part of the literature on scheduling problems is dedicated to single objective problems, whereas, in practice, the use of multiple objectives often enables computation of a more realistic solution for the decision maker. Research on multi-objective scheduling has been scarce, especially when compared to the research in single objective scheduling. The complexity of scheduling problems provides a possible explanation for the lack of published research in problems involving multiple objectives. It is well known that optimal solutions can be found for only relatively small single objective problems. In fact, only two algorithms have been reported for the single machine bi-objective scheduling problem (Nagar et al., 1995). One is the pseudo-polynomial algorithm, which was developed by VanWassenhove and Gelders (1980), while the other is the polynomial algorithm developed by Chen and Bulfin (1990).

Since most of the scheduling problems even with one measure of performance are NP-hard, the multi-objective scheduling research has been confined mainly to single machine or single stage scheduling problems. Most of the research in multi-objective scheduling has been directed towards modelling situations with bi-objective on single machine, which is covered by a number of review articles. Dileepan and Sen (1988) addressed single machine, bi-objective problems involving regular functions of completion time. Fry et al. (1989) considered single machine, multi-objective problems that involved both regular and non-regular functions of completion time. They noted that even though non-regular measures are becoming more important with the increasing acceptance of JIT manufacturing philosophy, there exists a gap in the area of bi-objective research where one criterion is not regular. Nagar et al. (1995) addressed multiple and bi-objective scheduling problems and categorized the literature on a new classification scheme. However, the problems and results surveyed are restricted to regular functions

of completion time. T'kindt and Billaut (2001, 2002) provided a framework to tackle multi-objective scheduling problems according to decision aid concept. The problem is decomposed into three different problems. The first problem is about obtaining a model, the second about how to take objective into account and the third one is about solving a scheduling problem.

In this thesis, the problem of bi-objective scheduling is studied in a dynamic single machine system with three pairs of conflicting objectives - (i) the mean cycle time and the maximum tardiness, (ii) the mean cycle time and the cycle time variance, and (iii) the earliness and the tardiness related objectives. Here, cycle time and flowtime are same except that the cycle time terminology is used in dynamic scheduling problems while flowtime terminology is used in static environment. For a manufacturing process, Little's law states that the work-in-process inventory is equal to the product of throughput and cycle time (Hopp and Spearman, 2000). Therefore, minimizing mean cycle time implies minimizing work-in-process inventory, provided throughput remains constant. However, minimizing maximum tardiness can be considered as a measure for customer satisfaction. On the other hand, minimizing cycle time variance helps in reliable predictions of the completion of jobs and thus in providing the same quality of service to the customers, and thus reflecting the customer's concern. In JIT manufacturing environment, minimizing earliness reflects in minimizing inventory holding costs for the manufacturer. The motivation for studying these pairs of conflicting objectives is to simultaneously incorporate the manufacturer's as well as the customer's concerns in scheduling, which are conflicting by nature.

2.1.1 Minimizing mean flowtime and maximum tardiness

Majority of research on multi-objective scheduling has considered the minimization of job flowtime as one of the objectives. Other objectives have often been due date oriented. The extensive use of flowtime as a criterion is due to the direct relationship

2. Literature Review and Methodology

between work in process inventory and flowtime in single machine sequencing. Thus, since normal objectives of production managers include inventory reduction and customer satisfaction improvements, it is logical to incorporate flowtime and a measure of due date performance as scheduling objective.

Smith (1956) was among the first researchers to deal with a bi-objective problem of minimizing the total flowtime ($= n \times \bar{F}$) and the maximum tardiness (T_{max}). The two objectives are straightforward when taken individually. Since this is single machine static case, the total flowtime is minimized by a shortest processing time (SPT) sequence and the minimization of the maximum tardiness is achieved by an earliest due date (EDD) sequence. Smith's *a priori* formulation is stated as follows:

$$\begin{aligned} & \text{Minimize} && \sum_{i=1}^n F_i \\ & \text{subject to} && T_{max} = 0 \end{aligned} \tag{2.1}$$

where F_i is the flowtime of job i and n is the number of jobs considered in static case.

Smith's algorithm is based on a theorem which states: Given that no jobs are tardy, the last job (k) in the sequence has the following two properties:

$$\begin{aligned} & \text{(i)} \quad \sum_{i=1}^n p_i - d_k \leq 0, \text{ and} \\ & \text{(ii)} \quad p_k \geq p_i, \text{ for all } i \text{ for which } \sum_{j=1}^n p_j - d_i \leq 0, \end{aligned}$$

where p_i is the processing time and d_i is the due-date for job i .

Heck and Roberts (1972) extended Smith's result by relaxing the no tardiness condition. Instead, they placed a user specified bound on maximum tardiness and formulated the problem as follows:

$$\begin{aligned} & \text{Minimize} && \sum_{i=1}^n F_i \\ & \text{subject to} && T_{max} \leq T_o \end{aligned} \tag{2.2}$$

2. Literature Review and Methodology

Given that T_o is the maximum tardiness of the EDD sequence, mathematically the extension is equivalent to rewriting property (i) as $\sum_{i=1}^n p_i - d_k \leq T_o$. Their solution procedure is similar to that given by Smith (1956).

Emmons (1975) considered the same problem with the objective of minimizing the total flowtime ($\sum_{i=1}^n F_i$) and the maximum weighted tardiness $\max\{w_i T_i\}$. Emmons treats the problem as a preemptive goal programming problem with the primary objective of minimizing the maximum value of the cost function over all jobs. Within the set of all jobs satisfying the primary objective, the schedule to minimize flowtime was determined.

An *a priori* procedure was published by VanWassenhove and Gelders (1980) to generate the set of all efficient points for the following objective:

$$\text{Minimize } \{\bar{F} \text{ and } T_{max}\} \quad (2.3)$$

(For a bi-objective minimization problem involving objective C_1 and C_2 , a point i is said to be Pareto optimal or efficient if $C_1(i) \leq C_1(j)$ and $C_2(i) \leq C_2(j)$, with one of the inequalities holding strictly, for all j in the solution space.) As shown by the authors, all SPT sequences are efficient while an EDD sequence may or may not be efficient. This is true because any schedule not in SPT order will increase total flowtime; however, an EDD schedule will often have several jobs that can be resequenced and still provide the same value for T_o .

Sen and Gupta (1983) considered the same two objective as done by VanWassenhove and Gelders (1980), but developed a different approach to solve the problem. They used a weighted combination of mean flowtime and maximum tardiness as the objective function.

$$\text{Minimize } \{w(F) \cdot \bar{F} + w(T_{max}) \cdot T_{max}\} \quad (2.4)$$

where $w(F)$ and $w(T_{max})$ are the relative weights for flowtime and maximum tardiness respectively. It is obvious that a schedule which is in both SPT order and EDD order,

will always be an optimal schedule for this problem. However, when the SPT schedule differs from the EDD schedule, Sen and Gupta developed a branch-and-bound scheme. Upper bounds are determined by a pairwise interchange and a potential reduction is used to produce a lower bound. The linear combination approach of Sen and Gupta has the potential to identify each and every efficient point as determined by VanWassenhove and Gelders. Later, Hoogeveen and van de Velde (1995) also proposed an algorithm for this problem and showed that this problem can be solved in polynomial time. Their algorithm is based on a greedy method to determine a strict Pareto optimum. Koksalan (1999) developed a heuristic procedure to quickly find out a good solution for the bi-objective scheduling. Koksalan implemented the procedure for two different bi-objective scheduling problems: (i) minimizing total flowtime and maximum tardiness and (ii) minimizing total flowtime and maximum earliness.

However, these efforts on bi-objective scheduling with the objectives of minimizing mean flowtime and maximum tardiness have been made in the static case of a serial processing machine problem. In this thesis, the efforts are directed towards dynamic bi-objective scheduling of a single machine having serial processing as well as parallel batch processing. In dynamic scheduling, the concept of Pareto optimal/ efficient points becomes vague, which is first redefined for the dynamic scheduling environment in Section 2.3. It is then applied to a serial processing machine in Chapter 3 and to a batch processing machine in Chapter 4.

2.1.2 Minimizing mean and variation of flowtime

Minimization of mean completion times is one of the most commonly occurring regular measures, perhaps due to its equivalence to mean waiting time, mean lateness, and average in-process inventory. Yet in certain situations it is also desirable to reduce the variability of completion times, resulting in performance measures that are non-regular. For instance, in a service oriented environment, one might be interested in providing

as much uniform quality of service as possible based on the customers waiting times in the system. Another example where one might be interested in a variability measure was given by Merten and Muller (1972) in the context of organization of computer databases. They noted that, in an organization of computer files in large databases, it is desirable to provide uniform response time to users. The objective then is to determine an arrangement that minimizes variation of access times to the different records in these files.

Different measures have been proposed and developed for measuring the variation of completion times/flowtimes. Merten and Muller (1972) considered Completion Time Variance (CTV). Kanet (1981) proposed using Total Absolute Difference in Completion times (TADC). While Aneja et al. (1998) proposed a new measure of variation as Weighted Mean Absolute Deviation (WMAD) of completion times. Mosheiov (2000) modified WMAD as Mean Absolute Deviation of job completion times from the Mean Completion time (MADMC). The formulation of all these measures are as follows:

$$CTV = \frac{1}{n} \sum_{i=1}^n (C_i - \bar{C})^2 \quad (2.5)$$

$$TADC = \sum_{i=1}^n \sum_{j=1}^n |C_j - C_i| \quad (2.6)$$

$$WMAD = \sum_{i=1}^n w_i |C_i - \bar{C}_w| \quad (2.7)$$

$$MADMC = \frac{1}{n} \sum_{i=1}^n |C_i - \bar{C}| \quad (2.8)$$

where C_i is the completion time of job i , $\bar{C}$ is the mean completion time, w_i is the weight assigned to job i , and $\bar{C}_w = \sum_{i=1}^n w_i C_i$.

Mittenthal et al. (1996) provided a review of bi-objective scheduling problems involving a measure of central tendency of job completion times and a measure of variation of job completion times. They considered two measures of central tendency

2. Literature Review and Methodology

(mean and median) and five different measures of variation and thus discussed ten possible combinations of static bi-objective scheduling problems.

Bagchi (1989) considered TADC as a measure of the variation and total completion time ($TC = \sum_{i=1}^n C_i = n\bar{C}$) as a surrogate for mean completion time. Assuming total cost as a linear and non-decreasing function of TC and/or TADC alone, Bagchi formulated a convex combination model of the cost function as follows:

$$\min\{\alpha(TC) + (1 - \alpha)(TADC)\}, 0 \leq \alpha \leq 1 \quad (2.9)$$

Bagchi developed a polynomial algorithm to solve the problem and showed that it has multiple V-shaped optimal sequences. A V-shaped sequence has jobs in non-increasing order of processing times before the job with the shortest processing time, and in non-decreasing order of processing times after it.

De et al. (1992) provided a dynamic programming algorithm to the completion time variance problem with pseudo-polynomial time complexity and proved that the optimal sequence is V-shaped. They further showed that a bi-objective problem of minimizing mean and variance of job completion times could be solved in pseudo-polynomial time. The convex combination model of their bi-objective problem is as follows:

$$\min\{\alpha(\bar{C}) + (1 - \alpha)(CTV)\}, 0 \leq \alpha \leq 1 \quad (2.10)$$

However, the problem of bi-objective scheduling with the objectives of minimizing flowtime central tendency and its variation had been directed only on static cases of a single serial processing machine with small number of jobs. In this thesis, the efforts are focused on dynamic bi-objective scheduling of single machine with serial processing. The concept of Pareto control in dynamic scheduling has been applied to bi-objective scheduling of single serial processing machine with the objectives of minimizing mean and variance of cycle time in Chapter 3.

2.1.3 Minimizing earliness-tardiness (ET) performance

Machine scheduling problems with ET performance have received tremendous attention in the last fifteen years due to the growing interest in Just-In-Time (JIT) production strategies in the industry. The widespread use of JIT production strategy in manufacturing to eliminate inventories lead to a new class of scheduling problems in which both earliness (and/or number of early jobs) and tardiness (and/or number of tardy jobs) are penalized. The JIT production strategy is to produce only the necessary items in the necessary quantities at the required times. Thus in a JIT manufacturing environment, finishing jobs too early induces high inventory holding costs and may mean loss in profit due to environmental degradation of product quality, while finishing too late implies contractual penalty and loss of customer good will. In this type of environment, instead of a single due date, a due date window is given, since the customers will generally allow a time interval for the delivery of the products. If the job finishes within the time period of the due date window, it incurs no penalty; otherwise it is penalized either for being early or for being late. The problems dealing with earliness and tardiness objectives in JIT scheduling are often referred as “ET problems” in the literature. Correspondingly, the earliness and tardiness objectives are referred as “ET performance” by Wu and Weng (2005).

The scheduling problems with the ET performance constitute the most important part of the literature on multi-objective scheduling problems. Sidney (1977) was probably the first researcher to study the single machine scheduling problem with earliness and tardiness penalties. He developed an efficient algorithm to minimize maximum earliness or tardiness on a single static, serial processing machine, where jobs had distinct due dates. Garey et al. (1988) showed that the single machine scheduling problem with distinct due dates, where the objective is to minimize the sum of the absolute deviation of job completion times from their corresponding due dates, is NP-complete by reducing the even-odd partition problem to it.

2. Literature Review and Methodology

Ow and Morton (1989) developed a procedure to solve JIT scheduling problems assuming no inserted idle time. They implemented a new heuristic search method, called the Filtered Beam Search, where a limited number of candidate solution paths are searched in parallel and a certain number of good paths are stored at each stage. Ow and Morton tested the performance of this search method with certain dispatch heuristics like earliest due date first (EDD) and showed that the Filtered Beam Search method performed better and was more robust. Lee and Choi (1995) developed a genetic algorithm for sequencing jobs on a single machine with distinct due dates and early/tardy penalties. They reported computational results for problem sizes of up to 80 jobs but the optimality of the solutions for such large problems could not be verified since genetic algorithms provide only a heuristic approach to problem solving. Ventura and Radhakrishnan (2003) formulated the same problem with the objective of minimizing the sum of the absolute deviations of the job completion times from their corresponding due dates and developed a Lagrangian relaxation procedure to solve the problem.

Considerable research has been done in single machine scheduling with non-common due dates and a comprehensive survey of work done on earliness/tardiness scheduling problems until the late 1980s is provided by Baker and Scudder (1990). For scheduling problems with the ET performance, no known algorithm exists that yields near optimal solutions. These problems are complicated due to the planning horizon being unrestricted and the possibility of inserted idle times in an optimal schedule. The majority of the previous work provides computational results for problem sizes of up to 40 jobs only. For larger problem sizes, heuristic procedures like genetic algorithms and other search algorithms have been utilized to attain solutions.

Almost the entire body of literature on earliness/tardiness problems deals with static scheduling and a majority of the studies focus on single machine models. In addition, a bulk of the literature is rather rhetorical in that a large number of studies focus on a special problem in which all jobs have the same common due date, which

2. Literature Review and Methodology

are not discussed in this thesis. To the best available knowledge, only two papers deal with dynamic problems, *i.e.*, distinct arrival times. The dynamic problem becomes very complicated even for the single machine case. Nandkeolyar et al. (1993) proposed a modular approach for the dynamic single machine problem with equal weights for earliness and tardiness. Sridharan and Zhou (1996) developed a decision theory based approach for the problem, where jobs have distinct arrival and processing times, distinct due dates, and possibly unequal weights for earliness and tardiness. However, all these efforts on scheduling for ET performance are conducted on a serial processing machine problem and only one paper is found to address the earliness and tardiness scheduling problems on a batch processor but in a static setting (Qi and Tu, 1999).

The ET performance can be quantified by several measures. Here, two measures are considered for the ET performance - $\overline{(|E| + |T|)}$ and $\overline{(E^2 + T^2)}$ (Baker and Scudder, 1990; T'kindt and Billaut, 2002; Gordon et al., 2002). The earliness and tardiness of lot i are represented by $E_i = \max(d_i - C_i, 0)$ and $T_i = \max(C_i - d_i, 0)$, where d_i and C_i are the due date and completion time of lot i respectively. For dynamic systems, two ET performance measures are defined below based on the data collection of n lots in steady state.

$$\overline{|E| + |T|} = \frac{1}{n} \sum_{\forall i} (E_i + T_i) \quad (2.11)$$

$$= \frac{1}{n} \sum_{\forall i} (|C_i - d_i|) \quad (2.12)$$

$$\overline{E^2 + T^2} = \frac{1}{n} \sum_{\forall i} ((E_i)^2 + (T_i)^2) \quad (2.13)$$

First measure, $\overline{(|E| + |T|)}$, represents the problem of minimizing the sum of absolute deviations of the job completion times from their corresponding due dates. The second measure, $\overline{(E^2 + T^2)}$, is relevant for the cases where large deviations from the due dates are highly undesirable, and so it might be more appropriate to use squared deviations from the corresponding due dates as the performance measure (Baker and Scudder, 1990).

In this thesis, the work is carried out on dynamic scheduling of a batch processing machine for minimizing the ET performance. First, a look ahead batching method is developed for solving the ET problem in dynamic scheduling of a batch processing machine in Chapter 5. Further, the concept of Pareto control has been applied for tardiness and earliness related objectives in dynamic scheduling of a batching machine in Chapter 6.

2.2 Scheduling of a Batch Processing Machine

A parallel batch processing machine is a machine that can process up to a number of jobs simultaneously. The jobs that are processed together form a batch. All jobs contained in the same batch start and complete at the same time, since the completion time of a job is equal to the completion time of the batch to which it belongs. Webster and Baker (1995) presented an overview of algorithms and complexity results for scheduling a batch processing machine.

Generally, manufacturing scheduling involves two types of batch problems, especially in semiconductor manufacturing: burn-in oven model and furnace model (Webster and Baker, 1995; Hung, 1998). In the burn-in oven model, the processing time of a batch is equal to the processing time of the longest job in the batch, which is exemplified by the problem of scheduling burn-in ovens for back-end test operations. In the furnace model, the batch processing time is determined by the product type in a batch and is independent of the number of jobs in the batch. The furnaces used for the deposition operation in wafer fabrication is a typical example.

2.2.1 Burn-in oven model of batching

A series of work on this problem is presented by Lee et al. (1992), Chandru et al. (1993a), Chandru et al. (1993b), Uzsoy (1994), and Sanjay and Uzsoy (1998). Brucker

et al. (1998) developed dynamic programming algorithms for a single batching machine with burn-in model. They proved that due date based scheduling objective gives rise to NP-hard problems. Further, Potts and Kovalyov (2000) presented an extensive literature review on models that integrate scheduling with batching decisions.

In the semiconductor manufacturing industry, burn-in oven operation is near the end of the production process. Its scheduling is thus critical in determining ET performance and throughput of the entire system. Lee et al. (1992) presented efficient dynamic programming-based algorithms for minimizing the number of tardy jobs. Chandru et al. (1993a) and Chandru et al. (1993b) developed an optimal branch-and-bound algorithm for the single machine problem and heuristic algorithms capable of consistently obtaining near-optimal solutions in reasonable computational times. Uzsoy (1994) developed both a heuristic and a branch-and-bound approach to minimize the total completion time and the makespan of a given mix of jobs. Sanjay and Uzsoy (1998) presented a dynamic programming formulation and heuristics for a problem of minimizing total tardiness on a batch processing machine. Although Lee et al. (1992) and Sanjay and Uzsoy (1998) considered the due date objectives in scheduling the batch processing machine, they focused only on the static problems.

However, in the available literature, it seems that there does not exist any literature on the bi-objective scheduling of a batch processing machine in either the static or dynamic case. In this thesis, a dynamic bi-objective scheduling method for simultaneous minimization of mean cycle time and maximum tardiness is proposed and developed for a burn-in oven type batching machine in Chapter 4.

2.2.2 Furnace model of batching

Many researchers have developed dynamic scheduling heuristics for the batch scheduling problem with a furnace model such as Ikura and Gimple (1986), Glassey and

2. Literature Review and Methodology

Weng (1991), Fowler et al. (1992), Weng and Leachman (1993), Robinson et al. (1995), and Fowler et al. (2000).

In the area of production, planning and control studies, Ikura and Gimple (1986) were probably the first to address the problem of scheduling jobs on a batch processing machine from a deterministic scheduling perspective. In the semiconductor manufacturing domain, some work has been done on batch processing machines. Glassey and Weng (1991) added to the model of Ikura and Gimple, a feed forward control system to take advantage of forecast arrival times. They called it Dynamic Batching Heuristic (DBH). Using simulations for scheduling jobs of a single job family on a single batch processing machine, they showed that forecasting information can reduce the waiting time of lots arriving at a batch workstation. This real-time information is available (from shop floor control systems) in most wafer fabrication facilities (fabs), but is seldom exploited in decision making. Glassey and Weng showed that use of the real-time information can yield improvements in operation.

Based on this standpoint, Fowler et al. (1992) noticed that the further ahead the production planners look, the lower is the potential impact of occurring arrivals. They modelled the activity of loading by triggering the starting rule whenever a new job reaches the queue. Their rule is referred to as Next Arrival Control Heuristic (NACH). They also extended the approach to the multi-product environment, where a decision taken for one product can have a major impact on the delays experienced by batches of different product types.

Weng and Leachman (1993) addressed the same problem examined by Glassey and Weng (1991) and by Fowler et al. (1992) and proposed a new rule, Minimum Cost Rate Heuristic (MCR). MCR attempts to minimize the holding cost per unit time, which essentially minimizes the waiting time of lots in the queue. Robinson et al. (1995) extended these rules by using the information on the current and expected future workload status of both upstream and downstream workstations. They proposed a

2. Literature Review and Methodology

Rolling Horizon Cost Rate Heuristic (RHCR), which is essentially a combination of NACH and MCR, *i.e.*, the cost rate function used in MCR is incorporated into the rolling planning horizon used in NACH. Fowler et al. (2000) later extended their rules for single server case (Fowler et al., 1992) to a multiple server case.

The above mentioned related works exploit only the arrival information of the future incoming lots and focus on controlling the waiting time in front of the machines and thus the cycle time related objectives of the schedule. However, the work done herein does not consider the ET performance (simultaneous minimization of earliness and tardiness related objectives) in a JIT manufacturing environment, which are very relevant to semiconductor manufacturing.

In this thesis, a new Look-Ahead Batching (LAB) method is proposed for the ET performance in dynamic batch scheduling in Chapter 5. LAB method exploits the arrival and due-date information of the future coming lots to optimize the batching decisions. A wide range of simulation experiments are conducted to show the effect of LAB over the MBS, DBH and NACH policies. Further, LAB method for ET performance is extended to develop the Pareto control between the conflicting objectives of earliness and tardiness in dynamic scheduling of a batch processing machine in Chapter 6. Following sections present a brief review of literature with respect to the due date related objectives in batch scheduling.

Hung (1998) presented a scheduling method for minimizing total tardiness of the schedule at E-beam writers in mask shop, which is also modelled as furnace batching. He proposed a dynamic programming formulation to calculate the optimal batching decisions at each batch processing machine and showed its improved results over the mask shop current scheduling method and the full batch method. But, this work is also focused on the static problems with consideration of only one due-date related objective, *i.e.* minimization of the total tardiness. In addition, it does not reap the benefit of the future arrival information from shop floor control systems.

Further, Kim et al. (2001) presented three batch scheduling rules, Modified Minimum Batch Size rule, Modified Dynamic Batching Heuristic, and Processing Urgency Classification Heuristic. Although they used a dynamic system that exploits the future arrival information, they did not focus on optimization of due date objectives as such. In all these modified rules, they used due dates of the jobs only to select a family to be processed next on the available batch processing machine in a multiple job family environment. A family with the least average slack time is selected to be processed next. They did not exploit the due date information of the future incoming lots in order to optimize the due date objectives.

In the available literature, it seems that there is no literature with respect to the ET performance in dynamic batch scheduling. In this aspect, the development of LAB for the ET performance and the ET performance schedule control is a new contribution of this research.

2.3 Methodology: Pareto Control in Multi-Objective Dynamic Scheduling

In this work, the concept of Pareto optimal solutions has been extended and applied to multi-objective scheduling problems in dynamic cases, where jobs are assumed to arrive continuously at discrete times. The multi-objective dynamic scheduling problem can be considered as a multi-stage multi-objective optimization problem in real time.

The single-stage multi-objective optimization problems are generally solved by combining the multiple objectives into one scalar objective, whose solution is a Pareto optimal point for the original problem. “Geoffrion’s theorem” states that the minimizer of these convex combined functions is Pareto optimal (Geoffrion, 1968). “Geoffrion’s theorem” concerns a necessary and sufficient condition for the determination of proper Pareto optima, as stated in the following theorem (T’kindt and Billaut, 2002):

Theorem: (Geoffrion, 1968)

Let S be the convex set of the solutions and P criteria f_j that are convex functions on S . x^* is a proper Pareto optimum if and only if $\exists w \in \mathbb{R}^P$, with $w_j \in]0; 1[$ and $\sum_{j=1}^P w_j = 1$, such that x^* is an optimal solution of the problem:

$$\text{Min } g(f(x)) \text{ with } g(f(x)) = \sum_{j=1}^P w_j f_j(x)$$

subject to: $x \in S$.

Summary of Algorithm:

Geoffrion (1968) achieved the proof of the theorem by the method of contradiction. First, it is assumed that x^* is an optimal solution of the above problem with w fixed and it is shown that x^* is a proper Pareto optimum. Then, x^* is assumed as a proper Pareto optimum and it is shown that $\exists w \in]0; 1[$ such that x^* is an optimal solution of the above problem. $\square$

Geoffrion's theorem can be extended to the multi-stage multi-objective optimization problems with an assumption that if an objective function $(F_j, j = 1, 2, \dots, P)$ can be optimized by optimizing a criterion $(f_j, j = 1, 2, \dots, P)$ at each stage, then optimizing a convex combination of these criteria $(\sum_{j=1}^P w_j \cdot f_j)$, such that $\exists w \in \mathbb{R}^P$, $w_j \in]0; 1[$ and $\sum_{j=1}^P w_j = 1$, at each stage will give a Pareto optimal solution for the multi-stage multi-objective optimization problem, $\text{Min}(F_1, F_2, \dots, F_P)$. Applying this in the context of multi-objective dynamic scheduling problems, if the objective function F_1 can be optimized by selecting jobs in real time according to criterion f_1 and the objective function F_2 can be optimized by choosing jobs in real time according to criterion f_2 , then it can be assumed that by selecting jobs in real time according to a convex combination of criteria f_1 and f_2 will lead to a Pareto optimal solution for the multi-objective dynamic scheduling problem, $\text{Min}(F_1, F_2)$.

Since the term "Pareto optimal solutions" or "Pareto sequences" as presented for the static problems in Section 1.2 may not be applicable to multi-objective dynamic scheduling (MODS) problems. In dynamic problems, the sequence of jobs remains

open-ended and the scheduling objectives are computed after long simulation runs in steady state, excluding the warm-up period. Therefore, the concept of Pareto control has been adopted for solving MODS problems. Pareto control in dynamic scheduling is defined as controlling the performance of conflicting objectives within the operating range. The concept of Pareto control in dynamic scheduling is shown in Figure 2.2. If two conflicting objectives are considered, then the trade-off between them is represented by a Pareto control curve as shown in Figure 2.2 (a). If there are three conflicting objectives considered simultaneously, then there will be a Pareto control surface in three dimensional space as presented in Figure 2.2 (b).

The concept of Pareto control can be explained as follows. Suppose there are two conflicting objective functions (also called bi-objective function), F_1 and F_2 , in dynamic single machine scheduling. Assuming that these objectives are individually optimized by scheduling a job at each decision instance in simulation clock according to the corresponding criteria of $f_1(x)$ and $f_2(x)$, for $x \in X$, where x is the job index and X is the set of all available jobs at that particular decision instance. For bi-objective scheduling, at each decision instance on the time horizon, a convex combination approach is used to define a joint criterion. This joint criterion is used to select a Pareto optimal job based on the relative importance of scheduling criteria $f_1(x)$ and $f_2(x)$. This results in a particular sequence of dynamically arriving jobs for scheduling and corresponding objective functions, F_1 and F_2 . Varying the relative importance of scheduling criteria at each decision instance will result in Pareto control of bi-objective dynamic scheduling as shown in Figure 2.2 (a).

For scheduling dynamically arriving jobs on the machine, a discrete-event simulation-based approach of scheduling is used. In this approach, at each decision instance in the simulation clock, an optimal job is selected from the dynamic queue of jobs and scheduled on the machine for processing. The simulation clock is then moved forward to the next decision instance. Generally, scheduling and simulation are studied as two sepa-

(a) Pareto control curve for two objectives case

(b) Pareto control surface for three objectives case

Figure 2.2: Pareto control in multi-objective dynamic scheduling

rate concepts; scheduling for sequencing jobs on the machine and simulation as a tool to validate the results. Here, the concepts of scheduling and simulation are combined and thus used to exploit the benefits of discrete-event simulation in dynamic scheduling of the shop. This combined approach is referred to as Conjunctive Simulated Scheduling (CSS) (Gupta and Sivakumar, 2005). Using CSS, a multi-objective dynamic scheduling problem is disintegrated into several decision instance based multi-objective optimiza-

tion problems. Thus, an appropriate job, which satisfies multiple criteria of scheduling objectives need to be selected at each decision instance in the simulation clock.

Therefore, unlike finding a Pareto sequence in static scheduling problems, in multi-objective dynamic scheduling problems a Pareto optimal job is selected for scheduling on the machine at each decision instance in the simulation clock. For this, first a Pareto job need to be determined among the queued up jobs at each decision instance.

2.3.1 Determination of Pareto optimal job

At each decision instance in dynamic scheduling, the Pareto optimal jobs are determined. Suppose the criteria $f_1(x)$ and $f_2(x)$ correspond to minimization of the processing time (p_i) and minimization of the due date (d_i), then at each job selection instance all jobs can be represented on a two dimensional graph between p_i and d_i . For the job data in Table 1.2, the graph is presented in Figure 2.3, where each point corresponds to a job which is a competitive candidate for being loaded on the machine. In this figure, an encircled point can be considered as a Pareto optimal job (Gupta and Sivakumar, 2002). A Pareto optimal job is selected to schedule on the machine at each decision instance in the simulation clock, and thus achieve a schedule of dynamic jobs, which gives a boundary solution for the bi-objective dynamic scheduling problem.

Various approaches exist to find the Pareto optimal solution for a multi-objective optimization problem as reported by Zeleny (1973); Yu (1973); Keeney and Raiffa (1976); Goicoechea et al. (1982); Zeleny (1982); Sawaragi et al. (1985); Yu (1985); Tabucanon (1988); T'kindt and Billaut (2001, 2002). These approaches generally transform the original multi-objective problem into a new single objective problem in which the objective function is an aggregation of different individual objectives with new parameters. Based on texts by Yu (1985); Steuer (1986); Tabucanon (1988); T'kindt and Billaut (2002), these approaches are enlisted as follows:

Figure 2.3: Graph showing all jobs in Table 1.2 with points (d_i, p_i) .

[Encircled points present Pareto optimal jobs with respect to $\min(p_i, d_i)$.]

- * Determination by convex combination of objectives,
- * Determination by means of the ϵ -constraint approach,
- * Use of the Tchebycheff metric, weighted Tchebycheff metric, *etc.*,
- * Use of Goal Programming, De Novo Programming
- * Use of Multicriteria Mixed Integer/Linear Programming, *etc.*

The description of these approaches is beyond the scope of this thesis. The most commonly used approach in multi-objective optimization is to transform the problem into single objective problem using convex combination of objectives and weights. The weights are decided based on the relative importance of each objective. Therefore, in this research, the focus has been only on the convex combination approaches, because of their simplicity of application in dynamic scheduling, low cost of computation, and ability to provide feasible solution in near-real-time.

2.3.2 Convex combination approaches

Most of the convex combined functions are either in a linear fashion or in the form of some kind of the distance derivatives (Tabucanon, 1988; Yu, 1985; Zeleny, 1982). The

prominent techniques in this relation are Weighted Aggregation (WA) and Compromise Programming (CP), which are described in detail in the following subsections.

Weighted aggregation (WA) method

A standard technique in MOO is to minimize a positively weighted convex sum of the objectives. However, it is up to the user to choose appropriate weights. Until recently, considerations of computational expense forced users to restrict themselves to performing only one such minimization, considering just one set of weights chosen with care. Newer, more ambitious approaches aim at minimizing convex sums of the objectives for various settings of the convex weights, thus generating various points in the Pareto set. Though computationally more expensive, this approach gives an idea of the shape of the Pareto surface and provides the user with more information about the trade-off among the various objectives. Thus, this method is aimed at minimizing convex sums of the objectives for various settings of the convex weights in order to get a trade-off among the various objectives. In this WA method, different objectives are weighted and summed up in a linear fashion to a single objective, formulated as:

$$\min F = \sum_{j=1}^2 w_j \cdot f_j(x) \quad (2.14)$$

where w_j are non-negative weights with $\sum w_j = 1$. By varying these weights, all the Pareto optimal points can be achieved as each Pareto optimal solution point on a convex surface corresponds to a set of w_j (Geoffrion, 1968). This method is the simplest possible approach to solving the MOO problem. From application point of view, the user may be having only an intuition of the importance of one objective over the other, without having any knowledge of an exact set of weights for their various objectives, as it is very tough to establish a relationship between these weights and the real outcome in terms of objective function values. This complexity can be dealt with by using the idea of finding a Pareto boundary by assigning varying weights to the

objectives. Thus, the user has the full set of Pareto optimal points, defining a Pareto boundary in static cases.

One problem with the WA method is that of dimensional inconsistency among various objectives. Therefore, it is often observed that due to the different units of the objectives, one objective functional value appears to become dominant in the overall weighted aggregation of objective functions, even in various combinations of the weights. As a result, this approach becomes misleading, *i.e.*, always deciding in favor of a particular objective, unless normalization is performed. One of the simple normalization techniques to overcome this problem is to use relative scaling, *i.e.*, dividing each objective function by its maximum value and using the weighted ratio sum. The maximum value of each objective function is taken from the available data set and thus it dynamically changes at each decision instance.

However, this method suffers from two more drawbacks in solving MOO problems (Das and Dennis, 1997). First, the relationship between the vector of weights and the Pareto curve even in convex MOO problems is such that a uniform spread of weight parameters rarely produces a uniform spread of points on the Pareto set. Often, all the points found are clustered in certain parts of the Pareto set with no solution in the interesting “middle part” of the set, thereby providing little insight into the shape of the trade-off curve. The second drawback is that the non-convex parts of the Pareto set cannot be obtained by minimizing convex combinations of the objectives, though the existence of a non-convex part in a Pareto boundary is a very rarely occurring phenomenon. Motivated by the obvious need of looking for a more powerful approach, Zeleny (1973, 1982), Yu (1985) and others developed the CP method.

Compromise programming (CP) method

The compromise programming (CP) method is a distance based technique (Zeleny, 1973; Yu and Leitmann, 1974; Zeleny, 1982). To define this, first let x^* denote the

utopia (or ideal) point, which gives the vector of the optimum value of each objective function. Thus $F^*(x^*) = [f_1(x^*), f_2(x^*)]$ is the utopia vector. This vector can not be obtained unless both objectives are non-conflicting, *i.e.*, this ideal objective vector is only a hypothetical solution that never exists in the practical real-life problems. Similarly, let x_0 denote the nadir (or anti-ideal or anti-utopia) point, which is defined by the individual function-wise worst values, but considering only Pareto optimal solutions. Thus $F_0(x_0) = [f_1(x_0), f_2(x_0)]$ is the nadir vector, which may not be an actual solution in the feasible space. The geometric interpretation of these points is demonstrated in Figure 2.4.

Figure 2.4: Utopia and Nadir points in compromise programming

CP method identifies solutions which are closest to the utopia solution as determined by some measure of distance (Zeleny's (1982) axioms of choice). The solutions identified as being closest to the utopia solution are called compromise solutions and constitute the compromise set. In the CP method, the point of interest is the comparison of distances of different efficient points $(f_j(x), j = 1, 2)$ from the utopia solution which is the point of reference. Since the objectives may be of different dimensions, so the distance measure needs to be corrected to make the individual objectives mutually commensurable. It is therefore necessary to use relative rather than absolute

2. Literature Review and Methodology

deviations. The individual relative deviations can be raised to any power ($r > 0$) before these are summed and also the weights w_j ($0 < w_j < 1$ and $\sum_j w_j = 1$) can be attached to the different relative deviations. For a MOO problem, with utopia point $F^*(x^*) = [f_1(x^*), f_2(x^*)]$, the overall minimizing objective function can be expressed as follows:

$$\min F_r = \left[\sum_{j=1}^2 \left[w_j \cdot \frac{|f_j(x^*) - f_j(x)|}{f_j(x^*)} \right]^r \right]^{1/r} \quad (2.15)$$

In this formulation, the choice of exponent r reflects the user's concern with respect to the maximal deviation. Introduction of w_j allows the expression of the user's intuition concerning the relative importance of the various objectives. Thus, a double-weighting scheme exists. The parameter r reflects the importance of the maximal deviation and the parameter w_j reflects the relative importance of the j^{th} objective. From the application point of view, both these parameters give the benefit of controlling the generated solution over the Pareto boundary. In the above CP formulation, keeping the utopia point the same but by changing the weights, one may reach all the efficient points located on the boundary (Zeleny, 1982). In the observations of Gupta and Sivakumar (2002), at any selection instance, all Pareto optimal jobs can be selected by the CP method by varying weights w_j within a particular spread of boundary. On the other hand, varying exponent r helps in controlling the spread of this boundary.

Similarly, CP method can also be defined with reference to the nadir point, where CP method will identify solutions which are farthest to the nadir point. The distances of different efficient points ($f_j(x), j = 1, 2$) from the nadir point $F_0(x_0) = [f_1(x_0), f_2(x_0)]$ are compared and the efficient point with the maximum distance is chosen. The corresponding overall maximizing objective function can be formulated as:

$$\max F'_r = \left[\sum_{j=1}^2 \left[w_j \cdot \frac{|f_j(x_0) - f_j(x)|}{f_j(x_0)} \right]^r \right]^{1/r} \quad (2.16)$$

Using the theory of fuzzy sets, Zeleny (1982) showed that minimizing F_r (equation 2.15) or maximizing F'_r (equation 2.16) for certain values of exponent r would

2. Literature Review and Methodology

provide precisely the same solutions, which is also supported by our observations in applying CP method for selecting Pareto optimal jobs at each selection instance in scheduling. Operationally, three points of the compromise set are of mathematical importance, those corresponding to $r = 1, 2$, and ∞ . With $r = 1$ and equal weights w_j , the CP method is equivalent to the Global Criteria method, whose single optimization function is formulated as (Tabucanon, 1988):

$$\min F_r = \sum_{j=1}^2 \left[\frac{|f_j(x^*) - f_j(x)|}{f_j(x^*)} \right] \quad (2.17)$$

where $f_j(x^*)$ is optimum value of the singular objective function j at its optima point x^* , $f_j(x)$ is the function value itself.

When $r = 2$, the equation becomes simply the quadratic function of distance between two points in 2-D space for a bi-objective problem, where each relative deviation is weighted in proportion to its magnitude. As r becomes larger and larger, the largest deviation receives more and more weight. For r approaching ∞ , the distance measure reduces to the Tchebycheff function (Zeleny, 1982):

$$\begin{aligned} \lim_{r \rightarrow \infty} F_r &= \lim_{r \rightarrow \infty} \left[\sum_{j=1}^2 \left[w_j \cdot \frac{|f_j(x^*) - f_j(x)|}{f_j(x^*)} \right]^r \right]^{1/r} \quad (\text{from Eqn. 2.15}) \\ &= \lim_{r \rightarrow \infty} [T_1^r + T_2^r]^{1/r} \quad \left(\text{where } T_j = \left| w_j \cdot \frac{f_j(x^*) - f_j(x)}{f_j(x^*)} \right|, j = 1, 2 \right) \\ &\quad (\text{taking } \ln \text{ both sides and interchanging } \lim \text{ and } \ln) \end{aligned}$$

$$\begin{aligned} \Rightarrow (a) \lim_{r \rightarrow \infty} \ln F_r &= \lim_{r \rightarrow \infty} \left[\frac{1}{r} \cdot \ln \left(T_1^r \left(1 + \left(\frac{T_2}{T_1} \right)^r \right) \right) \right] \quad (\text{assuming that } T_1 > T_2) \\ &= \ln T_1 + \lim_{r \rightarrow \infty} \left[\frac{1}{r} \cdot \ln \left(1 + \left(\frac{T_2}{T_1} \right)^r \right) \right] \\ &= \ln T_1 \quad \left(\because \frac{T_2}{T_1} < 1, \therefore \lim_{r \rightarrow \infty} \left[\frac{1}{r} \cdot \ln \left(1 + \left(\frac{T_2}{T_1} \right)^r \right) \right] = 0 \right) \end{aligned}$$

Similarly,

$$\begin{aligned} \Rightarrow (b) \lim_{r \rightarrow \infty} \ln F_r &= \ln T_2 \quad (\text{assuming that } T_2 > T_1) \\ &\quad (\text{thus combining both (a) and (b)}) \end{aligned}$$

$$\Rightarrow \lim_{r \rightarrow \infty} F_r = \max_{j=\{1,2\}} T_j$$

This leads to the following conclusion:

$$\lim_{r \rightarrow \infty} F_r = \max_{j=\{1,2\}} \left[\left| w_j \cdot \frac{f_j(x^*) - f_j(x)}{f_j(x^*)} \right| \right] \quad (2.18)$$

This is because the relative contribution of the largest relative deviation, when raised to a large exponent, would be extremely larger than all the rest combined and thus will dominate the distance determination.

Though the weights representing relative importance are used as the preference structure when applying CP, it has been mathematically proven that CP method is superior to WA method in locating the efficient solutions (Steuer, 1986). Thus, CP method is widely applied in mechanical engineering design problems, where single stage multi-objective optimization problems exist. However, there are very few applications of CP to multi-objective scheduling problems. Taboun et al. (1995) used CP to solve two bi-objective scheduling problems with objectives of (i) minimizing total tardiness and earliness *vs.* mean flowtime and (ii) minimizing total tardiness *vs.* total earliness. They modelled the scheduling problem using linear programming and used CP to generate efficient points of bi-objective static scheduling problem having 13 jobs to schedule on a single machine. Koksalan (1999) exploited CP to develop a heuristic procedure for solving two different bi-objective scheduling problems: (i) minimizing total flowtime and maximum tardiness and (ii) minimizing total flowtime and maximum earliness. Their application also addressed only static scheduling problems and presented an example of scheduling 15 static jobs on a single machine.

However, to the best knowledge available, research on multi-objective dynamic scheduling has not been addressed yet in the existing literature. Thus, this research addresses the use of achieving the Pareto control for the multi-objective dynamic scheduling problems.

2.4 Conclusion

In this chapter, first, a literature review framework has been developed for multi-objective dynamic scheduling of manufacturing processes. Then, the existing literature is reviewed in two parallel directions: multi-objective scheduling with serial processing and single objective scheduling with batch processing. Multi-objective scheduling review has been divided into three parts with respect to the three pairs of the conflicting objectives: (i) the mean cycle time and the maximum tardiness, (ii) the mean cycle time and the cycle time variance, and (iii) the earliness and the tardiness related objectives. The review on batch scheduling has been done in the two directions: the burn-in oven model and the furnace model of batching.

Further, the methodology involved adopting the concept of “Pareto control” in the context of multi-objective dynamic scheduling problems. The conjunctive simulated scheduling approach is used to schedule dynamically arriving jobs on the machine. Thus, a multi-objective dynamic scheduling problem is disintegrated into several decision instance based multi-objective optimization problems. At each decision instance in simulation clock, a Pareto job is selected to satisfy the multiple criteria of scheduling objectives. For selecting Pareto jobs, two convex combination approaches are presented: WA and CP methods.

The concept of “Pareto control” is applied and computationally experimented for the serial processing machine in Chapter 3 and for the batch processing machine in Chapter 4. Further, Pareto control is computationally experimented for look ahead batch processing in Chapter 6. Finally, a real-life case study example for Pareto control in serial processing is presented in Chapter 7.

Chapter 3

Pareto Control in Serial Processing

3.1 Introduction

In the previous chapter, the Pareto control concept was adopted for multi-objective dynamic scheduling problems. In this chapter, the Pareto control concept is applied and computationally experimented for bi-objective dynamic scheduling problems with a single serial processing machine. The two pairs of conflicting objectives considered are (i) mean cycle time *vs.* maximum tardiness and (ii) mean cycle time *vs.* variance of cycle time, as earlier discussed in Section 2.1.1 and Section 2.1.2.

In real life, manufacturing systems are dynamic and stochastic, and hence the decision maker makes use of dispatching rules to determine the order in which jobs are to be loaded and processed by the machines. A dispatching rule selects a job to be processed next from a set of jobs in the machine queue, when the facility becomes free. Development of dispatching rules has been normally intended to minimize the inventory and tardiness costs. A good review of reports on the use of dispatching rules can be seen from Blackstone et al. (1982), Haupt (1989) and Ramasesh (1990) for jobshop scheduling with flowtime related performance measures. There also have been quite a lot of studies on development of efficient rules (Russell et al., 1987) and combined rules (Holthaus and Rajendran, 1997). Lu et al. (1994) particularly addressed the prob-

lem of reducing the mean and variance of cycle time in semiconductor manufacturing plants. A customary practice in scheduling research is to minimize the flowtime related and tardiness-related measures of performance since the associated inventory and tardiness costs are assumed to be directly proportional to the time periods of flowtime and tardiness of jobs respectively (Blackstone et al., 1982).

In contrast to the above, this chapter focuses on bi-objective scheduling problem for single machine system with dynamic job arrival. The use of convex combination approaches is proposed for achieving Pareto control in dynamic scheduling. As discussed previously in Section 2.3, the most commonly used approach in multi-objective optimization is to transform the problem into a single objective problem using a convex combination of objectives. However, published research on simultaneous minimization of multi-objectives is limited to only static problems, even though the dynamic scheduling problems have been extensively investigated in the last few decades.

In dynamic single machine systems, the shortest processing time ($\min p_i$) criterion minimizes the mean cycle time (Conway and Maxwell, 1962). The earliest due date ($\min d_i$) criterion minimizes the maximum tardiness in static case of single machine scheduling. In case of dynamic scheduling also, the earliest due date criterion tends to minimize the maximum tardiness for single machine systems. These criteria are used in the convex combination approaches for simultaneous minimization of mean cycle time and maximum tardiness in the dynamic scheduling of a single machine. In both cases, serial and batch processing machine, the Pareto control is achieved for the wide range of weights associated with these objectives.

3.2 Model, Assumptions and Constraints

In this chapter, a serial processing machine has been modeled with dynamic job arrivals and the computational experiments are carried out for achieving Pareto control. The

3. Pareto Control in Serial Processing

motivation of this study is to facilitate the manufacturing management with a methodology to control the shop floor schedule according to the demands of dynamic market forces. Especially, in semiconductor manufacturing, the production is highly driven by the need of low cycle times and their high repeatability (*i.e.*, low cycle time variance) on one hand and by the requirements of meeting tight dead-lines on the other. The relative importance of these objectives changes with time. Due to the highly complex nature of shop floor at both front-end and back-end in semiconductor manufacturing, it is preferred to have dynamic near-real-time scheduling based on conjunctive simulated scheduling. The conjunctive simulated scheduling approach provides the flexibility to make effective changes for achieving control over the conflicting objectives.

The model consists of a single serial processing machine with the dynamic arrival of jobs, *i.e.*, the jobs arrive at discrete times. Figure 3.1 represents the schematic diagram of the model. Many production systems can be modeled as single machine models, particularly when, in a multiple-machine environment, the performance of a particular bottleneck machine determines the performance of the entire system.

Figure 3.1: Schematic diagram of the model with serial processing machine

The related assumptions and constraints can be summarized as follows:

1. The jobs are independent, *i.e.*, the jobs are processed without any precedence.
2. The jobs arrive dynamically at the machine (*i.e.*, $r_i \neq 0, \forall i$). The dynamic arrival of jobs can be either deterministic or stochastic in nature.

3. The serial processing machine can process only one job at a time.
4. No machine idle time and breakdowns; machine is always available for processing.
5. Setup times are independent of the job sequence and are included in the processing times.
6. No interruption to jobs in processing, implying no job pre-emption.
7. There are no other limiting resources such as material or labor.

3.3 Scheduling Objectives

The scheduling objective functions under study are described in the following.

1. Minimizing Mean Cycle Time ($\bar{F} = \sum_{i=1}^n F_i/n$): the average amount of time that jobs spend on the machine.
2. Minimizing Maximum Tardiness ($T_{max} = \max(T_1, T_2, \dots, T_n)$): the maximum value of tardiness of the jobs in steady state.
3. Minimizing Cycle Time Variance ($\sigma_F^2 = \sum_{i=1}^n (F_i - \bar{F})^2/n$): the variance of cycle times of the jobs in steady state.

where n is the number of jobs considered in steady state simulation statistics, F_i is the cycle time of job i and T_i is the tardiness of job i .

In scheduling practice, it is almost always desirable to minimize each of these three objectives. Mean cycle time and maximum tardiness are contradicting in nature as earlier presented in Section 2.1.1. With the decrease in mean cycle time, the spread of cycle time of jobs and so the maximum cycle time of jobs increases, which in turn increases the maximum tardiness. Similarly from Section 2.1.2, mean cycle time and

cycle time variance are contradicting in nature. As the mean cycle time decreases, the spread of the cycle time of jobs increases and vice-versa.

Therefore, in this chapter, two pairs of conflicting objectives are addressed for dynamic single machine scheduling problem.

- Minimizing the mean cycle time and the maximum tardiness

$$\min(\bar{F}, T_{max})$$

- Minimizing the mean cycle time and the cycle time variance

$$\min(\bar{F}, \sigma_F^2)$$

3.4 Computational Experiments for Pareto Control in Serial Processing

Table 3.1 summarizes the factors considered for the simulation experiments. The experimental design consists of three independently varying parameters, the processing times, the due dates and the utilization level.

The first factor is the utilization level (ρ) at two different levels, 0.75 and 0.85, which are considered for the normal operation of the machine. Above 0.85 utilization level, the system becomes unsteady; while 0.75 utilization level is taken as the average loaded condition of the machine. The second factor, processing time (p_i), is assumed to include the setup time in it. The processing times of jobs are assumed to have a uniform distribution with equal mean and range considering the nature of semiconductor manufacturing processes. Thus, the lower and upper limits of the interval are set at a distance of half of the mean from the mean of the distribution. The mean and range of the uniform distribution is assumed to be 30 time units as a sample study.

Table 3.1: Factors for simulation experiments

No.	Factor	Levels	Description
1.	Utilization Level (ρ)	Two	0.75, 0.85
2.	Processing Time (p_i)	Uniform	Mean (μ) = Range = 30 min, <i>i.e.</i> , $p_i \in U[15,45]$
3.	Interarrival Distribution	Exponential	Mean ($\lambda = \rho \cdot \mu$)
4.	Due Date (d_i)	Tight ($k = 1$) Loose ($k = 2$)	$d_i = r_i + k \cdot \bar{F}_i$
5.	Conflicting Objectives	Two pairs	Minimizing ($\bar{F}, T_{max}$) Minimizing ($\bar{F}, \sigma_F^2$)

Since the Poisson process is the most commonly used model for the arrival process of customers to a queuing system (Law and Kelton, 2000), so the inter-arrival time, the third factor, is assumed to be exponentially distributed with the mean being controlled by the utilization factor. For single machine (or server) queuing systems, the utilization factor can be related to the arrival rate as $\rho = \lambda/\mu$, where λ is mean arrival rate and μ is mean service rate. Hence, keeping the service rate at a constant value for all the experiments, the arrival rate is set to depend on the utilization factor.

The uniformly distributed processing times are generated as follows (Law and Kelton, 2000):

$$\begin{aligned} \text{Mean} &= 1/\mu, \text{ Range} = 1/\mu, \text{ Lower limit} = 1/(2\mu) \\ \Rightarrow \quad p_i &= [1/(2\mu) + U(100)/(100\mu)] \end{aligned} \quad (3.1)$$

3. Pareto Control in Serial Processing

where $U(100)$ generates a uniform random number between 0 and 100.

The exponentially distributed inter-arrival times are generated by the following expression (Law and Kelton, 2000):

$$r_i = -\ln [U(100)/100] / \lambda \quad (3.2)$$

The fourth factor is the due date (d_i), which is assigned to the jobs based on its arrival on the machine and average system cycle time as presented in the following expression:

$$d_i = r_i + k \cdot \bar{F}_i \quad (3.3)$$

where the constant k (> 0) represents the due date allowance factor and $\bar{F}_i$ represents the mean cycle time of jobs in the system at the time of arrival of job J_i . The same degree of allowance between the due date and the arrival time is assigned to all jobs. The due date allowance factor gives the flexibility to vary the tightness of due-dates in order to study its effects on achieving Pareto control. In the experimentation, $k = 1$ is considered as tight due-date setting, while $k = 2$ is loose due-date setting.

The final factor is the conflicting objectives in multi-objective dynamic scheduling of serial processing machine. As described in Section 3.3, two pairs of conflicting objectives are addressed: (a) minimizing the mean cycle time ($\bar{F}$) and the maximum tardiness (T_{max}) and (b) minimizing the mean cycle time ($\bar{F}$) and the cycle time variance (σ_F^2).

In this study, each simulation experiment consists of 10 different runs (or replications) and in each run the shop is loaded continuously with jobs that are numbered on their arrival. Thus, each run represents a different randomly generated problem instance. For determining the steady state of the system, the mean of the cumulative cycle times is continuously observed. It is found that the system reaches a steady state approximately after the arrival of about 400 jobs for all the runs. Figure 3.2 presents

3. Pareto Control in Serial Processing

the sample graph of the steady state of cumulative average cycle time for SPT method at 0.75 utilization level. To be on the safe side, the steady state data is collected from job completion number 500 onwards. Total sample size is in the order of thousands of job completions (Conway and Maxwell, 1962) and it is preferable to have a smaller number of replications and a larger run length (Law and Kelton, 1984). Hence, the number of replications is fixed at 10 and the run length as 1000 jobs. For computation of statistics the data is collected from job completions, numbering from 501 to 1500, and the machine is continuously loaded till the completion of all the 1500 jobs. A flow chart of the simulation experiment is presented in Figure 3.3. The simulation program is written in C++, implemented in Turbo C++ environment and run on an Intel 2000 Pentium (4), 1.8 GHz PC.

Figure 3.2: Steady state for cumulative average cycle time at 0.75 utilization level

Figure 3.3: Flow chart for simulation experiments on serial processing machine

3.5 Scheduling Criterion Selection Using Single Objective

In the main simulation experiments for Pareto control, a criterion is required for minimizing each of the individual scheduling objectives in dynamic scheduling of a serial processing machine. In this context, the criterion refers to the rule by which a job is selected for scheduling on the machine. This section presents a preliminary study for selecting a criterion ($f_j(x), j = 1, 2$) for each scheduling objective. This preliminary study is conducted on the simulation results of single objective dynamic scheduling of single machine. The experimental conditions for this preliminary study are kept similar to the conditions described in the above section. These selected criteria will then be used in WA (Equation 2.14) and CP (Equation 2.15) methods for selecting a Pareto job at each decision instance in simulation.

In order to select an appropriate criteria for each scheduling objective, several combinations are explored, which are based on the job and machine related information of processing time (p_i), due date (d_i), release time (r_i), time instance (t) and waiting time ($W_i = t - r_i$). These combinations serve the purpose of job priority index or the scheduling criteria at each decision instance in simulation clock. The job priority index is used to set the priority of queued jobs for scheduling on the machine. The scheduling objectives under consideration are minimizing mean cycle time, minimizing maximum tardiness and minimizing cycle time variance as discussed in Section 3.3. The experiments for single objective dynamic scheduling of a serial processing machine are conducted at two different levels of utilization factor ($\rho = 0.75, 0.85$) for the following job priority indices: (1) $\min(d_i)$ [earliest due date], (2) $\max(p_i)$ [longest processing time], (3) $\max(p_i/(d_i - p_i - t))$ [maximum critical ratio], (4) $\max(p_i + W_i)$, (5) $\min(p_i + W_i)$, (6) $\min(d_i - p_i - t)$ [minimum slack], (7) $\min(p_i/W_i)$ and (8) $\min(p_i)$ [shortest processing time].

3. Pareto Control in Serial Processing

Table 3.2: Mean cycle time performance of various priority indices ($\rho = 0.75$) (In each row, S_B represents the best priority index and S_W represents the worst priority index for that problem instance.)

Run No. (Problem instance)	Priority Indices							
	1	2	3	4	5	6	7	8
	min (di)	max (pi)	max ($pi/(di-pi-t)$)	max ($pi+Wi$)	min ($pi+Wi$)	min ($di-pi-t$)	min (pi/Wi)	min (pi)
0	81.28	91.85 (S_W)	74.29	83.54	74.88	83.47	78.82	71.00 (S_B)
1	81.40	92.69 (S_W)	73.92	84.66	71.33	84.66	78.87	65.79 (S_B)
2	72.96	90.13 (S_W)	63.97	76.58	67.37	76.58	70.42	58.65 (S_B)
3	63.34	71.44 (S_W)	61.60	66.77	61.78	66.78	61.93	55.43 (S_B)
4	67.05	77.80 (S_W)	62.63	70.17	61.22	70.14	65.34	55.77 (S_B)
5	74.12	84.81 (S_W)	70.60	77.32	70.60	77.28	70.94	62.82 (S_B)
6	67.30	74.62 (S_W)	59.20 (S_B)	68.80	61.21	68.80	65.76	61.05
7	61.64	66.21 (S_W)	58.24	63.25	56.87	63.16	60.11	54.00 (S_B)
8	73.72	82.10 (S_W)	68.83	77.62	65.17	77.57	71.16	61.76 (S_B)
9	61.75	65.61 (S_W)	59.54	63.73	59.00	63.72	60.08	55.23 (S_B)

The results of mean cycle time performance for various priority indices at $\rho = 0.75$ are presented in Table 3.2. In this table, the rows represent the individual runs and the columns represent the priority indices. For each run, each priority index gives a value of mean cycle time in the steady state simulation. Here each run of the simulation experiments represents a different randomly generated problem instance. It is to be noted that in this case the average value of the mean cycle time itself may not be a good measure to find out the best priority index, since the magnitudes of the solution values of a priority index depend on the problem instances (Kim et al., 1998). If average values are used, a problem instance in which (minimum and maximum) solution values are large has much more impact on the performance measure, the average value, than problems with small solution values. In other words, the average value of the mean cycle time for different problem instances may be biased by the problem data.

3. Pareto Control in Serial Processing

Table 3.3: RDI of various priority indices for Mean Cycle Time performance ($\rho = 0.75$)

Run No. (Problem instance)	Priority Indices							
	1 min (d_i)	2 max (p_i)	3 max ($p_i/(d_i-p_i-t)$)	4 max (p_i+Wi)	5 min (p_i+Wi)	6 min (d_i-p_i-t)	7 min (p_i/Wi)	8 min (p_i)
0	0.493022	1.000000	0.157786	0.601410	0.186082	0.598053	0.375042	0.000000
1	0.580194	1.000000	0.302177	0.701362	0.205911	0.701362	0.486159	0.000000
2	0.454509	1.000000	0.168972	0.569486	0.276962	0.569486	0.373835	0.000000
3	0.494147	1.000000	0.385447	0.708423	0.396692	0.709047	0.406062	0.000000
4	0.511993	1.000000	0.311371	0.653608	0.247372	0.652246	0.434377	0.000000
5	0.513894	1.000000	0.353813	0.659421	0.353813	0.657602	0.369276	0.000000
6	0.525432	1.000000	0.000000	0.622734	0.130385	0.622734	0.425535	0.120006
7	0.625965	1.000000	0.347394	0.757876	0.235146	0.750502	0.500608	0.000000
8	0.587909	1.000000	0.347535	0.779618	0.167623	0.777161	0.462069	0.000000
9	0.628139	1.000000	0.415227	0.818893	0.363203	0.817930	0.467251	0.000000
Av. RDI =	0.541520	1.000000	0.278972	0.687283	0.256319	0.685612	0.430021	0.012001

To resolve the bias due to randomly generated problem instances, Kim et al. (1998) used the *relative deviation index* (RDI) as a measure to compare the different priority indices. For each problem instance, the RDI of the priority index is computed as:

$$\text{Relative Deviation Index (RDI)} = \frac{(S_a - S_B)}{(S_W - S_B)} \quad (3.4)$$

where S_a , S_B , and S_W are solution values (mean cycle time) of the priority index a , the best priority index and the worst priority index, respectively, for that problem instance. Thus, the RDI value shows the relative quality of the solutions without being biased by the problem data. In each row of Table 3.2, S_B represents the best priority index and S_W represents the worst priority index for that problem instance. Table 3.3 presents the RDI values of mean cycle time performance at $\rho = 0.75$ for each priority index in the different runs. The average RDI of a priority index represents an unbiased measure to compare the priority indices with each other.

On the data in Table 3.3, first, the test of analysis of variance (ANOVA) is conducted in order to find out whether there is any significant statistical difference in the RDI values obtained from different priority indices. The results of ANOVA test on the RDI values for the three objectives at $\rho = 0.75$ and 0.85 are presented respectively in Figures B.1 and B.2 in the appendix. ANOVA test confirms the difference among the RDI values, obtained from different priority indices, to be significant for all the three objectives of mean cycle time, max. tardiness and cycle time variance. Then, Duncans multiple range test is conducted on the data in Table 3.3. Duncans multiple range test (Montgomery, 1997) is a statistical test widely used to classify sets of data into several groups according to the statistical differences in the data sets.

Table 3.4 gives the average RDI of ten problem instances for each priority index in an ascending order at both utilization levels, $\rho = 0.75$ and 0.85 . The table also gives the results of the Duncan's multiple range test. Using Duncan's test, the priority indices are classified into several groups (A, B, C, D, E, F). Between the priority indices within one group, there is no statistical difference in the mean cycle time performance at the significance level of 0.05. Based on these results under the specified conditions, it can be concluded that the mean cycle time objective is minimized by the priority index of $\min(p_i)$.

Similarly, ANOVA and Duncan's test are conducted on the RDI values for the objectives of maximum tardiness and cycle time variance. ANOVA results are presented in the appendix in Figures B.1 and B.2 for $\rho = 0.75$ and 0.85 respectively, which confirms the significant difference in performance of the different priority indices. The results of Duncan's test are presented in Tables 3.5 and 3.6 respectively for maximum tardiness and cycle time variance. These tables give the average RDI of ten problem instances for each priority index in an ascending order and also the group classification at both utilization levels, $\rho = 0.75$ and 0.85 .

3. Pareto Control in Serial Processing

Table 3.4: Steady state results of RDI and Duncan's test for Mean Cycle Time ($\bar{F}$)

Priority index	$\rho = 0.75$		$\rho = 0.85$	
	RDI	Duncan's test	RDI	Duncan's test
(8) $\min(p_i)$	0.012001	A	0.000000	A
(5) $\min(p_i + W_i)$	0.256319	B	0.245923	B
(3) $\max(p_i/(d_i - p_i - t))$	0.278972	B	0.275708	B
(7) $\min(p_i/W_i)$	0.430021	C	0.393034	C
(1) $\min(d_i)$	0.541520	D	0.503783	D
(6) $\min(d_i - p_i - t)$	0.685612	E	0.593938	E
(4) $\max(p_i + W_i)$	0.687283	E	0.594156	E
(2) $\max(p_i)$	1.000000	F	1.000000	F

[Duncan's test classifies the priority indices into several groups. Between the priority indices within one group, there is no statistical difference in the mean cycle time performance at the significance level of 0.05.]

Table 3.5: Steady state results of RDI and Duncan's test for Max. Tardiness (T_{max})

Priority index	$\rho = 0.75$		$\rho = 0.85$	
	RDI	Duncan's test	RDI	Duncan's test
(1) $\min(d_i)$	0.000908	A	0.001161	A
(4) $\max(p_i + W_i)$	0.006221	A	0.001296	A
(6) $\min(d_i - p_i - t)$	0.006236	A	0.001299	A
(7) $\min(p_i/W_i)$	0.035863	A	0.049174	A
(3) $\max(p_i/(d_i - p_i - t))$	0.372335	B	0.415989	B
(8) $\min(p_i)$	0.595498	C	0.623549	C
(2) $\max(p_i)$	0.773493	D	0.788514	D
(5) $\min(p_i + W_i)$	0.969161	E	0.995907	E

[Duncan's test classifies the priority indices into several groups. Between the priority indices within one group, there is no statistical difference in the maximum tardiness performance at the significance level of 0.05.]

3. Pareto Control in Serial Processing

Table 3.6: Steady state results of RDI and Duncan's test for Cycle Time Variance (σ_F^2)

Priority index	$\rho = 0.75$		$\rho = 0.85$	
	RDI	Duncan's test	RDI	Duncan's test
(7) $\min(p_i/W_i)$	0.008287	A	0.000000	A
(1) $\min(d_i)$	0.026494	A	0.011587	A
(6) $\min(d_i - p_i - t)$	0.061006	A	0.025499	A
(4) $\max(p_i + W_i)$	0.061739	A	0.025540	A
(8) $\min(p_i)$	0.202004	B	0.173389	B
(3) $\max(p_i/(d_i - p_i - t))$	0.326551	C	0.317463	C
(2) $\max(p_i)$	0.900243	D	0.938541	D
(5) $\min(p_i + W_i)$	0.924071	D	0.894486	D

[Duncan's test classifies the priority indices into several groups. Between the priority indices within one group, there is no statistical difference in the cycle time variance performance at the significance level of 0.05.]

From the results in Table 3.5, it can be stated that for minimizing maximum tardiness, there is no significant difference in the performance of Group A priority indices, *i.e.*, $\min(d_i)$, $\max(p_i + W_i)$, $\min(d_i - p_i - t)$ and $\min(p_i/W_i)$. Similarly, from the results in Table 3.6, it can be inferred that the performance of Group A priority indices, *i.e.*, $\min(p_i/W_i)$, $\min(d_i)$, $\min(d_i - p_i - t)$ and $\max(p_i + W_i)$ do not differ significantly in minimizing cycle time variance. This finding confirms the initial understanding that the objectives of minimizing maximum tardiness and minimizing cycle time variance are not conflicting to each other. Based on these results, it may be considered that the maximum tardiness is minimized by the priority index of $\min(d_i)$ and the cycle time variance is minimized by $\min(p_i/W_i)$. Therefore, for Pareto control in simultaneously minimizing mean cycle time and maximum tardiness, the priority indices of $\min(p_i)$ and $\min(d_i)$ are considered as scheduling criteria for these two objectives. Similarly, the priority indices of $\min(p_i)$ and $\min(p_i/W_i)$ are taken as scheduling criteria for Pareto control in minimizing mean cycle time and cycle time variance.

3.6 Results of Pareto Control in Serial Processing

As earlier stated in Section 3.2, the proposed concept of Pareto control in dynamic scheduling is evaluated for a serial processing machine with respect to the two pairs of conflicting objectives: (i) the mean cycle time and the maximum tardiness and (ii) the mean cycle time and the cycle time variance. In the following subsections, the results of Pareto control are presented with respect to each pair of conflicting objectives.

In applying weighted aggregation (WA) and compromise programming (CP) methods, the relative weights ($w_1, w_2; w_1 + w_2 = 1$) of the two criteria are varied linearly from value 0.05 to 0.95 at an interval of 0.05 in order to cover the whole range of weights, *i.e.*, 19 points are computed in total. In CP method, the exponent r is varied at the levels of 0.33, 0.66, 1, 2, 3 and 10. The CP curves for these exponent settings are correspondingly represented by CP0.33, CP0.66, CP1, CP2, CP3 and CP10. Among these, CP0.33 and CP0.66 are studied to see the effects of exponent value less than unity and similarly CP2 and CP3 for the effects of exponent value greater than unity. However, CP10 is studied as an approximation of Tchebycheff function, as earlier explained in Equation 2.18 (Koksalan, 1999).

3.6.1 Minimizing mean cycle time ($\bar{F}$) and maximum tardiness (T_{max})

As earlier discussed in Section 3.5, the scheduling objective of minimizing mean cycle time ($\bar{F}$) is achieved by the scheduling criterion of minimizing processing time (p_i) and the scheduling objective of minimizing maximum tardiness (T_{max}) is obtained by the scheduling criterion of minimizing due date (d_i). Thus, for this bi-objective scheduling problem, the two criteria used in Equations 2.14 and 2.15 are as follows:

$$f_1(x) = \text{minimize } (p_i), \quad f_2(x) = \text{minimize } (d_i) \quad (3.5)$$

3. Pareto Control in Serial Processing

Thus, using Equation 3.5, the Pareto objective function for the normalized WA method is formulated from Equation 2.14. Similarly based on Equation 2.15, the Pareto objective function for CP method is formulated.

WA method:

$$\min F(i) = w \cdot \frac{p_i}{\max_{\forall i}(p_i)} + (1 - w) \cdot \frac{d_i}{\max_{\forall i}(d_i)} \quad (3.6)$$

CP method:

$$\min F_r(i) = \left[\left(w \cdot \frac{|p_i - \min_{\forall i}(p_i)|}{\min_{\forall i}(p_i)} \right)^r + \left((1 - w) \cdot \frac{|d_i - \min_{\forall i}(d_i)|}{\min_{\forall i}(d_i)} \right)^r \right]^{1/r} \quad (3.7)$$

Here, $F(i)$ represents the WA function value for the i^{th} job and $F_r(i)$ represents the CP function value for the i^{th} job at the decision instance in the simulated time.

Then, among all the queued jobs in front of the machine, the best job is selected for loading on the serial processing machine. The job, which has the minimum value of Pareto objective function, $F(i)$ for WA method and $F_r(i)$ for CP method, is selected for loading at every decision instance in the simulation clock.

$$k = \arg \min_{\forall i} [F(i) \text{ or } F_r(i)] \quad (3.8)$$

where k implies the index of the job, which is selected for loading on the machine.

The results of these two conflicting objectives are presented for utilization factors of 0.75 and 0.85 respectively in Appendix C, Tables C.1 and C.2, for varying weights and exponent values in WA and CP methods. The results are plotted in two dimensional space with each objective on one axis. These plots are presented in Figures 3.4 and 3.5, which show the characteristic curves of Pareto control. The extreme points in these plots are achieved by the dispatching criterion represented in Equation 3.5.

In Figures 3.4 and 3.5, dispatching criterion $\min(p_i)$ gives the extreme point corresponding to minimization of mean cycle time and similarly dispatching criterion

Figure 3.4: Pareto control for Mean Cycle Time and Maximum Tardiness ($\rho = 0.75$)

Figure 3.5: Pareto control for Mean Cycle Time and Maximum Tardiness ($\rho = 0.85$)

$\min(d_i)$ gives the other extreme point corresponding to minimization of maximum tardiness. These two extreme points are the limits of Pareto control achievement, which are correspondingly referred to as p extreme and d extreme points.

In WA and CP methods, as the weight is varied linearly in a discrete fashion, different points are achieved on the boundary in a progressive manner from one end point to the other end point of the curve. From the observations in Figures 3.4 and 3.5, in comparison to the CP method, the WA method gives a short boundary between the weights 0.05 and 0.95. CP method has a double weighing scheme which gives it flexibility to cover the whole boundary from one end point to the other end point. In addition, the variation of exponent value in CP method gives it the higher flexibility of Pareto control between a specified range of objective functions.

In Figures 3.4 and 3.5, it may be questioned if there is any statistical significance to the differences in the graphs. To address this point, first it is to be noted that in these graphs each point is the average of 10 runs, where each run represents a different randomly generated problem instance. Then, to find out the statistical significance to the differences in the graphs, three consecutive points on CP1 curve in Figure 3.4 corresponding to weight (w) equal to 0.45, 0.50 and 0.55 are taken as a sample illustration. For these three consecutive points, the values of the two objectives, mean cycle time and maximum tardiness, are presented in Table 3.7 for each of 10 problem instances.

In Table 3.7, the values in each row are taken under homogeneous conditions (the same problem instance), but these conditions change from one problem instance to another. Therefore, instead of analyzing the difference between the average of 10 problem instances, the statistics of the mean of the differences in each problem instance must be computed and the paired t -test must be applied (Montgomery, 1997). Thus, the paired t -test is applied for the two consecutive pairs of points: (a) points with $w = 0.45$ and 0.55 and (b) points with $w = 0.50$ and 0.55 . These tests are presented in Table 3.8.

3. Pareto Control in Serial Processing

Table 3.7: Mean cycle time and maximum tardiness values for $w = 0.45, 0.50$ and 0.55 points on CP1 curve in Figure 3.4

Run no. (Problem)	$w = 0.45$		$w = 0.50$		$w = 0.55$	
	MeanCT*	MaxTar [†]	MeanCT*	MaxTar [†]	MeanCT*	MaxTar [†]
0	71.22	388.95	70.80	389.39	70.85	389.47
1	68.82	433.22	68.56	452.58	67.94	473.90
2	61.20	402.36	61.41	481.68	61.32	520.38
3	56.08	315.84	56.04	336.24	55.88	374.04
4	57.25	312.35	57.05	325.75	56.85	395.85
5	63.60	355.62	63.54	355.80	63.42	406.80
6	61.25	351.19	61.04	351.19	61.18	351.05
7	56.00	378.80	55.92	401.92	54.72	401.92
8	63.36	379.26	63.09	409.59	62.64	415.80
9	56.20	454.40	55.90	454.40	55.70	485.00
Average	61.50	377.20	61.34	395.85	61.05	421.42

[* MeanCT = mean cycle time (min), [†] MaxTar = maximum tardiness (min)]

From the results in Table 3.8, the 95% confidence intervals of the differences in mean cycle time between the points in two pairs are (0.037, 0.289) and (0.006, 0.564), which do not include zero. Also, the p -values (probability associated with a Student's paired t -test with a two-tailed distribution) are 0.017113 and 0.045911, both of which are less than 0.05. It confirms that the differences in the mean cycle times are statistically significant at 0.05 significance level. This also tells that as w increases from 0.45 to 0.50 to 0.55, there is a significant reduction in the mean cycle times. Similar observations are made for the differences in maximum tardiness. For the two pairs of points, the 95% confidence interval of the differences in maximum tardiness are (-35.910, -1.401) and (-42.982, -8.153) and the p -values are 0.037023 and 0.008927. This confirms that the maximum tardiness of the two points in each pair are significantly different at 0.05 significance level. With the increase in w from 0.45 to 0.50 to 0.55, the maximum tardiness significantly improves.

3. Pareto Control in Serial Processing

Table 3.8: Paired t -test for mean cycle time and maximum tardiness values between the two pairs of consecutive points on CP1 curve in Figure 3.4

Run no. (Problem)	Between $w = 0.45$ and 0.50		Between $w = 0.50$ and 0.55	
	d_MeanCT*	d_MaxTar [†]	d_MeanCT*	d_MaxTar [†]
0	0.42	-0.44	-0.05	-0.08
1	0.26	-19.36	0.62	-21.32
2	-0.21	-79.32	0.09	-38.70
3	0.04	-20.40	0.16	-37.80
4	0.20	-13.40	0.20	-70.10
5	0.06	-0.18	0.12	-51.00
6	0.21	0.00	-0.14	0.14
7	0.08	-23.12	1.20	0.00
8	0.27	-30.33	0.45	-6.21
9	0.30	0.00	0.20	-30.60
Average	0.16	-18.66	0.28	-25.57
Std. Dev.	0.1767	24.1218	0.3894	24.3455
Conf. Int.	(0.037, 0.289)	(-35.910, -1.401)	(0.006, 0.564)	(-42.982, -8.153)
p -value	0.017113	0.037023	0.045911	0.008927

[* d_MeanCT = difference in mean cycle times (min), [†] d_MaxTar = difference in maximum tardiness (min), Std. Dev. = standard deviation of the differences, Conf. Int. = 95% confidence interval of the differences, p -value = probability associated with a Student's paired t -test with a two-tailed distribution]

In Figure 3.4, all the points of CP0.33 are concentrated around p extreme point. For CP0.66, the points get spread over a wider range. CP1 gives the maximum spread of boundary between the extreme points. As the exponent value is increased to 2 and 3, the spread of the boundary gets reduced and all the points shift towards the side of d extreme point. For CP10, all the points more or less converge to d extreme point as clearly visible in the magnified view in Figure 3.4. For a clear presentation of this observation, the end points (corresponding to weights $w = 0.05$ and $w = 0.95$) of CP curves for various exponent values are presented in Figures 3.6 (a) and (b).

3. Pareto Control in Serial Processing

Figure 3.6: End points of CP for Mean Cycle Time and Maximum Tardiness ($\rho = 0.75$)

In Figure 3.6 (a), as the CP exponent value increases from 0.33 to 1, the points corresponding to $w = 0.05$ move further away from p extreme point. With further increase in exponent value from unity onwards, the points corresponding to $w = 0.05$ remain in the vicinity of d extreme point. Figure 3.6 (b) presents the contrary behavior of the $w = 0.95$ points with respect to the CP exponent value. Here, as the CP exponent value increases from 0.33 to 1, the points corresponding to $w = 0.95$ remain in the vicinity of p extreme point. While, with further increase in exponent value from unity onwards, the points corresponding to $w = 0.95$ move closer to d extreme point.

3. Pareto Control in Serial Processing

Based on these observations and similar observations from Pareto control curve for 0.85 utilization factor (see Figure 3.5), it can be inferred that for the small positive exponent value ($r \approx 0$), all the points are cornered to p extreme point with low spread of the boundary. As the exponent value is increased up to unity, the boundary spread increases to its maximum between the two extreme points. On further increase in exponent value ($r > 1$), the boundary spread starts decreasing and the points start converging towards d extreme point.

An intuitive explanation of such a behavior may be that as the value of the exponent r remains near unity, the two terms in the right hand side of Equation 3.7, $(w \cdot \frac{|p_i - \min(p_i)|}{\min(p_i)})$ and $((1 - w) \cdot \frac{|d_i - \min(d_i)|}{\min(d_i)})$, contribute equally. However, for $r \gg 1$, the maxima of these two terms dominates and for $r \ll 1$, the minima of these two terms prevails. This results in achieving the maximum breadth of the CP curve for $r = 1$, while shifting all points towards p extreme point for $r \ll 1$ and towards d extreme point for $r \gg 1$ as shown in Figure 3.6. However, the variation of curve with respect to the weight remains unpredictable as it will depend on the individual values of w, p_i and d_i for the arriving jobs. The exact mathematical correlation between w and r and their effect on achieving Pareto control is still open to further research.

As presented in Section 2.3, if the problem was of single stage multi-objective optimization, then CP and WA methods might have resulted in the same curves. But, since the problem is of multi-objective dynamic scheduling which is solved by selecting and scheduling a Pareto optimal job at each decision instance in the simulated time, this problem can be considered as a case of multi stage multi-objective optimization. That's why in Figures 3.4 and 3.5, CP and WA methods resulted in different Pareto control curves.

Further, a point can be raised that whether it is possible to get the same point through different r and w combinations in the CP method. Intuitively it can be explained that the CP method for various values might result in the same points in

3. Pareto Control in Serial Processing

Figures 3.4-3.5. That is, in Equation 3.7, for each w value at a particular value of r , there seems to be a w value (not necessarily the same) at a different value of r , that produces almost the same point in Figure 3.4. This is because under similar conditions, the same choices of Pareto jobs are made at the decision instances in the simulated time.

However, in Figure 3.4, if only one CP curve is plotted for the whole range of w ($w \in]0, 1[$) at a particular value of r , then the Pareto curve will become highly sensitive with respect to w value at some portions of the curve. For example in Figure 3.4, with w varying linearly from 0.05 to 0.95 at an interval of 0.05, the mean cycle time value ranges from 68.14 to 60.22 min for CP1 curve, while for CP2 curve it ranges from 70.27 to 63.28 min (please see Table C.1). Therefore, if mean cycle time value needs to be achieved between 68.14 and 70.27 min on CP1 curve, then w value needs to be varied in the interval $]0, 0.05]$. Similarly, if mean cycle time value needs to be achieved between 60.22 and 63.28 min on CP2 curve, then w value needs to be varied in the interval $[0.95, 1[$. In both of these cases, mean cycle time will become highly sensitive to the change in w value, *i.e.*, even very slight change in w will cause a high change in mean cycle time value. This kind of high sensitivity is not desirable for the practical purpose of shop floor management. Therefore, the double weighing scheme, w and r , in CP method is actually beneficial for the real-life applications of Pareto control in shop floor management.

In static problems of bi-objective scheduling, there exists Pareto optimal solutions and a Pareto boundary as shown in Figure 1.2. However, these are ill defined in the case of a bi-objective dynamic scheduling problem, which led to the development of the concept of Pareto control in MODS environment. Thus, the behavior of Pareto control curves for the conflicting objectives of minimizing mean cycle time and minimizing maximum tardiness in the dynamic scheduling of a serial processing machine has been investigated. The characterization of Pareto control curves gives an additional benefit

of the ability to control the scheduling objectives within a specified range. By selecting an appropriate exponent value, the decision maker can control the machine scheduling within the specific range of scheduling objectives.

3.6.2 Minimizing mean cycle time ($\bar{F}$) and cycle time variance (σ_F^2)

In the second bi-objective problem, the simultaneous minimization of mean cycle time and variance of cycle time is considered for the dynamic scheduling of a serial processing machine. From Section 3.5, the priority indices for individually minimizing mean cycle time and cycle time variance are considered as $\min(p_i)$ and $\min(p_i/W_i)$. These two criteria are used in Equations 2.14 and 2.15 as follows:

$$f_1(x) = \text{minimize } (p_i), \quad f_2(x) = \text{minimize } (p_i/W_i) \quad (3.9)$$

Using Equation 3.9, the Pareto objective function is formulated for the normalized WA method from Equation 2.14 and for CP method from Equation 2.15.

WA method:

$$\min F(i) = w \cdot \frac{p_i}{\max_{\forall i}(p_i)} + (1 - w) \cdot \frac{p_i/W_i}{\max_{\forall i}(p_i/W_i)} \quad (3.10)$$

CP method:

$$\min F_r(i) = \left[\left(w \cdot \frac{|p_i - \min_{\forall i}(p_i)|}{\min_{\forall i}(p_i)} \right)^r + \left((1 - w) \cdot \frac{|\frac{p_i}{W_i} - \min_{\forall i}(\frac{p_i}{W_i})|}{\min_{\forall i}(\frac{p_i}{W_i})} \right)^r \right]^{1/r} \quad (3.11)$$

Then, among all the queued jobs in front of the machine, the best job is selected for loading on the serial processing machine similar to the job selection procedure in Section 3.6.1. The job, which has the minimum value of Pareto objective function, $F(i)$ for WA method and $F_r(i)$ for CP method, is selected for loading.

$$k = \arg \min_{\forall i} [F(i) \text{ or } F_r(i)] \quad (3.12)$$

3. Pareto Control in Serial Processing

where k implies the index of the job, which is selected for loading on the machine.

The results of mean and variance of cycle times are presented in Tables C.3 and C.4 in Appendix C respectively for 0.75 and 0.85 utilization factors. These results show the values of the two objectives with respect to varying weights and exponent values in WA and CP methods. Figures 3.7 and 3.8 show the plots in two dimensional space with each objective on one axis. The extreme points of these characteristic curves are obtained by the dispatching criterion represented in Equation 3.9.

In Figures 3.7 and 3.8, dispatching criterion $\min(p_i)$ gives the extreme point corresponding to minimization of mean cycle time and the dispatching criterion $\min(p_i/W_i)$ gives the other extreme point corresponding to minimization of cycle time variance. These two extreme points of Pareto control achievement are correspondingly referred as p extreme and p/W extreme points.

In WA and CP methods, as the weight is varied linearly in a discrete fashion, different points are achieved in a progressive manner from one end point to the other end point of the curve. The performance of the methods is compared in terms of the Pareto dominance of points achieved by WA and CP methods corresponding to the equal weight. As earlier presented in Section 1.2, a point is said to be dominating over another point if it is strictly better on one objective while better or equal to on the second objective. For example, in Figure 1.1 point Z_0 is dominating over point Z_1 and also over point Z_2 .

From the observations in Figure 3.7, the WA and CP methods perform equally well for 0.75 utilization factor. Here, for some values of weights the WA points dominate the CP points and vice versa for some other values of weights. While in Figure 3.8, there is a consistent dominance by the CP points over the WA points for all values of weights. For example, the points in CP0.33 curve clearly dominate the corresponding points in WA curve. For 0.85 utilization factor, CP method performs better than WA method

Figure 3.7: Pareto control for Mean Cycle Time and Cycle Time Variance ($\rho = 0.75$)

Figure 3.8: Pareto control for Mean Cycle Time and Cycle Time Variance ($\rho = 0.85$)

3. Pareto Control in Serial Processing

in achieving Pareto control between mean cycle time and cycle time variance. Similar to the results presented in section 3.6.1, CP method has an additional advantage of having double weighing scheme (weight and exponent), which gives it greater flexibility to control the operation within a specified range of objectives. CP method gives better performance in achieving Pareto control in dynamic bi-objective scheduling. Therefore, here onwards, the emphasis is given on the CP method in characterizing the Pareto control curves in multi-objective dynamic scheduling.

In Figure 3.7, there is a peculiarity that although the p/W extreme point is corresponding to the individual minimization of cycle time variance objective, there are various points on WA and CP curves which show a lesser value than the p/W extreme point on the cycle time variance axis. This behavior simply indicates that the priority index $\min(p_i/W_i)$ is only an approximate minimizer of the individual objective of cycle time variance. However, it still solves the purpose of achieving Pareto control curves in dynamic single machine scheduling for minimizing the mean cycle time and the cycle time variance.

Unlike the results in Figure 3.4, in Figure 3.7, the points of CP0.33 curve give a wide spread of boundary between the p and p/W extreme points. For CP0.66, the spread of points gets reduced. Further, as exponent value in CP method increases, the spread of points gets reduced more and more. Figures 3.9 (a) and (b) show the end points (corresponding to weights $w = 0.05$ and $w = 0.95$) of CP curves for various exponent values. In Figure 3.9 (a), the points corresponding to $w = 0.05$ for various CP curves remain in the neighborhood of p/W extreme point, but goes away from p/W extreme point as the exponent value increases. In Figure 3.9 (b), the points corresponding to $w = 0.95$ for various CP curves move far away from p extreme point as the exponent value increases. Thus, the range of the achieved boundary gets reduced with the increase in the exponent value of CP.

Based on the observations of Pareto control for 0.75 and 0.85 utilization factor in

3. Pareto Control in Serial Processing

Figure 3.9: End points of CP for Mean Cycle Time and Cycle Time Variance ($\rho = 0.75$)

Figures 3.7 and 3.8, it can be inferred that as the exponent value increases, the spread of the boundary gets reduced. In addition, with the increase in CP exponent value, both the end points of Pareto control curve move away from the two extreme points, $\min(p_i)$ and $\min(p_i/W_i)$. The behavior of the end points of the CP curves in Figure 3.9 is not similar to the behavior observed in Figure 3.6.

The possible intuitive explanation for this behavior can be searched in the two terms in the right hand side of Equation 3.11, $(w \cdot \frac{|p_i - \min(p_i)|}{\min(p_i)})$ and $((1-w) \cdot \frac{|\frac{p_i}{W_i} - \min(\frac{p_i}{W_i})|}{\min(\frac{p_i}{W_i})})$. Unlike in the case of Equation 3.7, here in case of Equation 3.11 the two terms are not

completely independent. Here both the terms very much depend on p_i and therefore the value of the complete function very much depends on the relative values of these two terms. Thus, the possible explanation of such a behavior is still open to further research. However, the characterization of such Pareto trade-off curves and the investigation of the behavior of the Pareto control for simultaneous minimization of mean and variance of cycle times can be used to control these scheduling objectives within a specified range, by selecting an appropriate exponent value.

3.7 Effects of Parameter Variation on Pareto Control in Serial Processing

In Section 3.6, the results of Pareto control were presented for the bi-objective dynamic scheduling of a serial processing machine. In this section, the parameter settings of the due dates, weight patterns and the number of replications are changed and their effect on achieving Pareto control is observed. The results for the specific simulation studies are presented in following.

3.7.1 Effect of due date tightness

The simulation experiments are repeated for minimizing mean cycle time and maximum tardiness with loose due date settings. As described in Section 3.4, the due date of a job J_i is equal to the release time plus a multiple of the mean cycle time of the jobs in system before job J_i , that is $d_i = r_i + k \cdot \bar{F}_i$, where the constant $k > 0$ represents the due date allowance factor and $\bar{F}_i$ represents the mean cycle time of jobs in the system at the time of arrival of job J_i . $k = 1$ is considered as tight due date setting, while $k = 2$ as loose due date setting.

In the experimental results presented in Section 3.6.1, unity due date allowance factor is used. To study the effect of loosening due dates on achieving Pareto control,

Effect of Tight vs. Loose Due-dates on Pareto-controllability ($\rho = 0.85$)

Figure 3.10: Effect of tight vs. loose due-dates on Pareto control ($\rho = 0.85$)

the due date allowance factor is loosened to two. The corresponding Pareto control results are presented in Figure 3.10, in contrast with the previous results for 0.85 utilization factor.

From the observations in Figure 3.10, the only difference in the CP curves for the two settings of due dates is the shift towards the X-axis. It reflects a decrease in the maximum tardiness value for all the weights, which is due to the loosening of the due dates. However, the nature of the Pareto control curve repeats itself for the tight and loose due date settings.

3.7.2 Effect of non-linear weights

In Section 3.6, the results of Pareto control were presented with linear setting of relative weights. The CP points corresponding to these linear weights span the complete Pareto control curve. In order to test the achievability of Pareto control curve irrespective of the linear pattern of the weights, the weight pattern is changed to non-linear setting.

In this case, the nature of the relative weights is varied from linear setting to two non-linear settings and its effect is studied on the nature of Pareto control. In the experimental results of Section 3.6, the linear weights are taken from 0.05 to 0.95 at a constant interval of 0.05. To study the effect of non-linear weights, the weights are chosen in such a way that the interval between two consecutive weights changes in a systematic trend such as $\sin \theta$ and $\sin^2 \theta$.

The two non-linear settings correspond to the weights equivalent to $\sin \theta$ and $\sin^2 \theta$, where the angle θ varies from 5° to 85° at an interval of 5° . The non-linear settings $\sin \theta$ and $\sin^2 \theta$ are referred as Non-Linear 1 and Non-Linear 2 correspondingly. On a common scale, the weights of the linear and two non-linear settings are presented in Figure 3.11. From this figure, Non-Linear 1 setting gives more dense weights as the weights increase towards unity. While Non-Linear 2 setting gives symmetric weights

Figure 3.11: Relative weight patterns
 [Non-Linear 1 = $\sin \theta$, Non-Linear 2 = $\sin^2 \theta$]

around 0.5, which become more dense towards both extremes of 0 and 1.

The simulation results for these three settings of weights are presented in Figure 3.12 for CP1 in minimizing mean cycle time and maximum tardiness for 0.85 utilization factor. Both extremes of the curves are presented in a magnified view for clear presentation. From these observations, Non-Linear 1 weights increase the boundary length towards p extreme side while reducing the boundary on d extreme side. On the contrary, Non-Linear 2 weights increase the boundary on both extreme sides, *i.e.*, Non-Linear 2 weights generate points closer to the p and d extreme points for CP1. These results corroborate the fact that the weights of the objectives determine the position of the point in achieving Pareto control. Non-Linear 2 weights ($\sin^2 \theta$) enhance the Pareto control boundary on both sides of the curve, *i.e.*, p extreme and d extreme sides.

3.7.3 Effect of number of replications

In this chapter, for all the simulation experiments, the number of replications are kept to 10, following Kim et al. (1998, 2001). In these experiments, each replication represents a different randomly generated problem instance with different job arrival times, different due dates, etc. The run length is kept to 1000 jobs in the steady

Figure 3.12: Effect of non-linear weights on Pareto control ($\rho = 0.85$)

3. Pareto Control in Serial Processing

state. The average values of statistical estimates are plotted for achieving Pareto control curves. However, two questions may be raised: (a) Is the Pareto control curve achievable even for a single replication of the simulation experiment? and (b) How does the number of replications affect the statistical fluctuations in the Pareto control

(a) Number of replications = 1

(b) Number of replications = 10

3. Pareto Control in Serial Processing

(c) Number of replications = 30

(d) Number of replications = 100

Figure 3.13: Effect of varying the number of replications ($\rho = 0.75$)

curves? For example, in Figures 3.7 and 3.8, although the steady state data is collected for experiment outputs, still there are heavy fluctuations in the Pareto control curves.

To address these two questions, a series of simulation experiments is conducted

3. Pareto Control in Serial Processing

with 1, 10, 30 and 100 replications for 0.75 and 0.85 utilization levels as a continuation of Section 3.6.2. Due to the limited array size problem in coding, there is a limitation on the run length. Hence, to study the effects of statistical fluctuations the run length is kept same and only the number of replication is widely varied. The respective

(a) Number of replications = 1

(b) Number of replications = 10

3. Pareto Control in Serial Processing

(c) Number of replications = 30

(d) Number of replications = 100

Figure 3.14: Effect of varying the number of replications ($\rho = 0.85$)

Pareto control results of these experiments are plotted in Figures 3.13 and 3.14. In these figures, only p and p/W points and CP0.33 and WA curves are presented for the clarity of the pictures. The reason for using CP0.33 is that CP0.33 gives heavier

3. Pareto Control in Serial Processing

fluctuations than CP1 in Pareto control curves in Figures 3.7 and 3.8. Similarly, this set of experiments could have been presented for the case of minimizing mean cycle time and maximum tardiness (Section 3.6.1), but there the fluctuations in the Pareto control curves of Figures 3.4 and 3.5 are not as heavy as in the Pareto control curves of Figures 3.7 and 3.8. Therefore, in this subsection the set of experiments for the effect of varying the number of replications has been conducted on the case of minimizing mean cycle time and cycle time variance (Section 3.6.2).

Figure 3.13 (a) and Figure 3.14 (a) present the Pareto control curves for the single replication of the simulation experiments. But, there exist high statistical fluctuations in the curves of CP0.33 and WA methods. However, the point of achieving Pareto control curve by single replication has been addressed here.

In the sequence of Figures 3.13 (a), (b), (c) and (d) for 0.75 utilization level, it is observed that as the number of replications increases, the fluctuations in the Pareto control curves of CP0.33 and WA methods decrease and the curves become smooth. Similar observations are presented in Figures 3.14 (a), (b), (c) and (d) for achieving Pareto control curves at 0.85 utilization level. Therefore, taking 100 replications will give fairly smooth Pareto control curves as shown in Figures 3.13 (d) and 3.14 (d), but at the expense of much higher computational time. However, here it is to be stressed that even for one replication, the Pareto control can be achieved in the dynamic scheduling of a single machine. This is important from the point of the real-life application of Pareto control in the shop floor management for there may be often only one replication of these experiments.

From Figure 3.13 (d), it becomes reemphasized that at 0.75 utilization level the WA points dominate over the CP0.33 points for some values of weights; while the CP0.33 points dominate over the WA points for some other values of weights. Therefore, it can be said that the WA and CP methods perform equally well for 0.75 utilization factor. While, Figure 3.14 (d) presents that there is a consistent dominance by the

CP0.33 points over the WA points for all values of weights. Thus, it can be concluded that for 0.85 utilization factor the CP method gives better performance than the WA method in achieving Pareto control points.

3.8 Conclusion

This chapter presented the computational experiments for Pareto control in multi-objective dynamic scheduling of a serial processing machine. Conjunctive simulated scheduling approach is used to schedule the jobs on the machine in a dynamic setting, which fulfills the requirements of near-real-time schedules, especially in semiconductor manufacturing. The study showed that by varying the relative weights of the selection criteria at each selection instance in simulation clock, a Pareto control curve can be achieved for two conflicting objectives. The methodology includes first finding the selection criteria for optimizing an individual objective in dynamic scheduling and then forming a combined selection criterion using convex combination methods of WA and CP.

It has been shown how Pareto control can be achieved in scheduling a serial processing single machine using CP method in conjunctive simulated scheduling with respect to the two sets of conflicting objectives, (i) mean cycle time *vs.* maximum tardiness and (ii) mean cycle time *vs.* variance of cycle time, in a dynamic setting. Further, the sensitivity of these trade-off curves has been experimented for the variation in the due date tightness, the weight pattern and the number of replications in the simulation experiments. It has been found that irrespective of the variations in these parameters, the Pareto control can still be achieved between the conflicting objectives.

This chapter significantly contributes to the characterization of Pareto control curves for conflicting objectives, which can be exploited to enhance the ability to choose and operate the shop in a controlled way in semiconductor manufacturing. By

3. Pareto Control in Serial Processing

appropriate selection of weight range and exponent value, the conflicting scheduling objectives can be controlled and maintained within a specified range for a particular operation. Therefore, once the characterization of the Pareto control curves has been achieved, the choices of weight and exponent are left to the decision of the shop floor manager according to the desired operating range of the objectives.

Pareto control is highly relevant in a dynamic manufacturing environment, where the demands of objectives frequently changes according to the market forces. Further, it is shown that using CP method gives wider control over achieving the trade-off curves between the conflicting objectives as CP method involves double weighing scheme with parameters of weight and exponent. A real-life case study example on this concept is presented in Chapter 7 which shows the importance of using Pareto control over the factory heuristics. Due to the strong NP-hard nature of multi-objective dynamic scheduling problems, the true Pareto optimal frontier of these problems are left undefined. Therefore, Pareto control gives the feasible solutions to the multi-objective dynamic scheduling problems. However, the test of the optimality of these feasible solutions is beyond the scope of this thesis work.

In this chapter, the concept of CP in conjunctive simulated scheduling was applied to achieve the Pareto control in multi-objective dynamic scheduling of a serial processing machine. The next chapter presents the application of Pareto control in multi-objective dynamic scheduling of a batch processing machine.

Chapter 4

Pareto Control in Batch Processing

4.1 Introduction

In Chapter 2, the Pareto control concept was adopted for multi-objective dynamic scheduling problems, which was then applied to a serial processing machine in Chapter 3. In this chapter, the Pareto control concept is applied and computationally experimented for the bi-objective dynamic scheduling of a batch processing machine.

In semiconductor manufacturing, the back-end involves batching operations at burn-in ovens, which is a bottleneck process in the final testing stage. Most of the processes in semiconductor supply chain are driven by the needs of low cycle times and due date accuracy, which are contradicting to each other. Scheduling of the back-end batch processes is very important considering the productivity of the whole manufacturing process and customers' due-date requirements. This results in the need for schedule control of the semiconductor manufacturing processes to meet the frequent changes in requirements of the market demands. In this chapter, the concept of Pareto control is studied on a dynamic batch processing machine considering two objectives of minimizing mean cycle time and maximum tardiness.

Research reported on scheduling problems concerned with batch processing machines can be divided into two categories according to the batch processing time pat-

tern. In the first category, the processing time of each batch is fixed and independent of the jobs grouped together in the batch. In the second category, the processing time is dependent on the jobs assigned to each batch. In the semiconductor industry, such independent processing-time scheduling models are mainly applicable to batch processing machines in the wafer fabrication process such as oxidation furnaces. The dependent processing-time models, where the processing time of each batch is represented by the maximum processing time among those of all the jobs grouped together in the batch, are applicable to the burn-in oven process in the assembly and test.

In the back end burn-in operation, IC chips are loaded onto boards and are placed in an oven to expose them to high temperatures so as to get any failed IC chips screened out. The burn-in time of each IC chip is pre-specified and they vary based on type and the customer's requirements. The burn-in oven has a limited capacity, which requires the boards loaded with the chips to be sub-grouped into batches. It is possible for different IC chips to be processed in the oven simultaneously. The processing time of each batch is represented by the longest burn-in time among those of all the IC chips (jobs) grouped together in the batch. Hence, the scheduling problems for such an oven can be modeled as that for a general type of batch processing machine where each batch, loaded fully with jobs up to the oven capacity, can be processed simultaneously. The processing time of a batch is represented by the longest processing time of a particular IC chip in the batch (Sung et al., 2002).

Research on scheduling of burn-in ovens has been reported in the literature for only static problems. Lee et al. (1992) have proposed an optimal algorithm to minimize the number of tardy jobs and the maximum tardiness. Chandru et al. (1993b) have proposed a branch-and-bound algorithm and several heuristic algorithms for a problem of minimizing total completion time in a burn-in oven and also heuristics for a parallel oven system. Uzsoy (1994) has examined a problem of scheduling jobs with non-identical capacity volume requirements on a single burn-in oven to minimize total

completion time and makespan. However, to the best available knowledge, the problem has not been dealt with in a dynamic scheduling environment, where two conflicting objectives are simultaneously considered.

4.2 Model, Assumptions and Constraints

The model consists of a single batch processing machine, where the jobs are assumed to arrive dynamically at the machine and each job has its pre-specified processing time which may depend on the family type to which it belongs. The schematic diagram of the model has been presented in Figure 4.1. The available/queued jobs are grouped into a batch of c jobs. At each decision instance of loading, the Pareto jobs are sequentially selected from the queued jobs and the batch is processed simultaneously.

Figure 4.1: Schematic diagram of the model with batch processing machine

The assumptions and constraints can be summarized as follows:

1. The jobs are independent, *i.e.*, the jobs are processed without any precedence.
2. The jobs arrive dynamically at the machine.
3. The batch capacity is limited to c jobs per batch, *i.e.*, only up to c jobs can be processed simultaneously.
4. If the number of jobs in the queue is less than that of the batch capacity, all the available jobs are formed into a batch and processed simultaneously.

4. Pareto Control in Batch Processing

5. The processing time of a batch is represented by the longest job processing time among all the jobs contained in a batch.
6. No machine idle time and breakdowns; machine is always available for processing.
7. Setup times are included in the processing times.
8. No job preemption is allowed during each batch process and all the jobs belonging to the same batch are completed at the same time.
9. There are no other limiting resources such as material or labor.

The conflicting objectives of scheduling are:

1. Minimizing Mean Cycle Time ($\bar{F} = \sum_{i=1}^n F_i/n$) and
2. Minimizing Maximum Tardiness ($T_{max} = \max(T_1, T_2, \dots, T_n)$).

In batch scheduling, Brucker et al. (1998) proved that SPT indexing of jobs leads to an optimal schedule for minimizing total completion time in static case. This result is extended for the dynamic case of batch scheduling and experimented through simulation studies. It is found that choosing a job having the shortest processing time ($\min p_i$) at each decision instance minimizes mean cycle time ($\bar{F}$) in the dynamic scheduling of a batch processing machine. Similarly, it is experimentally found that selecting a job having the earliest due date ($\min d_i$) minimizes maximum tardiness (T_{max}) for the levels of utilizations (0.5, 0.6 and 0.7) considered in this study.

However, there is no definite selection rule for minimizing cycle time variance in the scheduling of a batch processing machine either in the static or in the dynamic case. Therefore, only one pair of conflicting objectives has been considered for achieving Pareto control in batch processing, unlike two pairs of conflicting objectives in serial processing in Chapter 4.

Thus, for achieving the Pareto control between the conflicting objectives of minimizing mean cycle time and minimizing maximum tardiness, CP method is applied for job selection at each decision instance in simulation clock. The two criteria used in CP method in Equation 2.15 are:

$$f_1(x) = \text{minimize } (p_i), f_2(x) = \text{minimize } (d_i) \quad (4.1)$$

Thus, using Equation 4.1, the Pareto objective function for the CP method is formulated from Equation 2.15 as follows:

$$\min F_r(i) = \left[\left(w \cdot \frac{|p_i - \min_{\forall i}(p_i)|}{\min_{\forall i}(p_i)} \right)^r + \left((1 - w) \cdot \frac{|d_i - \min_{\forall i}(d_i)|}{\min_{\forall i}(d_i)} \right)^r \right]^{1/r} \quad (4.2)$$

Since, the selection of jobs for making a batch is required only when there are more than c jobs in the queue. In such case, among all the queued jobs in front of the machine, c jobs are selected for loading on the batch processing machine. c jobs are selected based on the minimum value of Pareto objective function, $F_r(i)$.

4.3 Batch Scheduling Methods

This section presents two batch scheduling methods: (i) Natural Batch Scheduling (NBS) and (ii) Minimum Batch Size (MBS), which are common in industrial practices. These methods consider only the queued jobs in front of the batch processing machine for making a batch. These two methods are taken for the purpose of comparison and analysis of the results of Pareto control in batch processing.

Natural batch scheduling (NBS)

The most natural and simple way of scheduling a batch is that whenever the batching machine becomes idle; it immediately starts the processing of the next batch if there

are jobs available in the queue in front of the machine. If the available queue length is greater than or equal to the batch capacity, then the formed batch is full and the job selection is done by some simple priority rules such as shortest processing time (SPT), earliest due date (EDD), *etc.* Else, a partial batch is loaded on the machine. In case of no job present in the queue, the machine waits for the future job arrival. This way of scheduling a batch is addressed here as Natural Batch Scheduling (NBS) strategy, which is called as Greedy Batching by Glassey and Weng (1991). For clarity, the NBS algorithm is presented in Figure 4.2.

```

If the batching machine becomes idle, then
  if there are jobs in the queue ( $q > 0$ ), then
    if  $q \leq c$ 
      Start processing a batch of  $q$  jobs;
    else
      Start processing a batch of  $c$  jobs based
      on any priority rule (SPT, EDD, etc.)
    endif
  else
    wait until a job arrives;
    apply the procedure again;
  endif
endif.

```

Figure 4.2: Natural Batch Scheduling (NBS) rule

Minimum batch size (MBS)

The Minimum Batch Size (MBS) strategy means that a number is picked such that an operation is started when the number of waiting jobs is greater than or equal to this number (Neuts, 1967). This is to say that if the machine is idle and the number of waiting jobs is smaller than MBS, these jobs will not be processed. The possible value of MBS is greater than or equal to 1 and is smaller than or equal to the batch

capacity, c . The maximum number of jobs that can be processed in a batch is the minimum of the queue length and the batch capacity. The MBS algorithm is presented in Figure 4.3.

```

If the batching machine becomes idle, then
  if there are jobs in the queue ( $q > 0$ ), then
    if  $q < MBS$ 
      Wait for future jobs to arrive;
    elseif  $q \leq c$ 
      Start processing a batch of  $q$  jobs;
    else
      Start processing a batch of  $c$  jobs based
      on any priority rule (SPT, EDD, etc.)
    endif
  else
    wait until a job arrives;
    apply the procedure again;
  endif
endif.

```

Figure 4.3: Neuts's Minimum Batch Size (MBS) rule (Neuts, 1967)

NBS rule is a special case of MBS rule with $MBS=1$ *i.e.* NBS is equivalent to MBS1. Therefore, in further discussions, NBS is not referred to as such independently; but is covered within MBS wherever it is required for comparison.

In using MBS algorithm for batch scheduling, if the number of jobs in the queue are more than the batch capacity (c) then simple priority rules are applied to select c jobs from the queue in order to load a full batch. For bi-objective scheduling of a batch processing machine, the MBS algorithm is modified at the step of selecting c jobs from the queue. Instead of using simple priority rules, CP method is applied to select c jobs from the queue to form a full batch.

4.4 Computational Experiments for Pareto Control in Batch Processing

In this study, a single batch processing machine is considered for dynamic scheduling, where jobs arrive continuously at discrete times. Table 4.1 summarizes the factors considered in the design of the simulation experiments.

Table 4.1: Factors for simulation experiments

No.	Factor	Levels	Description
1.	Utilization Level (ρ)	Three	0.5, 0.6, 0.7
2.	Processing Time (p_i)	Uniform	Mean (μ) = Range = 30 min, <i>i.e.</i> , $p_i \in \text{U}[15,45]$
3.	Interarrival Distribution	Exponential	Mean ($\lambda = \rho \cdot c\mu$)
4.	Due Date (d_i)	One	$d_i = r_i + \bar{F}_i$
5.	Conflicting Objectives	One	Minimizing ($\bar{F}, T_{max}$)
6.	Batching Strategy	Six	MBS1 - MBS6

For the simulation experiment, first the jobs are generated with their processing time, due date and arrival time. The processing times of jobs are assumed to be uniformly distributed with a mean and range of 30 time units, *i.e.*, the lower and upper limits of the interval are set at a distance of half of the mean from the mean of

4. Pareto Control in Batch Processing

the distribution. The arrival process is assumed to be exponentially distributed with the mean being controlled by the utilization factor.

For the single batch processing machine, the utilization factor can be related to the arrival rate as:

$$\rho = \lambda / (c \cdot \mu) \quad (4.3)$$

where λ is mean arrival rate, μ is mean service rate and c is the batch capacity. Hence, keeping the service rate at a constant value for all our experiments, the arrival rate is set to depend on the utilization factor. Three different utilization levels, viz., 0.5, 0.6 and 0.7 are considered for normal operation of the machine. At utilization level below 0.5, there is no variation in the schedule irrespective of different scheduling strategies due to very poor loading. However, at utilization level above 0.7, the system shows unsteady performance of cumulative average cycle time. It is due to the fact that the batch processing time is represented by the longest job processing time among all the jobs contained in a batch. This leads to the loss of the utilization capacity of the batch processing machine and thus the unsteady queue in front of the machine.

Due-dates are assigned to the jobs based on their arrival on the machine and average system cycle time with unity due-date allowance factor as:

$$d_i = r_i + \bar{F}_i \quad (4.4)$$

where r_i is the arrival time of job i and $\bar{F}_i$ is the system mean cycle time at the time of release of job i . The batch scheduling method is varied at six different settings according to MBS1, MBS2, MBS3, MBS4, MBS5 and MBS6.

Each simulation experiment consists of 10 different runs (or replications) and in each run the shop is loaded continuously with jobs that are numbered on their arrival. In order to make certain that the system is studied under the steady state, the mean of the cycle times is continuously observed. It is found that the system reaches a

4. Pareto Control in Batch Processing

steady state approximately after the arrival of about 1500 jobs for all runs. Figure 4.4 presents the sample graph of the steady state of cumulative average cycle time with MBS 1 scheduling strategy and CP method with power unity at 0.60 utilization level.

Figure 4.4: Steady state of cumulative average cycle time with MBS 1 scheduling strategy and CP method with unity power at 0.60 utilization level

The number of replications is fixed at 10 and the run length is kept as 4000 jobs in order to have a smaller number of replications and a larger run length as suggested by Law and Kelton (1984). In Figure 4.4, the steady state is achieved approximately after the completion of 1800 jobs. Thus, to be on safe side, the data for computation of statistics is collected from job completions, numbering from 2001 to 6000. The machine is continuously loaded till the completion of all the 4000 jobs. The simulation program is written in C++, implemented in Turbo C++ environment and run on an Intel 2000 Pentium (4), 1.8 GHz PC.

4.5 Results of Pareto Control in Batch Processing

This section presents the analysis of the results of achieving Pareto control for minimizing the conflicting objectives of mean cycle time ($\bar{F}$) and maximum tardiness (T_{max}) in dynamic scheduling of a batch processing machine. In Section 3.8, it is concluded that using the CP method gives wider control over achieving the boundary solutions of objectives in dynamic scheduling of a serial processing machine as CP involves double weighing scheme with parameters of weight and exponent. Therefore, in the simulation experiments of Pareto control in batch processing, only CP method is used to select the jobs for making a batch.

In case of Pareto control in serial processing (Section 3.6), CP method was used to select a job for scheduling at each decision instant in simulation clock. For Pareto control in batch processing, MBS method is used for batch scheduling. In case, the number of queued jobs becomes greater than the batch capacity, c , CP method is used to select c jobs to form a full capacity batch. Here, only the results of NBS, *i.e.*, MBS1 method are presented. In CP method, the relative weights ($w_1; w_2; w_1 + w_2 = 1$) of the two criteria are varied linearly from value 0.1 to 0.9 at an interval of 0.1 in order to cover the whole range of weights, *i.e.*, 9 points are computed in total. In CP method, the exponent r is varied at the levels of 0.33, 0.66, 1, 2, 3 and 10. The CP curves for these exponent settings are correspondingly represented by CP0.33, CP0.66, CP1, CP2, CP3 and CP10 in graphs. Among these CP0.33 and CP0.66 are studied to see the effects of exponent value less than unity and similarly CP2 and CP3 are studied for the effects of exponent value greater than unity. CP10 is studied as an approximation of Tchebycheff function (Equation 2.18) (Zeleny, 1982).

The two conflicting objectives under study are the minimization of mean cycle time and the minimization of maximum tardiness. The results of these objectives are presented for utilization factors of 0.5, 0.6 and 0.7 respectively in Figures 4.5, 4.6

Figure 4.5: Mean Cycle Time vs. Maximum Tardiness for dynamic scheduling of a batch processing machine ($\rho = 0.5$, MBS = 1)

Figure 4.6: Mean Cycle Time vs. Maximum Tardiness for dynamic scheduling of a batch processing machine ($\rho = 0.6$, MBS = 1)

Figure 4.7: Mean Cycle Time vs. Maximum Tardiness for dynamic scheduling of a batch processing machine ($\rho = 0.7$, MBS = 1)

and 4.7 for varying weights and exponent values in CP method. The points on the characteristic curve show the values of performance measures obtained for different levels of weights for dynamic single batch processing machine. The extreme points in these plots are achieved by the selection criterion presented in Equation 4.1. The selection criterion $\min(p_i)$ gives the extreme point corresponding to minimization of mean cycle time and similarly selection criterion $\min(d_i)$ gives the other extreme point corresponding to minimization of maximum tardiness. These two extreme points are the limits of control, which are correspondingly referred to as p extreme and d extreme points. The weight is varied linearly in a discrete fashion and different points are achieved on the boundary in a progressive manner from one end point to the other end point of the curve.

The points of CP0.33 are concentrated around p extreme point. The points of CP0.66 get spread over a wider range. The maximum spread of the control boundary between the two extreme points is achieved by CP2 at the utilization levels of 0.5 and 0.6 and by CP1 at 0.7 utilization level. As the exponent value is increased to 3, the spread of the boundary gets reduced and all the points shift towards the side of d extreme point. Further, at a very high exponent value such as 10, all the points get clustered around d extreme point. Thus, first with an increase in the exponent value, the boundary spread of Pareto control curve increases to its maximum value. On further increase in the exponent value, the boundary spread starts decreasing and the points start converging towards the d extreme point. The intuitive explanation of this behavior is similar to the intuitive explanation presented for Pareto control in serial processing in Section 3.6.1.

From this characterization of Pareto control curves, it is clear that the exponent value of CP method has a direct correlation with the spread of trade-off curves between the conflicting objectives of minimizing mean cycle time and minimizing maximum tardiness. This behavior of Pareto control curves gives scope for controlling the scheduling

objectives within a specified range. By selecting an appropriate exponent value, the decision maker can control the batch scheduling within the specific range of scheduling objectives.

The Pareto control curves for MBS1 with CP1 are presented in Figure 4.8 for various levels of utilization (ρ) considered in this study, *i.e.*, 0.5, 0.6 and 0.7. From this figure, it is clearly observed that at utilization level below 0.5, the curve will simply converge to a single point. This is because, when the arrival of jobs is very light, there is very low probability that there will be more than c jobs (batch capacity) in the queue at any decision instance and so there will be no variation in the schedule while using various weights in applying CP method for job selection. As the utilization level increases up to 0.7, the spread of steady state values of two objectives gets widened. At utilization level above 0.7, the arrival of jobs becomes very heavy and the jobs start accumulating in front of the machine. Thus, the queue becomes larger and larger, eventually leading to an unsteady state of the system.

4.6 Effects of Parameter Variation on Pareto Control in Batch Processing

The effect of varying due-dates and weight pattern on achieving Pareto control in bi-objective dynamic scheduling of a batch processing machine is found similar to the case of serial processing machine, as discussed in Section 3.7. Without being repetitive, the effect of varying the MBS value on the Pareto control in dynamic batch scheduling is addressed herein.

The simulation experiments of multi-objective dynamic batch scheduling are repeated for various MBS values at all the three utilization levels. The results of CP method with unity exponent are presented in Figures 4.9, 4.10 and 4.11 respectively for 0.5, 0.6 and 0.7 utilization levels. In these figures, each curve (MBS1 - MBS6) will

Figure 4.8: Effect of ρ variation on range of Pareto control between Mean Cycle Time and Maximum Tardiness

Figure 4.9: Effect of MBS value variation on Pareto control between Mean Cycle Time and Maximum Tardiness for dynamic scheduling of a batch processing machine for CP1 at $\rho = 0.5$

Figure 4.10: Effect of MBS value variation on Pareto control between Mean Cycle Time and Maximum Tardiness for dynamic scheduling of a batch processing machine for CP1 at $\rho = 0.6$

Figure 4.11: Effect of MBS value variation on Pareto control between Mean Cycle Time and Maximum Tardiness for dynamic scheduling of a batch processing machine for CP1 at $\rho = 0.7$

have its own extreme points on both sides. For the purpose of clarity of figures, these extreme points are not shown in these figures, as these extreme points are not relevant for the discussion. Here, the focus is to present that the Pareto control curves can be achieved for any MBS value.

At 0.5 utilization level, MBS curves are clearly separated as presented in Figure 4.9. MBS6 gives very poor performance with respect to both the objectives, since it requires full batch loading. At light job arrival, it incurs high waiting time to load the full batch. MBS1 gives the best performance on both objectives because at light arrival rate, it is preferable to load the available batch than to wait for more jobs to arrive. At 0.6 utilization in Figure 4.10, MBS curves come closer to each other. While, at 0.7 utilization in Figure 4.11, all MBS curves merge leaving no clear distinction. This is because, at heavy job arrival, the queue length is always greater than the batch capacity, c , and so at every decision instance a batch of c jobs is loaded on the machine thus leaving no differentiation among the various MBS curves.

4.7 Conclusion

This chapter presented the computational experiments for Pareto control in multi-objective dynamic scheduling of a batch processing machine. It has been shown that by using CP method, Pareto schedule control can be achieved in dynamic scheduling of a batch processing machine with respect to the conflicting objectives of minimizing mean cycle time and minimizing the maximum tardiness.

Conjunctive simulated scheduling approach is used at each decision instance in simulation clock. A batch is formed by using the MBS method of batching and in each batch the Pareto optimal jobs are selected using the CP method. The study showed that by varying the relative weights of selection criteria at each selection instance in simulation clock and the exponent value in CP method, a Pareto controllable curve can

be achieved for the conflicting objectives. By appropriate selection of weight range and exponent value, the conflicting scheduling objectives can be controlled and maintained within a specified range of operation. Further, it has been experimentally found that the Pareto control curve is highly sensitive to the variation in MBS value at low levels of utilization (0.5 and 0.6); while it becomes insensitive to the variation in MBS value at high utilization level (0.7).

However, in most of the CIM oriented production environments, especially at semiconductor factories, real-time information about the incoming jobs is easily available from the shop floor control systems, which can be exploited to significantly improve the batch scheduling performance. Generally in batch processing, use of the predicted future arrival data gives a significant improvement in the cycle time performance, as shown by Glassey and Weng (1991) and Fowler et al. (1992). Extending the use of available real-time information about the incoming jobs, a look ahead batching method is proposed in the next chapter in order to optimize the earliness and tardiness related objectives. These earliness and tardiness related objectives are of utmost importance in the JIT manufacturing environment.

Chapter 5

Single Objective Look Ahead Batch Processing

5.1 Introduction

In Chapter 2, the Pareto control concept was adopted for multi-objective dynamic scheduling problems, which was computationally experimented with serial processing and batch processing respectively in Chapters 3 and 4. In this chapter, a look ahead batch processing method is proposed, which acts as a prerequisite for applying the concept of “Pareto control” in look ahead batch processing in Chapter 6.

In Chapters 3 and 4, it was assumed that at any decision instance, a decision maker has information only about the arrived/queued up lots, *i.e.*, there is no real-time information available about the arrival epochs of lots arriving in the near-future. However, in most of the CIM oriented production environments, the real-time information about the arrival epochs and the due dates of incoming lots is easily available from shop floor control systems, but is seldom exploited in decision-making. This real-time information of incoming lots in near-future time can significantly improve the scheduling performance, especially for batch processing workstations where multiple jobs are simultaneously processed. “To start the machine now or to wait for the next lot to arrive” is the question, which stresses the essence of the control task for many batch

5. Single Objective Look Ahead Batch Processing

processing systems.

In the furnace model of batching, when a batch-processing machine is available and full loads are waiting, the decision to process a selected batch is obvious, since no advantage in cycle time performance can be gained by waiting for the incoming lots in the near-future. However, when there is a batch-processing machine available and only partial loads of products are waiting, a decision must be made to either start a (partial) batch or wait for more lots to arrive. In such situations, when the total number of lots in the queue is less than the capacity of the batch processing machine, starting a batch immediately underutilizes the batch processing machine. Delaying the initiation of processing until more lots arrive increases the queuing time for the lots that are currently awaiting processing and creates idleness of the machine. Both decisions may lead to deterioration of the schedule performance. Therefore, the dynamic optimization of the number of lots and starting time of the batch for loading at each decision instance are very essential.

To optimize the tradeoff between the number of lots and starting time of the batch, there exist look-ahead methods such as Dynamic Batching Heuristic (DBH) and Next Arrival Control Heuristic (NACH), as discussed earlier in Section 2.2. These methods exploit the arrival information of the incoming lots and minimize the mean cycle time performance in dynamic batch scheduling. However, the existing scheduling literature lacks any look-ahead batch scheduling strategy for optimizing due date related objectives, which are gaining increasing importance in the JIT manufacturing environment. In this chapter, a Look Ahead Batching (LAB) method is proposed, which utilizes the arrival and due date information of the upcoming lots and evaluates the optimal batch decision to achieve ET performance in dynamic batch scheduling.

The next section presents DBH and NACH methods for improving mean cycle time performance in dynamic batch scheduling. The proposed LAB method for improving ET performance is described in the succeeding section.

5.2 Existing LAB Methods for Cycle Time Performance

As presented in Section 4.3, NBS and MBS batch scheduling methods do not take advantage of forecast arrival epochs if such information is available from the computerized systems in the CIM environment. In this section, two look ahead batch scheduling methods, DBH and NACH, are presented, which take advantage of the predictions of future arrival in order to minimize the mean cycle time at a batch processing machine.

Dynamic Batching Heuristic (DBH)

Dynamic Batching Heuristic (DBH) (Glassey and Weng, 1991) utilizes the available information of future arrival epochs in order to decide whether the batch should start now or wait for more lots to come and join the present batch. DBH algorithm is presented in Figure 5.1.

```

If the batching machine becomes idle, then
  if there are lots in the queue, then
    let  $t_0$  be the time epoch that the batching machine becomes idle;
    start decision heuristic (explained in the text);
  else
    wait until a lot arrives and let  $t_0$  be the arrival epoch;
    start decision heuristic (explained in the text);
  endif
endif.

```

Figure 5.1: Glassey and Weng's Dynamic Batching Heuristic (DBH) algorithm

DBH considers the future arrivals that will occur in a fixed planning horizon equal to the batch processing time (T). DBH makes a decision whether to start a partial batch at the current time (t_0) or to wait for the i^{th} future arrival (at time t_i). For each

5. Single Objective Look Ahead Batch Processing

t_i in the planning horizon, the net delay caused by waiting for the arrival is determined. Net delay is the amount of time saved for the arrivals after t_0 but before $t_0 + T$ minus the delay incurred by those lots already waiting (q = the number of lots in queue at epoch t_0). Among all of the possible loading times as presented in Figure 5.2, the one to be chosen should have the largest improvement, *i.e.*, wait until time t_i where

$$i = \arg \max_{0 \leq j \leq j_{max}} \{j \times (T + t_0 - t_j) - q \times (t_j - t_0)\} \quad (5.1)$$

where j_{max} is the last future arrival considered in the look-ahead period.

Figure 5.2: Cumulative number of lots in queue through time

In Equation 5.1, the first term, $j \times (T + t_0 - t_j)$, represents the decrease in waiting time for the arrivals after t_0 but before $t_0 + T$, if the decision is to load the batch at time t_j . The second term, $q \times (t_j - t_0)$, implies the increase in waiting time of the already waiting lots at time t_0 for the same decision. Thus, the whole term represents the savings in the waiting time of lots for the decision of loading the batch at time t_j . The future arrival time with maximum savings in waiting times is selected as the final loading instance. Ties are broken in favor of earliest start time. Glassey and Weng concluded that the degree of improvement is affected by the middle traffic intensity levels. At low or high levels, DBH often yields the same decisions as the MBS rule.

Next Arrival Control Heuristic (NACH)

Glassey and Weng (1991) pointed out that about half of the potential benefit in using DBH is gained by looking ahead at only the next arrival. Fowler et al. (1992) modified DBH to consider only the next arrival and to make the decision whether to start the batch now or wait for the next arrival. In case of starting the batch now, the next decision is made at the time of batch completion. In case of waiting for the next arrival, another decision is made at the time of the next arrival. Thus, the modified heuristic is a rolling horizon policy. This contrasts with DBH which decides at time t_0 that how many arrivals should occur before starting a batch. They called their modified heuristic as Next Arrival Control Heuristic (NACH). Thus, modifying Equation 5.1 only for the next one arrival, *NACH* value is defined as follows:

$$NACH = (T + t_0 - t_1) - q \times (t_1 - t_0) \quad (5.2)$$

If $NACH < 0$, then it is beneficial to start the batch now for processing. Else, it is desirable to wait till the next arrival and apply the decision logic again at that point. Figure 5.3 provides a simplified flowchart of the logic of the simulation program. There are two possible decisions: (1) when a lot arrives (a “push” decision); and (2) when the batching machine finishes processing (a “pull” decision).

However, both DBH and NACH methods are developed to improve the cycle time performance at a batch processing machine, taking benefit of the predicted information about future arrivals. In the following section, a new LAB method is proposed for a single product single batch processing machine case with the objective of improving the ET performance. The proposed LAB method exploits the predicted information about arrival time and due dates of incoming lots and optimizes the ET performance.

Figure 5.3: Flowchart of simulation logic of Fowler et al.'s Next Arrival Control Heuristic (NACH) algorithm

5.3 Proposed LAB method for ET Performance

In LAB, the near-future arrivals of the incoming lots are looked ahead in the simulated time. First, all the LAB scenarios are formed from the arrived (queued-up) lots and the future incoming lots. A LAB scenario is defined as a batch of several lots equivalent to either partial or full batch capacity, which can be processed only after the arrival of all the lots in the batch at the machine. Lots for every LAB scenario are selected using arrival and due date information of the lots. Then, the best LAB scenario in terms of due-date objectives is selected to schedule on the available batch processing

5. Single Objective Look Ahead Batch Processing

machine for processing. The batch processing either starts immediately or after some delay till the arrival of selected future lots in the best LAB scenario. It depends on whether the best LAB scenario is formed from all available (queued-up) lots or from the future incoming lots at the machine.

The overall flow chart of the proposed LAB method is presented in Figure 5.4. In brief, the overall procedure is as follows.

1. Identify the lots arrived and to be arrived at in the near-future at the decision point in the simulation clock.
2. Construct the LAB scenarios based on the future arrivals of the lots.
3. Find out the best LAB scenario based on ET performance measure.
4. Process the best LAB scenario on the machine and advance the simulation clock to the next decision instance.

LAB algorithm has two stages of functioning. In the first stage, there is lot selection for forming various LAB scenarios with the consideration of incoming lots in the near-future. In the second stage, the best LAB scenario is selected to process on the batch processing machine. All the four steps of the LAB algorithm are separately elaborated in the following sub-sections in detail.

Step 1

A decision point in the simulation clock is defined as the availability of the machine or as the arrival of a new job on the idle machine. The decision of which batch should be loaded on the available machine needs to be made. First, at each decision point in the simulation clock, the lots arrived and the future incoming lots are identified based on the arrival information from the upstream process of manufacturing.

5. Single Objective Look Ahead Batch Processing

Figure 5.4: Flow chart of the simulation logic of LAB algorithm for ET performance

5. Single Objective Look Ahead Batch Processing

Step 2

At any decision point, in general, a batch is formed from the available (arrived) lots in front of the machine. However, there may be cases of a lot with tight due date arriving in the near-future. Since a batch under processing can not be stopped in between, the future lot with urgency has to wait up to the completion of the running batch on the machine in order to be processed next. Therefore, there is a need for looking ahead in future time for the arrival of any lot with tight due-date. However, the question is how much time to wait and how many lots to select for forming a batch. To deal with this, a LAB scenario is defined as the consideration of future arrivals while forming a batch at each decision instance in the simulation clock. This means the possibility of keeping the machine idle until future arrival(s), forming a batch considering these future arrivals and then loading the constructed batch for processing on the machine in a future time.

Since the lots of the same product type have equal processing times, in constructing the LAB scenarios, the maximum waiting time for future lots to arrive will always be less than one batch processing time of that product type. This is so because if the tight lot arrives after one batch processing time from now, then it might as well be processed after the completion of the present batch on the machine.

First, there is a need to construct the LAB scenarios for future period equivalent to one batch processing time. In each LAB scenario, the batch will consist of the number of lots equivalent to the minimum value between the number of lots arrived up to simulated time in consideration and the batch capacity. In case the number of lots arrived is greater than the batch capacity, the appropriate lots need to be selected to form full capacity batch based on due date. The construction of LAB scenarios is presented by an example in Figure 5.5. In Figure 5.5, Scenarios 1, 2, and 3 are formed of 4, 5, and 6 lots respectively, according to the number of lots arrived till

5. Single Objective Look Ahead Batch Processing

that simulated instance. Assuming batch capacity to be 6 and batch processing time to be 2 hours, scenario 4 is formed of 6 best lots out of 7 arrived lots, based on due date. In constructing these LAB scenarios, 8th lot is not considered for batching as it arrives after one batch processing time period from simulation clock. At the end of this step, we will have various LAB scenarios being considered for loading on the batch-processing machine.

Figure 5.5: An example of constructing look ahead batching (LAB) scenarios

Step 3

Among constructed LAB scenarios in the above step, the best LAB scenario is selected for loading on the batch processing machine. The scenario, which has the minimum root mean square (RMS) value of the earliness-tardiness measure of the lots, is selected as best for loading.

$$k = \arg \max_{0 \leq j \leq j_{max}} \left(\sqrt{\sum_{i=1}^{q+j} [(E_i)^2 + (T_i)^2]} \right) \quad (5.3)$$

where j is the indicator of LAB scenario, i represents the lots in each LAB scenario and k implies the best LAB scenario to be loaded for processing on the batching machine.

Step 4

Finally, the best LAB scenario will be loaded on the batch processing machine. The simulation clock of the machine is forwarded by the waiting time along with the setup and processing times. Then, it is required to find the next availability of machine in order to forward the simulated time up to next simulation event. Then, the whole procedure is repeated from step 1 again for batch decision making on the simulated instance, until all the lots are scheduled on the batch processing machine.

5.4 Model, Assumptions and Parameters

The proposed LAB method for ET performance is computationally experimented for the problem of scheduling a batch processing machine (e.g., oxidation furnaces in semiconductor fabs) with dynamic arrivals of the lots. The basic model is schematically presented in Figure 5.6.

Figure 5.6: Schematic diagram of the experimental model

The following assumptions and constraints are considered in the computational study:

1. The lots arrive dynamically at the machine.

5. Single Objective Look Ahead Batch Processing

2. Batch capacity, *i.e.*, the maximum batch size is constant.
3. Prior knowledge of the arrival time and the due date of the upcoming lots at batch processing machine is available.
4. The lots are independent, *i.e.*, the lots are processed without any precedent relationship.
5. No machine breakdowns or preventive maintenance is considered, *i.e.*, machine is always available for processing.
6. Setup times are independent of the sequence, and are included in the processing times. The processing times in the furnace batch operations are assumed to be constant.
7. No interruption to lots in processing, implying no pre-emption.
8. There are no other limiting resources such as material or labor.

Lots arrive dynamically at the furnace and wait in a queue. If they are unable to be processed immediately, the arrival epochs of incoming lots are predicted. For simplicity, the number of wafers in each lot is assumed to be the same and the batch capacity is assumed to be six lots, with unlimited buffer capacity.

Table 5.1 summarizes the factors considered in the study of computational experiments. The first factor in Table 5.1 is the traffic intensity (utilization). The value selected for this factor are 0.70 and 0.80. The second factor, batch processing time, is considered only at one constant level of 60 min, since in the furnace model of batching, the processing time of the batch is taken as constant.

The third factor is the inter arrival time distribution, which is selected as exponential distribution. The actual parameters of the distribution are determined by specifying the traffic intensity. Based on the known relation from the queuing results,

5. Single Objective Look Ahead Batch Processing

Table 5.1: Factors for computational experiments

No.	Factor	Levels	Description
1.	Traffic Intensity (ρ)	Two	0.70, 0.80
2.	Processing Time ($T = 1/\mu$)	Constant	60 min (Furnace model of batching)
3.	Interarrival Distribution	Exponential	Mean ($\lambda = \rho \cdot c\mu$)
4.	Due Date Tightness ($d_i = r_i + k \cdot T$)	Tight Loose	$k \in \text{U}[-2,6]$ $k \in \text{U}[1,3]$
5.	Batch Scheduling Method	DBH NACH LAB	Arrival time of future lots Arrival time of future lots Arrival time and due date of future lots

the traffic intensity for a single batch processor can be related to the arrival rate as:

$$\rho = \frac{\lambda}{c \cdot \mu} \quad (5.4)$$

where ρ is the traffic intensity, λ is the mean arrival rate, μ is the mean service rate and c is the batch capacity. Hence, keeping the service rate at a constant value for all the experiments, the arrival rate is set to depend on the traffic intensity. The inter-arrival time is utilized to find random arrival times of the lots according to Poisson distribution function. The exponential case represents high variation, *i.e.*, the coefficient of variation is equal to 1.0.

The fourth factor deals with the tightness of the due dates in the system. It rep-

5. Single Objective Look Ahead Batch Processing

resents the allowance between the arrival time and the due date of the job. This factor is considered at two levels: tight and loose. The final factor is the batch scheduling methods employed. These are DBH, NACH and the proposed LAB method.

Figure 5.7: Steady state simulation results of cumulative average cycle time in dynamic batch scheduling ($\rho = 0.7$).

[Out of results for total 4000 lots, initial 1000 lots are considered to be in the initial warm-up period. Steady state data collection is done from 1001 to 3000 lots.]

Finally, all experimental results are collected after achieving the steady state performance, *i.e.*, after the initial warm-up period of the simulation runs. For each combination of factors (*i.e.*, each data point), 10 replications are taken to reduce the effect of variance in each replication. The steady state performance of cumulative average cycle time is shown in Figure 5.7. Steady state of cumulative average cycle time is achieved after about 800 lots. Therefore, each replication contains the same number of observations (2000) after an initial warm-up period of 1000 observations. A total of

4000 jobs are continuously released in the single machine system and the data collection for the steady state performance measure is done from lot no. 1001 up to lot no. 3000 for 2000 lots. The simulation program is written and implemented in MatLab6 environment and run on an Intel 2000 Pentium (4), 1.8 GHz PC.

5.5 Results of Single Objective Look Ahead Batch Processing

The ET performance is measured by the minimization of the average absolute sum of earliness and tardiness of the lots, $\overline{(|E| + |T|)}$, and the minimization of their average square sum, $\overline{(E^2 + T^2)}$. These measures are respectively represented in Equations 2.11 and 2.13 in Chapter 2. The results of the proposed LAB method are compared against DBH and NACH. Often the maximum batch size of such operations is greater than the size of the arriving lots, and therefore these decision-making situations become very crucial for the improved performance of these batching operations.

Tables 5.2 and 5.3 present the steady state results of average absolute sum and average square sum of earliness and tardiness for DBH, NACH, and LAB methods. Each column represents an independent setting of experiments. The first three rows respectively represent the values of the ET performance measure for DBH, NACH and the proposed LAB method. These values are the average of ten replications. The next two rows represent the percentage reduction in the ET performance measure respectively for LAB over DBH and LAB over NACH.

From the results in Tables 5.2 and 5.3, it is clearly observed that LAB improves the performance of average absolute sum of earliness and tardiness by more than 35% in comparison to DBH and NACH method. For example, with loose due date setting and 0.70 traffic intensity (ρ), the value of the average sum of absolute earliness and tardiness is 38.83 min for the proposed LAB method, which is 40.60% reduction from

5. Single Objective Look Ahead Batch Processing

Table 5.2: Steady state results of average sum of absolute earliness and tardiness

[The results of LAB method are compared against DBH and NACH for all four experimental settings. LAB method improves performance by more than 35%.]

Method	$\overline{(E + T)}$ (min)			
	Tight due date		Loose due date	
	$\rho = 0.7$	$\rho = 0.8$	$\rho = 0.7$	$\rho = 0.8$
1. DBH	122.41	122.50	65.37	65.44
2. NACH	122.48	122.71	65.64	65.60
3. LAB	79.34	74.43	38.83	41.03
% Reduction (LAB over DBH)	35.19	39.24	40.60	37.30
% Reduction (LAB over NACH)	35.22	39.34	40.84	37.45

corresponding value for DBH method (65.37 min) and 40.84% reduction from corresponding value for NACH method (65.64 min). Similarly, for the average square sum of earliness and tardiness, LAB improves the performance by more than 45% in comparison to DBH and NACH methods. Figure 5.8 (a) and (b) presents the column graphs for the % reduction in (a) average sum of absolute earliness and tardiness $\overline{(|E| + |T|)}$ and (b) average square sum of earliness and tardiness $\overline{(E^2 + T^2)}$ on using LAB method over DBH and NACH methods.

Though, for the ET performance, the comparison of the proposed LAB method with that of DBH and NACH is unfair, as DBH and NACH are intended for the cycle time objectives. However, these results demonstrate the benefit of using knowledge about the arrival times and due-dates of future coming lots in significantly reducing the ET performance measures in scheduling semiconductor batch processes, which

5. Single Objective Look Ahead Batch Processing

Table 5.3: Steady state results of average square sum of earliness and tardiness

[The results of LAB method are compared against DBH and NACH for all four experimental settings. LAB method improves performance by more than 45%.]

Method	$\overline{(E^2 + T^2)}$ (min ²)			
	Tight due date		Loose due date	
	$\rho = 0.7$	$\rho = 0.8$	$\rho = 0.7$	$\rho = 0.8$
1. DBH	20417.13	20449.94	6101.48	6124.29
2. NACH	20525.32	20524.47	6198.90	6159.70
3. LAB	10731.68	9551.94	2423.27	2730.51
% Reduction (LAB over DBH)	47.44	53.29	60.28	55.42
% Reduction (LAB over NACH)	47.71	53.46	60.91	55.67

generally constitute a bottleneck for the semiconductor manufacturing operations. ET performance is important to both wafer fabrication and back-end factories, where both early and tardy completion of lots is strongly discouraged.

In this chapter, the ET performance is measured in terms of a joint measure of earliness and tardiness such as the minimization of the average absolute sum of earliness and tardiness of the lots $\overline{(|E| + |T|)}$ and the minimization of their average square sum $\overline{(E^2 + T^2)}$. However, if the ET performance is measured in terms of two individual objective functions such as simultaneous minimization of average earliness and of average tardiness, then it becomes a multi-objective dynamic batch scheduling problem with availability of look ahead information of the incoming lots. This is further addressed in the next chapter as “Pareto control in look ahead batch processing”. Thus, the proposal of LAB method for the ET performance in this chapter becomes a

5. Single Objective Look Ahead Batch Processing

(a) average sum of absolute earliness and tardiness $\overline{(|E| + |T|)}$

(b) average square sum of earliness and tardiness $\overline{(E^2 + T^2)}$

Figure 5.8: Column graphs for % reduction in (a) average sum of absolute earliness and tardiness $\overline{(|E| + |T|)}$ and (b) average square sum of earliness and tardiness $\overline{(E^2 + T^2)}$

prerequisite to the application of LAB method for Pareto control in Chapter 6.

5.6 Conclusion

In this chapter, a novel LAB method is proposed, which utilizes the arrival and due date information of the upcoming lots and achieves the optimal batch decision for the ET performance in dynamic batch scheduling. ET performance is measured in terms of a joint measure of earliness and tardiness such as the minimization of the average absolute sum of earliness and tardiness of the lots $(|E| + |T|)$ and the minimization of their average square sum $(E^2 + T^2)$. By varying the due date pattern and the traffic intensity, simulation experiments are conducted to show the effect of LAB over DBH and NACH for ET performance measures. The steady state results of dynamic scheduling of batch processing machine indicate the well-improved performance of the LAB method under a wide variety of service rate and utilization level conditions. For average absolute sum of earliness and tardiness, LAB method gives more than 35% improvement over DBH and NACH, while more than 45% enhancement on average square sum of earliness and tardiness.

In this chapter, while using the proposed LAB method to minimize the joint earliness-tardiness measures, in effect *a priori* weights (*i.e.*, 50-50 weight method) are used to evaluate the earliness and tardiness objectives, which are conflicting in themselves when seen independently. In the next chapter, earliness and tardiness are taken as two conflicting individual objectives and the weights are assigned *a posteriori* according to the preference of the decision maker. Thus, using the adopted Pareto control, the proposed LAB method for ET performance is further exploited to develop Pareto control curves between earliness and tardiness related objectives in Chapter 6.

Chapter 6

Pareto Control in Look Ahead Batch Processing

6.1 Introduction

In Chapter 5, a new LAB method was proposed to exploit the future arrival and due date information for minimizing ET performance measures in dynamic scheduling of a batch processing machine. The considered ET performance measures are just the convex combinations of earliness and tardiness with *a priori* chosen weights, in particular with equal weights. In this chapter, the weights of earliness and tardiness are not decided *a priori*; rather a control curve is achieved by taking the full range of weights and leaving the decision of choosing an operating range on the decision maker itself. This has been achieved by using CP method to get a Pareto control curve on the two sets of conflicting objectives: (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs.

This chapter presents a combined concept of LAB and Pareto control with respect to the problem of simultaneously minimizing the conflicting objectives of earliness and tardiness related measures in dynamic scheduling of a batch processing machine. Earliness and tardiness are taken as two individual objectives, which are by definition perfectly negatively correlated for an individual job. Therefore, earliness and tardiness

related objectives for a system as a whole are highly negatively correlated and so conflicting in nature. In production scheduling, the requirements of earliness and tardiness related objectives vary with time according to the dynamic market forces. This creates a need to control the ET performance schedule of a batch processing machine in a dynamic environment. The Pareto control in look ahead batch processing has been achieved by using CP method for earliness and tardiness objectives.

6.2 Earliness-Tardiness (ET) Performance

In general JIT manufacturing environment, instead of a strict due date, a due date window is given as a criterion to meet the on-time delivery of the jobs (Baker and Scudder, 1990). A due date window is defined by a lower and an upper limit of the due date as $[d_{1i}, d_{2i}]$. The job (J_i) is said to be early, if it gets completed before d_{1i} ; late, if it gets completed after d_{2i} ; and on-time, otherwise. Figure 6.1 presents the concept of due date window clearly.

Figure 6.1: Due date window concept

Therefore, the earliness (E_i) and tardiness (T_i) of job J_i and their respective indicators are redefined as follows:

$$E_i = \max(0, d_{1i} - C_i) \quad (6.1)$$

$$T_i = \max(0, C_i - d_{2i}) \quad (6.2)$$

$$U_{Ei} = \begin{cases} 1, & \text{if } C_i < d_{1i} \\ 0, & \text{else} \end{cases} \quad (6.3)$$

$$U_{Ti} = \begin{cases} 1, & \text{if } C_i > d_{2i} \\ 0, & \text{else} \end{cases} \quad (6.4)$$

In the previous chapter, the strict due date was considered for the simplicity in validating the results of the proposed LAB method for the ET performance. The ET performance was measured by the minimization of the average absolute sum of earliness and tardiness of the lots, $\overline{(|E| + |T|)}$, and the minimization of their average square sum, $\overline{(E^2 + T^2)}$.

In this chapter, the due date window is considered for a more realistic picture of JIT manufacturing scenario. The ET performance is measured in terms of two sets of earliness and tardiness related conflicting objectives:

- average earliness *vs.* average tardiness

$$\min \left\{ \frac{1}{n} \sum_{\forall i} (E_i) \text{ and } \frac{1}{n} \sum_{\forall i} (T_i) \right\} \quad (6.5)$$

- number of early jobs *vs.* number of tardy jobs

$$\min \left\{ \sum_{\forall i} (U_{Ei}) \text{ and } \sum_{\forall i} (U_{Ti}) \right\} \quad (6.6)$$

Due to the conflicting nature of these objectives, the Pareto control curve needs to be developed and investigated for controlling the ET performance in look ahead batch processing.

6.3 Pareto Control using LAB

For Pareto control between earliness and tardiness related objectives, Steps 1, 2 and 4 of the proposed LAB algorithm in Section 5.3 remain the same, while the Step 3 to select the best LAB scenario is replaced by the following procedure.

6. Pareto Control in Look Ahead Batch Processing

As shown in Section 5.3, at each decision instance in simulation clock, various LAB scenarios can be defined within one batch processing time as also presented in Figure 6.2.

Figure 6.2: Look Ahead Batching (LAB) scenarios

Considering that j^{th} LAB scenario starts its processing at time t_j , a measure of batch earliness (BE_j) and tardiness (BT_j) can be defined for each LAB scenario as follows:

$$BE_j = \sum_{i=1}^{q+j} E_i, \quad 0 \leq j \leq j_{max} \quad (6.7)$$

$$BT_j = \sum_{i=1}^{q+j} T_i, \quad 0 \leq j \leq j_{max} \quad (6.8)$$

where j is the indicator of LAB scenario, i represents the jobs in each LAB scenario and q is the cumulative number of jobs in queue at simulation clock (t_0). In a batch, there may be some jobs finishing early, some finishing late and some on-time. Therefore,

6. Pareto Control in Look Ahead Batch Processing

batch earliness (BE_j) is defined as the sum of the earliness of all the early finishing jobs in the batch of j^{th} LAB scenario. Similarly, the batch tardiness (BT_j) is defined as the sum of the tardiness of all the late finishing jobs in the batch of j^{th} LAB scenario.

Thus, each LAB scenario has two measures, batch earliness and batch tardiness, both of which need to be minimized for JIT scheduling, *i.e.*, the selection of the best LAB scenario constitutes a bi-objective optimization problem at each decision instance in simulation clock.

$$\min_{0 \leq j \leq j_{max}} \{BE_j \text{ and } BT_j\} \quad (6.9)$$

Now, for this bi-objective optimization problem at each decision instance, the Pareto optimal solution, *i.e.*, the Pareto optimal LAB scenario is determined by using compromise programming (CP) method. CP method is a convex combination approach to solve the multi-objective optimization problems. It has been discussed earlier in detail in Section 2.3.2.

Using CP method with weight w and exponent r , a Pareto objective function, $F_r(j)$, can be formulated for each LAB scenario as follows:

$$F_r(j) = \left[\left(w \cdot \frac{|BE_j - \min(BE_j)|}{\min(BE_j)} \right)^r + \left((1 - w) \cdot \frac{|BT_j - \min(BT_j)|}{\min(BT_j)} \right)^r \right]^{1/r} \quad (6.10)$$

where $\min(BE_j)$ represents the minimum value of batch earliness, BE_j , for all possible LAB scenarios within one batch processing time ($0 \leq j \leq j_{max}$) and similarly $\min(BT_j)$ represents the minimum value of batch tardiness, BT_j , $0 \leq j \leq j_{max}$.

Then, among all constructed LAB scenarios, the best LAB scenario is selected for loading on the batch processing machine. The scenario, which has the minimum value of Pareto objective function $F_r(j)$, is selected as the best for loading.

$$k = \arg \min_{0 \leq j \leq j_{max}} (F_r(j)) \quad (6.11)$$

where k implies the best LAB scenario to be loaded for processing on the batching machine.

Thus, by varying the weight w and exponent r in CP method, different control curves can be achieved for the conflicting earliness and tardiness objectives. Pareto control in look ahead batch processing is computationally experimented in the next section for the dynamic scheduling of a batch processing machine.

6.4 Computational Experiments for Pareto Control using LAB

Computational experiments for Pareto control in look ahead batch processing have been conducted for the problem of scheduling a batch processing machine with dynamic arrivals of the lots. The model, assumptions and constraints of the experiments are kept similar to Chapter 5, as earlier described in Section 5.4. The basic model was schematically presented in Figure 5.6. The parameters and setting are changed to include the due date window factor. Table 6.1 summarizes the factors considered for the computational experiments.

The first factor is the traffic intensity (ρ), i.e., the machine utilization which is important because it affects the queue length. The traffic intensity is considered at three different levels, 0.50, 0.60 and 0.70, as considered in Chapter 4. Some experiments are tried at ρ values lower than 0.5 and also at ρ values higher than 0.7 and the following observations are made. If the traffic intensity is much lower than 0.50, there may be longer inter-arrival times and so very few future arrivals will occur within the look ahead window. Thus, at every decision instance in simulation clock, there are not many options of batch formation and so there is no variation in performance of LAB strategy with respect to variation in weight. The 0.70 value represents moderately heavy traffic, but not so heavy that the decisions made by different strategies are

6. Pareto Control in Look Ahead Batch Processing

Table 6.1: Factors for computational experiments

No.	Factor	Levels	Description
1.	Traffic Intensity (ρ)	Three	0.50, 0.60, 0.70
2.	Processing Time ($T = 1/\mu$)	Constant	60 min (Furnace model of batching)
3.	Interarrival Distribution	Exponential	Mean ($\lambda = \rho \cdot c\mu$)
4.	Due Date Tightness ($d_i = r_i + k \cdot T$)	DD1: Tight DD2: Medium DD3: Loose	$k \in \text{U}[-2,6]$ $k \in \text{U}[0,4]$ $k \in \text{U}[1,3]$
5.	Due Date Window ($[d_{1i}, d_{2i}]$)	DDW0: Strict DDW1: Small DDW2: Medium DDW3: Large	$d_{1i} = d_{2i} = d_i$ $d_{1i} = d_i - 0.2 \cdot T$; $d_{2i} = d_i + 0.1 \cdot T$ $d_{1i} = d_i - 0.4 \cdot T$; $d_{2i} = d_i + 0.2 \cdot T$ $d_{1i} = d_i - 0.6 \cdot T$; $d_{2i} = d_i + 0.3 \cdot T$
6.	Batch Scheduling Strategy	MBS1-MBS6 DBH NACH LAB Based	No look-ahead information Arrival time of future lots Arrival time of future lots Arrival time and due date of future lots

always the same, i.e., a full batch loading. For ρ values greater than 0.70, the system with batch processing machine becomes so heavily loaded that at each decision instance a full batch is loaded on the machine for processing. This makes the evaluation of the effectiveness of LAB strategy difficult or almost impossible. Therefore, the ρ value is considered within the region of interest at 0.50, 0.60 and 0.70 levels.

The single batch processing machine is considered under the furnace model of

6. Pareto Control in Look Ahead Batch Processing

batching, where the processing time of the batch is taken as constant. Therefore, the second factor, batch processing time, is considered only at one constant level of 60 minutes, as all other parameters are directly proportionate to it. The third factor is the inter arrival time distribution, which is selected as exponential distribution. The inter arrival time is utilized to find random arrival times of the lots according to Poisson distribution function, as done in the previous chapter in Section 5.4.

The next two factors deal with the nature of the due dates in the system. The due date tightness factor represents the allowance between the job arrival time and its due date. This factor is considered at three levels: tight, medium and loose. It is designed in such a way that the mean of allowance is equal at all levels, while the variance is high at tight level and low at loose level. The due date window factor represents the period (d_{1i}, d_{2i}) in which the job completion time reflects the on-time delivery of the job J_i . This factor is considered at four levels: strict, small, medium and large. Strict due date window implies $d_{1i} = d_{2i} = d_i$. Other three levels are taken to represent the JIT scenario.

The final factor is the batch scheduling methods employed. These are MBS policies from MBS1 to MBS6, DBH, NACH and the proposed LAB method.

The default setting of these factors is assumed as traffic intensity (ρ) at 0.70, due date tightness at DD1 and due date window at DDW1. For Pareto control between the conflicting objectives of earliness and tardiness, the proposed LAB strategy employs CP method, which uses various values of individual weights and the exponent. In general, in CP method, the weights are taken as 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 0.95 and the exponent is regarded at 1, 2 and 3.

The flow chart of the simulation experiment is presented in Figure 6.3. All experimental results are collected after achieving the steady state performance, *i.e.*, after the initial warm-up period of the simulation runs. For each combination of factors

6. Pareto Control in Look Ahead Batch Processing

Figure 6.3: Flow chart of the simulation experiment for the dynamic scheduling of a batch processing machine

6. Pareto Control in Look Ahead Batch Processing

(a) $\rho = 0.5$

(b) $\rho = 0.7$

Figure 6.4: Steady state simulation results of cumulative average cycle time in dynamic batch scheduling at two levels of traffic intensities: (a) $\rho = 0.5$ and (b) $\rho = 0.7$.

[Out of results for total 4000 lots, initial 1000 lots are considered to be in the initial warm-up period. Steady state data collection is done from 1001 to 3000 lots.]

(*i.e.*, each data point), 10 replications are taken to reduce the effect of variance in each replication. The steady state performance of cumulative average cycle time is shown in Figure 6.4 for two different arrival rates, *i.e.*, traffic intensities ($\rho = 0.5$ and 0.7). The steady state curve for $\rho = 0.6$ is also similar to the graphs in Figure 6.4 and so it is not presented here. Steady state of cumulative average cycle time is achieved after about 800 lots. Therefore, each replication contains the same number of observations (2000) after an initial warm-up period of 1000 observations. A total of 4000 jobs are released at discrete intervals in the single machine system and the data collection for the steady state performance measure is done from lot no. 1001 up to lot no. 3000 for 2000 lots. The simulation program is written and implemented in MatLab6 environment and run on an Intel 2000 Pentium (4), 1.8 GHz PC.

The Pareto control in LAB processing for the ET performance is characterized for two sets of earliness and tardiness related conflicting objectives: (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs. The results of these two sets of conflicting objectives are plotted on 2-dimensional graphs and discussed in the following.

6.5 Results of Pareto Control in Look Ahead Batch Processing

The performance of LAB method is compared against several other batch scheduling methods: MBS (requiring no look-ahead information), DBH and NACH (requiring future arrival times). Figure 6.5 shows the comparison of the results for two sets of bi-objective cases: (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs. From Figure 6.5(a), it is observed that MBS1 - MBS6 methods give the same level of average earliness and average tardiness. The values of average earliness for MBS1 - MBS6 are 62.39, 62.53, 62.79, 63.03, 62.37 and 61.19 mins. While the corresponding values of average tardiness for MBS1 - MBS6 are 44.10,

6. Pareto Control in Look Ahead Batch Processing

43.96, 44.18, 44.56, 45.68 and 47.94 mins with slight increasing trend.

DBH and NACH methods are meant to reduce the cycle time but at the cost of increasing average earliness. Therefore, the average earliness for DBH and NACH methods gets increased up to 68.77 and 69.89 mins respectively, while the average tardiness, 44.69 and 43.61 mins, remains in the same range.

In contrast to these, instead of just one point on the two dimensional graph, LAB method gives a control line trading off between average earliness and average tardiness. LAB method uses CP formulation, which has double weighting scheme in it, the weight and the exponent. For exponent values 1, 2 and 3, CP curves are represented by CP1, CP2 and CP3 in Figure 6.5. Changing the weight value gives the control line on a two dimensional graph. The weight is varied at 11 different values at 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 0.95. Thus, for each CP1, CP2 and CP3, a total of 11 points are shown in the graph, which constitute a controllable curve. From various experimental results, it is observed that the variation of exponent in CP formulation does not significantly affect the performance of LAB method.

In Equation 6.10, w reflects the weight of earliness objective and so in LAB method, 0.05 weight gives a point near to minimizing only tardiness. As this weight increases towards 0.95, the tardiness increases and the earliness decreases. Thus, changing weights in a linear fashion gives a control line between the objectives of earliness and tardiness. It is observed that the control line is almost a straight line with a negative slope which corroborates the mathematical fact that the earliness and tardiness related objectives are highly negatively correlated.

Similarly, Figure 6.5 (b) shows the schedule control with respect to the number of early jobs and the number of tardy jobs. These are also found to be highly negatively correlated, giving a linear control line with negative slope. The sum of the number of early jobs and the number of tardy jobs should always be less than or equal to the

6. Pareto Control in Look Ahead Batch Processing

(a) Average earliness *vs.* Average tardiness

(b) Number of early jobs *vs.* Number of tardy jobs

Figure 6.5: Performance of LAB method over MBS, DBH and NACH methods for schedule control on two objectives (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs. (Experimental settings : $\rho = 0.7$, DD1, DDW1)

6. Pareto Control in Look Ahead Batch Processing

number of total jobs considered in steady state data collection, *i.e.*, total 2000 jobs from job no. 1001 up to job no. 3000. For the strict due date case (DDW0), this sum will be strictly equal to 2000 and thus the curve between the number of early jobs and the number of tardy jobs will be a straight line with slope of negative unity.

Similar to observations in Figure 6.5 (a), MBS1 - MBS6, DBH and NACH methods give the single points in the close region in Figure 6.5 (b) also. In contrast to these, LAB method gives a control line trading off between the number of early jobs and the number of tardy jobs. Here also, the variation of exponent in CP formulation does not significantly affect the performance of LAB method. However, the variation of weight in a linear fashion gives a control line trading off between the number of early jobs and the number of tardy jobs. This linear behavior of the curve between the number of early jobs and the number of tardy jobs gives the benefit of controlling the performance of the schedule based on the user preference in JIT scenario.

Further, the performance of LAB method is compared against MBS, DBH and NACH for two ET performance measures. The two ET performance measures are $\overline{(|E| + |T|)}$ and $\overline{(E^2 + T^2)}$, as represented in Equations 2.11 and 2.13 in Chapter 2. The LAB method gives a whole series of points according to the weight in CP formulation in Equation 6.10. These results are presented in pictorial form in Figure 6.6 in order to show the pattern of LAB method with respect to the weight. The results of average sum of absolute earliness and tardiness, $\overline{(|E| + |T|)}$, are presented in Figure 6.6 (a), while the results of the average square sum of earliness and tardiness $\overline{(E^2 + T^2)}$ are shown in Figure 6.6 (b).

Since MBS1 - MBS6, DBH and NACH methods do not include the consideration of weight, in Figures 6.6 (a) and (b), these points are presented on the Y-axis. While LAB method gives a controllable curve with respect to the weight. These curves are represented by CP1, CP2 and CP3 according to the exponent value in CP formulation in Equation 6.10. All the points on the CP curves give a value less than the value

6. Pareto Control in Look Ahead Batch Processing

(a) Average sum of absolute earliness and tardiness $\overline{(|E| + |T|)}$

(b) Average square sum of earliness and tardiness $\overline{(E^2 + T^2)}$

Figure 6.6: Performance of LAB method over MBS, DBH and NACH methods for two ET performance measures (a) $\overline{(|E| + |T|)}$ and (b) $\overline{(E^2 + T^2)}$. (Experimental settings : $\rho = 0.7$, DD1, DDW1)

of MBS1 - MBS6, DBH and NACH methods. From these results, it is quite evident that for ET performance measures LAB method gives a significant improvement over the other batching methods. In addition, by varying the weight in CP formulation, LAB method gives the additional benefit of Pareto control between the earliness and tardiness related conflicting objectives.

In the next section, the default settings ($\rho = 0.70$, DD1 and DDW1) of the experimental parameters are varied one by one and the variability effects on the schedule control are studied.

6.6 Effects of Parameter Variation on Pareto Control using LAB

In this section, the issue of changing various parameter settings (traffic intensity, due date tightness and due date window) and observing its effect on achieving Pareto control is addressed in bi-objective dynamic scheduling of a batch processing machine with look ahead information of the upcoming jobs.

6.6.1 Effect of traffic intensity

In this set of experiments, the traffic intensity (ρ) is varied at 0.50, 0.60 and 0.70 levels, while keeping the due date tightness at DD1 and due date window at DDW1 levels. For ρ values greater than 0.70, the system with batch processing machine becomes heavily loaded. While for a serial processing machine, the system becomes heavily loaded for ρ values greater than 0.85. This is because a batch processing machine involves the starting and finishing the processing of several jobs together, which results in capacity loss. On the other hand, for ρ values less than 0.50, at every decision instance in simulation clock, there are not many options of batch formation and so there is no variation in performance of LAB method with respect to variation in weight. The

6. Pareto Control in Look Ahead Batch Processing

(a) Average earliness *vs.* Average tardiness

(b) Number of early jobs *vs.* Number of tardy jobs

Figure 6.7: Effect of varying traffic intensity on schedule control between (a) Average earliness *vs.* Average tardiness and (b) Number of early jobs *vs.* Number of tardy jobs (Experimental settings: DD1, DDW1)

results of LAB method for (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs are presented in Figures 6.7 (a) and (b) respectively.

In Figures 6.7 (a) and (b), it is observed that the points corresponding to 0.05 weight are very close to each other for different curves of ρ variation. While the points corresponding to 0.95 weight get more and more distanced with the increase in ρ value. So, with the increase in the traffic intensity (ρ), the breadth of the schedule control curves for (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs increases. This observation corroborates with the fact that with increase in arrival rate there will be an increase in the number of possible LAB scenarios at each decision instance in simulation clock and so there will be a wider range of options to select.

6.6.2 Effect of due date tightness

The due date tightness is varied at three levels: DD1 (tight), DD2 (medium) and DD3 (loose). In all the cases, the due date allowance factor (k) is varied uniformly around a common mean value of 2. k is uniformly generated from $U[-2, 6]$, $U[0, 4]$ and $U[1, 3]$ for DD1, DD2 and DD3 respectively. The difference of interval length sets the due-date tightness from the arrival time. The results of LAB method with respect to variation in due date tightness are presented in Figures 6.8 (a) and (b) respectively for (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs. The observations of these results state that as the due date turns from loose to tight, the schedule control curve shifts towards the origin. Along with it, the width of the curve for average earliness *vs.* average tardiness gets reduced.

6. Pareto Control in Look Ahead Batch Processing

(a) Average earliness *vs.* Average tardiness

(b) Number of early jobs *vs.* Number of tardy jobs

Figure 6.8: Effect of varying due date tightness on schedule control between (a) Average earliness *vs.* Average tardiness and (b) Number of early jobs *vs.* Number of tardy jobs (Experimental settings: $\rho = 0.7$, DDW1)

6. Pareto Control in Look Ahead Batch Processing

(a) Average earliness *vs.* Average tardiness

(b) Number of early jobs *vs.* Number of tardy jobs

Figure 6.9: Effect of varying due date window on schedule control between (a) Average earliness *vs.* Average tardiness and (b) Number of early jobs *vs.* Number of tardy jobs (Experimental settings: $\rho = 0.7$, DD1)

6.6.3 Effect of due date window

Due date window defines a job to be early, tardy or on-time. Due date window is varied at four levels: DDW0 (strict), DDW1 (small), DDW2 (medium) and DDW3 (large). The corresponding results of schedule control between the two pairs of conflicting objectives related to earliness and tardiness are shown in Figures 6.9 (a) and (b).

From these results, it is observed that as the due date window becomes broader, the schedule control curves shift towards the origin in both graphs. This corroborates the fact that the increase in due date window signifies more number of jobs on-time and thus reduction in both the number of tardy jobs and the number of early jobs. This reduces the average earliness and the average tardiness as well and thus makes the curve in both the graphs to shift towards the origin.

For strict due date case (DDW0), the sum of the number of early jobs and tardy jobs will always be equal to 2000 and thus it represents a straight line with negative unity (-1) slope in Figure 6.9 (b). As the width of the due date window increases from zero, this straight line gets converted in to a convex line and starts shifting towards the origin. In this figure, the slope of DDW1, DDW2 and DDW3 lines are not equal to -1. This is due to the fact that the due date windows are specifically designed to tilt towards earliness, as shown in Table 6.1.

6.7 Conclusion

In this chapter, the LAB method for ET performance has been coupled with CP method in order to characterize the Pareto control curves for the conflicting objectives related to earliness and tardiness. This enables the decision maker to have the flexibility of controlling the ET performance in dynamic batch scheduling. The two sets of earliness and tardiness related conflicting objectives are (a) average earliness *vs.* average

6. Pareto Control in Look Ahead Batch Processing

tardiness and (b) number of early jobs *vs.* number of tardy jobs.

The concept of LAB is exploited to develop Pareto control between earliness and tardiness related objectives, which are highly negatively correlated. To achieve Pareto control in look ahead batch processing, CP method is used with *a posteriori* assigned weights according to the preference of the decision maker. In using CP method, it is found that herein the exponent variation of CP formulation is ineffective while the variation of weight enables one to achieve control over the full breadth of the curve.

Further, the experimental study showed that the trade-off curves between earliness and tardiness related objectives are sensitive to the variation in traffic intensity, due date tightness and due date window. However, the developed methodology of using LAB method with CP formulation is found capable of achieving Pareto controllable curves for the earliness and tardiness related conflicting objectives in all these variations.

Developing Pareto control between earliness and tardiness related conflicting objectives is very helpful for the production scheduling in JIT environment, where there is a need to control the schedule of a batch processing machine in a dynamic environment according to the varying requirements of earliness and tardiness objectives.

Chapter 7

Industrial Application: A Case Study on Photolithography Cluster

In this chapter, the research findings are summarized and discussed from the angle of relevance in the real-life industrial applications. In addition, a case study with the factory data from a wafer fab plant in Singapore is being presented. The case study is built on the model of Pareto control in serial processing as presented in Chapter 3. In the case study, two sets of conflicting objectives are considered: (1) simultaneously minimizing mean cycle time and maximum tardiness and (2) simultaneously minimizing mean cycle time and cycle time variance. Further, a 3-D case of Pareto control is briefly discussed considering all three above mentioned objectives simultaneously.

7.1 Relevance of Research Findings in Industrial Application

The various studies and simulation experimental investigations presented in this thesis are highly relevant from the point of view of semiconductor industries in Singapore such as National Semiconductors, ST Microelectronics, Chartered Semiconductor Manufacturing, Infineon Technologies AG, Systems on Silicon Manufacturing Co., *etc.* These companies have shown their interest in this research and may provide the testing

7. Industrial Application: A Case Study on Photolithography Cluster

ground for implementing these novel ideas in their factories. Primarily, from the point of view of industrial application, it is important to develop a methodology for operating the shop in a user-controlled trade-off between the needs of conflicting scheduling objectives.

In a dynamic scheduling environment, the trade-off between conflicting objectives hinges on selecting an appropriate job for scheduling on the machine in such a way that it will give the overall schedule according to the defined emphasis on the particular objectives. Since the emphasis on different scheduling objectives keeps varying according to the dynamic market forces, it is of high importance to develop a Pareto control method in multi-objective dynamic scheduling of manufacturing processes. Therefore, the research emphasis has been laid on the ability to control the scheduling performance along the trade-off curves between the conflicting objectives.

In summary, the concept of Pareto control has been proposed and applied for the conflicting pairs of scheduling objectives in three environments: (i) serial processing, (ii) batch processing and (iii) look ahead batch processing, which have been respectively presented in Chapters 3, 4 and 6. The methodology of achieving Pareto control curves for two conflicting objectives includes first finding the selection criterion for optimizing each individual objective in dynamic scheduling and then forming a combined selection criterion using convex combination methods such as CP method.

The experimental studies showed that by varying the relative weights of the selection criteria at each selection instance in simulation clock, a Pareto control curve can be achieved for two conflicting objectives. This characterization of Pareto control curves for the conflicting objectives can be exploited to enhance the ability to choose and operate the shop in a user controlled way. By appropriate selection of parameters, the conflicting scheduling objectives can be controlled and maintained within a specified range for a particular operation. This Pareto control is highly relevant for the shop floor operations management.

7. Industrial Application: A Case Study on Photolithography Cluster

From the industrial engineering and management perspective of a manufacturing concern, the shop floor management has to simultaneously incorporate the conflicting concerns of the manufacturer and the customer in making the scheduling decisions. The main performance measures in any factory include the cycle time statistics, *i.e.*, cycle time mean and variance. In a manufacturing process, mean cycle time is closely related to the work-in-process inventory, both of which are major concerns for the manufacturer. Minimizing mean cycle time implies minimizing work-in-process inventory. On the other hand, minimizing cycle time variance helps in reliable predictions of the completion of jobs and thus in providing the same quality of service to the customers and therefore it reflects the customer's concern. In addition, minimizing tardiness related measure can be considered as a measure of customer satisfaction. In JIT manufacturing environment, finishing jobs too early induces high inventory holding costs and may mean loss in profit due to environmental degradation of product quality, while finishing too late implies contractual penalty and loss of customer good will. Thus, since the normal objectives of the production managers include inventory reduction and customer satisfaction improvements, the following pairs of conflicting scheduling objectives have been considered in this research: simultaneous minimization of (a) average cycle time and maximum tardiness, (b) average cycle time and cycle time variance and (c) earliness and tardiness related measures.

In the following section, a real-life case study has been presented in order to show the effectiveness of Pareto control curves for the shop floor management. This case study shows the application of Pareto control in serial processing with respect to the two sets of conflicting objectives: 1. mean cycle time and maximum tardiness and 2. mean cycle time and cycle time variance.

7.2 An Overview of Semiconductor Manufacturing

The overall manufacturing flow for a typical IC manufacturing process in a semiconductor industry can be divided into four main stages: wafer fabrication (also known as wafer fab), wafer probe/sort, assembly/packaging and final test. The flow between these processing stages is illustrated in Figure 7.1. In wafer fabrication, the integrated circuits are layered through successive operations onto a smooth wafer of a semi-conducting material, typically silicon. Each wafer may contain many individual dice. In wafer probe/sort, the individual dices are tested electronically using thin probes and the defective dices are marked. The wafer then goes to assembly where the wafer is sawed. Defective dices are discarded. The good dices are bonded to the lead frames for connection and capsulated for protecting them from the environment. After the assembly cycle, the IC chips are subjected to the final test. Among these stages, the wafer fabrication and wafer sort are usually known as the “front-end” and the IC assembly and testing are referred to as the “back-end”. The total manufacturing cycle time in the semiconductor industry, ranges from 8 to more than 30 weeks. The stage cycle time may range from 3-15 weeks in wafer fabrication; 2 days to 2 weeks in wafer probe; 3 days to 3 weeks in assembly and 2 days to 4 weeks in final test (Qi et al., 2002).

Figure 7.1: A simplified semiconductor manufacturing flow

7.3 Case Study Model, Data and Experimentation

This case study is based on the real factory data from one of the largest semiconductor wafer fabrication facilities in the world situated in Singapore. The wafer fabrication process dominates the economics of IC production and it is the most technologically complex and capital-intensive stage in semiconductor manufacturing. In the wafer fab, the wafers are processed in order to build up layers of patterns to produce the required circuitry. This involves a complex sequence of processing steps with a number of operations that require different kinds of equipment.

Wafer fabrication process basically involves the following process steps: photolithography, etching, deposition, chemical mechanical polishing, ion implantation, diffusion. Among these process steps, the photolithography process is considered the most crucial and bottleneck process. Due to the large number of photolithography steps needed in IC manufacturing, photolithography typically accounts for about 30% of the cost of manufacturing. Photolithography or optical lithography is basically a photographic process by which a light sensitive polymer, called a photoresist, is coated, exposed and developed to transfer a pattern from a photomask (also called reticle) to the surface of a wafer. The stepper machines are the pieces of equipment used for the photolithography process, which is the most expensive bottleneck resource in the wafer fabrication process.

The general sequence of processing steps for a typical photolithography process is as follows: substrate preparation, photoresist spin coat, prebake, exposure, development and postbake. This sequence is generally performed on several tools linked together into a contiguous unit called a lithographic cluster. A brief description of these processing steps is as follows:

1. Substrate Preparation: Substrate preparation is intended to improve the adhesion of the photoresist material to the substrate.

7. Industrial Application: A Case Study on Photolithography Cluster

2. Photoresist Coating: A thin, uniform coating of photoresist at a specific, well controlled thickness is accomplished by the process of spin coating. The photoresist, rendered into a liquid form by dissolving the solid components in a solvent, is poured onto the wafer, which is then spun on a turntable at a high speed producing the desired film.
3. Prebake: A prebake or a softbake process involves drying the photoresist after spin coat by removing the excess solvent. After cooling, the wafer is ready for its lithographic exposure.
4. Alignment and Exposure: A photomask, a square glass plate with a patterned emulsion of metal film on one side, is aligned with the wafer, so that the pattern can be transferred onto the wafer surface. The photoresist is then exposed through the pattern on the mask with a high intensity ultraviolet light.
5. Development: Once exposed, the photoresist must be developed by using some aqueous bases as developers.
6. Postbake: The postbake is used to harden the photoresist and improve adhesion of the photoresist to the wafer surface so that it will withstand the harsh environments of implantation or etching.

Figure 7.2: Flow of processing steps in a photolithography cluster

The main processing steps in photolithography are coat, expose and develop. In the photolithography cluster, there are two parallel coat tools, one expose tool and two parallel develop tools. The flow of these processing steps is presented in Figure 7.2. In this flow, the expose tool is a bottleneck as all the wafers has to necessarily pass

7. Industrial Application: A Case Study on Photolithography Cluster

through this single tool. This expose tool can be modeled as a single serial processing machine and the concept of Pareto control in serial processing can be applied in order to control the trade-off in multi-objective dynamic scheduling.

A sample of the factory data is presented in Table 7.1. In this table, second column represents the Lot ID, third column gives the step begin time, which is basically the lot arrival time at the photolithography cluster, fourth column shows the priority of the lot, fifth column gives the turn ratio of the lot, sixth column gives the lot quantity in terms of the number of wafers in the lot and finally seventh column represents the expose time per wafer in the lot. For the confidentiality reasons, the Lot IDs are changed as shown in the second column of Table 7.1.

Table 7.1: Sample presentation of the factory data

S.No.	Lot ID	Step Begin Time	Lot Priority	Turn Ratio	Current Lot Quantity	Expose Time (sec/wafer)
1	XYZ001	30/11/2005 15:52	4	6.37	1	78.5414
2	XYZ002	30/11/2005 20:26	4	0	25	72.2
3	XYZ003	30/11/2005 20:27	4	0	25	72.2
4	ABC001	30/11/2005 20:27	4	0	25	72.2
5	ABC002	30/11/2005 20:28	4	0	25	72.2
6	ABC003	30/11/2005 20:35	4	0	25	72.2
7	ABC004	30/11/2005 20:35	4	0	25	72.2
8	ABC005	30/11/2005 20:36	4	0	25	72.2
9	XYZ004	30/11/2005 20:36	4	0	25	72.2
10	ABC006	30/11/2005 20:36	4	0	25	72.2
11	XYZ005	30/11/2005 21:23	4	0	25	72.2
12	XYZ006	30/11/2005 21:24	4	0	25	72.2
13	XYZ007	30/11/2005 21:24	4	0	25	72.2
14	XYZ008	30/11/2005 21:24	4	0	25	72.2
15	XYZ009	30/11/2005 21:25	4	0	25	72.2
16	PQR001	30/11/2005 22:47	4	0	25	72.2
17	PQR002	01/12/2005 1:45	3	7.61	1	72.2
18	PQR003	01/12/2005 4:05	4	0	25	72.2
19	ABC007	01/12/2005 4:05	4	0	25	72.2
20	ABC008	01/12/2005 4:06	4	0	25	72.2
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
2508	PQR---	31/01/2006 19:29	2	6.99	25	80.1524

7. Industrial Application: A Case Study on Photolithography Cluster

The factory data is available for a period of two months, which involves the dynamic arrival of total 2508 lots. One requirement for applying the concept of Pareto control at the expose tool is the data for the arrival time at the expose tool, but this data is not available in the lot tracking system of the factory. In order to make the situation suitable to the model, the lot arrival time at the photolithography cluster is assumed to be the lot arrival time at the expose tool. Then, based on the advice of the industrial engineering experts at the factory, the expose processing time is multiplied with a random factor between 1 and 1.3 in order to balance the workload at the expose tool. Thus, the lot processing time at the expose tool is computed according to the following equation:

$$\text{lot processing time} = (1 + 0.3 * U(0, 1)) * \text{expose time per wafer} * \text{lot quantity} \quad (7.1)$$

where, $U(0,1)$ is the uniform random number between 0 and 1.

Another requirement for applying the concept of Pareto control is the operation due date for each lot at the expose tool, which is computed based on the following equation:

$$\text{operation due date} = \text{arrival time} + (\text{lot priority} * \text{lot processing time}) \quad (7.2)$$

The actual operating heuristic in factory is priority minimization followed by turn ratio maximization followed by first arrival, which is represented as factory heuristic. Other simple dispatching rules considered in this study are FCFS, SPT and EDD. The simulation experiments on the factory data are conducted for two cases: first with just 1-month data and second with 2-month data.

7.4 Case Study Results and Discussion

The concept of Pareto control in serial processing is applied on the expose tool for the two pairs of conflicting objectives:

7. Industrial Application: A Case Study on Photolithography Cluster

- (1) mean cycle time and maximum tardiness and
- (2) mean cycle time and cycle time variance.

Similar to the simulation experiments conducted in Chapter 3, the relative weights ($w_1, w_2; w_1 + w_2 = 1$) of the two criteria in the CP method are varied linearly from 0.05 to 0.95 at an interval of 0.05, computing total 19 points. The exponent r is varied at the levels of 0.33, 0.66, 1, 2, 3 and 10. The Pareto curves achieved for this case study are similar in nature to the Pareto curves presented in Section 3.6.

7.4.1 Mean cycle time and Maximum tardiness

The mean cycle time is minimized by the SPT rule and the maximum tardiness is minimized by the EDD rule. The two criteria of CP method are similar to that were presented in Equation 3.5, *i.e.*, $f_1(x) = \text{minimize } (p_i)$ and $f_2(x) = \text{minimize } (d_i)$. Equation 3.7 presented the Pareto objective function for the CP method.

The Pareto curves for this pair of conflicting objectives are presented in Figures 7.3 (a) and (b), which respectively show the Pareto curves for 1-month and 2-month factory data. In these figures, the extreme points are achieved by the SPT and EDD rules. All the points of CP0.33 coincide with the SPT point. For CP0.66 and CP1, the points get spread over a wider range and the curve shifts towards EDD point. For CP2, CP3 and CP10, all the points coincide with the EDD point. These results are similar to the results presented in Figures 3.4 and 3.5.

7.4.2 Mean cycle time and Cycle time variance

For this pair of conflicting objectives, the mean cycle time is minimized by the SPT rule and the cycle time variance is minimized by the P-W rule. Equation 3.9 presented these two criteria of CP method as $f_1(x) = \text{minimize } (p_i)$ and $f_2(x) = \text{minimize } (p_i/W_i)$. The Pareto objective function was presented in Equation 3.11.

7. Industrial Application: A Case Study on Photolithography Cluster

Mean Cycle Time vs. Maximum Tardiness (1-month data)

(a) Pareto control curve with 1-month data

Mean Cycle Time vs. Maximum Tardiness (2-month data)

(b) Pareto control curve with 2-month data

Figure 7.3: Case study: Pareto curves for mean cycle time and maximum tardiness

7. Industrial Application: A Case Study on Photolithography Cluster

(a) Pareto control curve with 1-month data

(b) Pareto control curve with 2-month data

Figure 7.4: Case study: Pareto curves for mean cycle time and cycle time variance

7. Industrial Application: A Case Study on Photolithography Cluster

The Pareto curves for this pair of conflicting objectives are presented in Figures 7.4 (a) and (b) for 1-month and 2-month factory data. In these figures, the extreme points are achieved by the SPT and P_W rules. These results are similar to the results presented in Figures 3.7 and 3.8. In Figures 7.4 (a) and (b), the points of CP10 curve give a wide spread of boundary between the SPT and P_W points. For CP3, the spread of points gets reduced. Further, as exponent value in CP method decreases, the spread of points gets reduced more and more. The points corresponding to $w = 0.05$ for various CP curves remain in the neighborhood of P_W point. On the other hand, the points corresponding to $w = 0.95$ for various CP curves move far away from SPT point as the exponent value decreases. Thus, the range of the achieved boundary gets reduced with the decrease in the exponent value of CP.

Under the given model and assumptions, the points corresponding to FCFS and the factory heuristic are far from the Pareto control curves in all these figures, Figure 7.3 (a) and (b) and Figure 7.4 (a) and (b). This observation reflects the importance of using the concept of Pareto control in order to trade-off the scheduling performance between the conflicting objectives. The characterization of Pareto control curves gives the ability to control the scheduling objectives within a specified range. Thus, the Pareto control curves can be used as a tool for the shop floor control management. By selecting an appropriate weight and exponent value, the decision maker can control the machine scheduling within the specific range of the scheduling objectives.

7.5 Case Study with Three Objectives

In this section, the above case study is extended to the case of simultaneously considering three objectives as briefly mentioned in Section 2.3. In order to present the concept of Pareto control surface, the three objectives considered are: minimizing mean cycle time, minimizing maximum tardiness and minimizing cycle time variance. These scheduling objectives are individually minimized by the corresponding schedul-

7. Industrial Application: A Case Study on Photolithography Cluster

ing criteria of SPT rule, EDD rule and P_W rule. These rules can be represented as $f_1(x) = \text{minimize } (p_i)$, $f_2(x) = \text{minimize } (d_i)$ and $f_3(x) = \text{minimize } (p_i/W_i)$. In applying CP method, the relative weights ($w_1, w_2, w_3; w_1 + w_2 + w_3 = 1$) of these three criteria are varied linearly from 0.05 to 0.95 at an interval of 0.10. Here, the CP exponent is kept constant at a value of 1.

The Pareto surface for these three conflicting objectives for 2-month factory data is presented in Figures 7.5 (a) and (b), which respectively show the Pareto surface from two different viewpoints. The viewpoint is specified in terms of azimuthal (az) and elevation (el) angles. The azimuth, az , is the horizontal rotation about the z-axis as measured in degrees from the negative y-axis. The elevation, el , is the vertical elevation of the viewpoint in degrees. In these figures, the extreme points are achieved by the SPT, EDD and P_W rules. This characterization of Pareto surface can enable the shop floor management to control the scheduling objectives within the specified range.

Figures 7.6 (a), (b) and (c) correspondingly present the orthographic projections of the Pareto surface on the x-y, x-z and y-z planes. These are respectively achieved by (a) $az = 0^\circ$ and $el = 90^\circ$, (b) $az = 0^\circ$ and $el = 0^\circ$ and (c) $az = 90^\circ$ and $el = 0^\circ$. Among these figures, Figure 7.6 (a) shows the conflicting nature of the objectives of minimizing mean cycle time and maximum tardiness. Similarly, Figure 7.6 (b) indicates the conflicting nature of the objectives of minimizing mean cycle time and cycle time variance. However, Figure 7.6 (c) expresses the opposite and confirms that the objectives of minimizing maximum tardiness and cycle time variance are not conflicting to each other.

If all the three objectives were conflicting to each other, then a real Pareto surface could have been generated. Such possibility exists in the case of scheduling a serial processing machine with sequence dependent setup times, which adds a third dimension of the objectives, mainly maximizing machine utilization or minimizing machine setups.

7. Industrial Application: A Case Study on Photolithography Cluster

(a) Projection view 1 at -37.5° azimuthal angle and 30° elevation angle

(b) Projection view 2 at -45° azimuthal angle and 60° elevation angle

Figure 7.5: Case study: two different projection views of the Pareto surface for the three objective Pareto control case

7. Industrial Application: A Case Study on Photolithography Cluster

Figure 7.6: Case study: orthographic projection views of the Pareto surface for the three objective Pareto control case

Such a case of multi-objective dynamic scheduling is presented in Sivakumar and Gupta (2006). Exploration of this case is still open to further research.

Chapter 8

Conclusions and Recommendations

In this concluding chapter, the research of this thesis is summarized. The originality of the research and its specific contribution to the field of multi-objective dynamic scheduling of serial and batch manufacturing processes are highlighted. Recommendations for improving and extending experimental investigations and theoretical analysis are presented featuring avenues for future research.

8.1 Originality of Research

The field of production scheduling is vast and well established over the last six decades. Likewise, the field of multi-objective scheduling has been researched in depth but mostly in static cases. In this thesis, some ideas and methods have been adopted from previous researchers and applied in dynamic cases. For clear demarkation, the ideas and methods that are original to this research are hereby highlighted.

1. Pareto-optimal solutions have by far been adopted as a standard technique in multi-objective optimization field. Researchers have developed various techniques to find out the complete set of Pareto-optimal solutions and thus to construct a complete Pareto-boundary. However, all the theory related to Pareto-optimal solutions is developed for static models with a limited number of jobs, especially in

the field of multi-objective scheduling. *The originality of the research presented in this thesis is in the application of the concept of Pareto control in a multi-objective dynamic scheduling environment, which has been dealt with as a multi-stage, multi-objective optimization problem.*

2. The use of look-ahead control policies at a batch processing machine have been in existence since early 1990. It is also well known that the exploitation of the available information about future arrival times leads to significant improvement in the cycle time performance. A similar concept has been exploited in this research for controlling the ET performance. *The originality of the proposed method lies in the idea of extracting the arrival and the due date information of future incoming lots and then analyzing a Pareto control between earliness and tardiness related measures.*
3. The earliness and tardiness related measures have been extensively used in scheduling research especially from the last two decades with the wide application of Just-in-Time manufacturing strategy. Researchers have developed various methods for optimizing these objectives. However, most of the existing research deals only with static cases for a serial processing machine. *In this thesis, a look ahead batching method is developed for minimizing ET performance at a batch processing machine in a dynamic scheduling environment.* In the semiconductor manufacturing environment, this research has wide scope of application especially for scheduling the bottleneck crucial operations such as oxidation furnaces in wafer fabs and burn-in-ovens in back-end.

8.2 Research Contributions

The research presented in this thesis reviews and substantiates the multi-objective scheduling methods. The research further extends the application of multi-objective

scheduling in dynamic environment of serial and batch processing machines. The extensive simulation experimental results and the analysis of schedule control in different scheduling environments discussed in this thesis provide a valuable addition to the knowledge base of this field. The specific research contributions of this thesis to the field of scheduling are as follows:

1. The first contribution of this research is the application of the concept of Pareto control in a dynamic environment for conflicting objectives. The concept of Pareto control has been applied to serial as well as batch processing machines for different conflicting objectives.
2. The second contribution is the research on the control of cycle time variability. For bi-objective dynamic scheduling on a serial processing machine, a schedule control is developed to trade-off between cycle time mean and cycle time variance.
3. The third contribution is the look ahead batching (LAB) algorithm for the objective of earliness and tardiness. The proposed method exploits the predicted information about the arrival times and due dates of the future incoming lots at a batch processing machine.
4. Fourthly, the concept of Pareto control has been applied with the LAB method between the conflicting objectives of earliness and tardiness related measures for the dynamic scheduling of a batch processing machine.

8.3 Conclusions and Findings

The connecting thread of the thesis is the application of the concept of Pareto control in the multi-objective dynamic scheduling environment. The emphasis has been laid on the ability to control the scheduling performance along the trade-off curves between the conflicting objectives. Pareto control has been applied for different pairs of conflicting

objectives in three scheduling environments: (i) serial processing, (ii) batch processing and (iii) look ahead batch processing. For each case, the sensitivity of achieving the Pareto control curves has also been studied for different experimental parameters.

Some essential conclusions and findings from the studies and experimental investigations are summarized below.

1. The concept of Pareto control has been applied for multi-objective dynamic scheduling in serial as well as batch processing environment. The study showed that by varying the relative weights of the selection criteria at each selection instance in simulation clock, a Pareto control curve can be achieved for two conflicting objectives. This characterization of Pareto control curves for the conflicting objectives can be exploited to enhance the ability to choose and operate the shop in a controlled way. By appropriate selection of parameters, the conflicting scheduling objectives can be controlled and maintained within a specified range for a particular operation. This Pareto control is highly relevant in a semiconductor manufacturing environment, where the demands of objectives frequently change according to the dynamic market forces. One such application has been presented in the case study on the real fab data.
2. The methodology of achieving Pareto control curves for two conflicting objectives includes first finding the selection criterion for optimizing each individual objective in dynamic scheduling and then forming a combined selection criterion using convex combination methods of WA and CP. It has been shown that using CP method gives wider control over achieving the trade-off curves between the conflicting objectives as CP method involves double weighing scheme with parameters of weight and exponent.
3. For the dynamic scheduling of a serial processing machine, the concept of Pareto control has been tested with respect to the two sets of conflicting objectives, (i)

mean cycle time *vs.* maximum tardiness and (ii) mean cycle time *vs.* variance of cycle time. While for the dynamic scheduling of a batch processing machine, the Pareto control has been tested only for the above first set of conflicting objectives.

4. The sensitivity of achieving Pareto control in serial processing has been experimented for the variation in the due-date tightness, the weight pattern and the number of replications in the simulation experiments. It has been shown that irrespective of the variations in these parameters, the Pareto control can still be achieved between the conflicting objectives.
5. The sensitivity of achieving Pareto control in batch processing has been particularly experimented for the variation in MBS value. It has been found that the Pareto control curve is highly sensitive to the variation in MBS value at low levels of utilization (0.5 and 0.6); while it becomes insensitive to the variation in MBS value at high utilization level (0.7).
6. For improving the ET performance in dynamic batch scheduling, a novel LAB method has been proposed and experimented against the existing DBH and NACH methods. By varying the due date pattern and the traffic intensity, simulation experiments are conducted to show the effect of LAB over DBH and NACH for two ET performance measures (a) average absolute sum of earliness and tardiness $\overline{(|E| + |T|)}$ and (b) average square sum of earliness and tardiness $\overline{(E^2 + T^2)}$. The steady state results of dynamic batch scheduling has indicated the well-improved performance of the LAB method under a wide variety of service rate and utilization level conditions. For average absolute sum of earliness and tardiness, LAB method gives more than 35% improvement over DBH and NACH, while more than 45% enhancement on average square sum of earliness and tardiness.
7. The above proposed LAB method for ET performance has been coupled with CP method in order to characterize the Pareto control curves for the earliness and

tardiness related conflicting objectives such as (a) average earliness *vs.* average tardiness and (b) number of early jobs *vs.* number of tardy jobs. In using CP method for achieving Pareto control in look ahead batching, it has been found that herein the exponent variation of CP formulation is ineffective while the variation of weight enables one to achieve control over the full breadth of the trade-off curves.

8. The sensitivity of achieving Pareto control in look ahead batching has been experimented for the variation in the traffic intensity, due date tightness and due date window. The developed methodology of using LAB method with CP formulation has been found to be capable of achieving Pareto control curves for the earliness and tardiness related conflicting objectives in all these variations. Developing Pareto control in look ahead batching between earliness and tardiness related conflicting objectives is very helpful for the production scheduling in JIT environment, where there is need to control the schedule of a batch processing machine in a dynamic environment according to the varying requirements of earliness and tardiness objectives.

8.4 Limitations of Present Research

This research presents an initial effort to investigate Pareto control curves in the multi-objective dynamic scheduling environment. The limitations of the present research are summarized as follows:

1. This research presented three models of Pareto control, *i.e.*, in serial processing, in batch processing and in look ahead batch processing. However, all the three models are limited to the processing of single family jobs on a single machine.
2. In all the three above mentioned models, there is no consideration of the setup

8. Conclusions and Recommendations

times associated with the jobs or the setup times are considered to be included in the processing times of the jobs.

3. The flow of job-arrivals on the machine should not be so low that the machine has to stay idle and wait for the job to arrive. Otherwise, there will be no option to select the Pareto job at each decision instance in the simulated time. In addition, in the case of Pareto control in look ahead batch processing, the flow of job arrivals should not be so high that at each decision instance a full batch is loaded on the machine for processing. Otherwise, all LAB scenarios will be similar which will make the evaluation of the effectiveness of LAB strategy difficult or almost impossible.
4. The presented concept of Pareto control in multi-objective dynamic scheduling is limited to one condition on the scheduling objectives. Each scheduling objective should have a corresponding scheduling criterion at each decision instance. This is due to the basic assumption that in dynamic scheduling, each objective is individually optimized by scheduling a job at each decision instance in simulated time according to the corresponding scheduling criterion.
5. In dynamic scheduling, there is no known way to define the Pareto optimum curve for multi-objective scheduling, except the enumeration method which works only for very small size problems.
6. In the present research, all models are considered with various combination of two objectives. This is primarily due to the limitation of orthogonality of one objective with another. In the case of considering more than two objectives, all objectives may not be orthogonal to each other, as presented in the case study example in Section 7.5. Also, the two objectives are considered in order to have a clear representation of Pareto curves in 2-D graphs.
7. Pareto control in batch processing as presented in Chapter 4 is limited to burn-

in oven type of batching. This is due to the fact that the individual objective of mean cycle time is minimized by selecting the jobs with minimum processing time. While, in the case of furnace type batching, all the jobs in one batch should be compatible to each other and so they have equal processing times.

8.5 Recommendations for Future Work

Future work primarily includes algorithm development and simulation experiments considering the effects of various approaches on parameters such as mean cycle time, cycle time variance, machine utilization, ET performance, *etc.*, in order to maintain the global competitiveness of the semiconductor industry. Based on the experience gained and the limitations observed during the study, the following are the recommendations for possible improvements and further investigations:

1. **Scheduling with other objectives:** Further research can explore the application of the concept of Pareto control to the scheduling problems in serial and batch processing environments with various other combination of objectives. These objectives may involve average cycle time, machine utilization, average tardiness, number of tardy jobs, average earliness, number of early jobs, *etc.*
2. **Scheduling with sequence dependent setups:** The concept of Pareto control needs to be extended to scheduling problems with sequence dependent setup times in serial processing machines, as mostly found in semiconductor manufacturing. Along with the cycle time performance and the ET performance, this will add one important dimension to the objectives: the utilization of the machine. This will lead to the development of a Pareto surface in 3-D space and thereafter controlling the schedule along the surface.
3. **Mathematical analysis of weight and exponent variation in CP method:** Further research is required on the mathematical modelling and analysis of the

CP method for multi-objective dynamic scheduling. CP method has two variants, the weight and the exponent, which are intricately correlated. Further analysis is required to find their correlation, which will enhance the control aspect of trade-off curves between the conflicting objectives.

4. **Look ahead batching with multiple families:** In this research, a look ahead batching method is developed for minimizing ET performance at a batch processing machine with a single family of products. This concept can be further developed for cases of multiple families and multiple parallel machines, which will be a more realistic case as far as the present semiconductor manufacturing industry is concerned.

Also, this model can be extended to the burn-in oven model of batching, where it will be very important to segregate the jobs according to their processing times in order to effectively utilize the batch capacity.

5. **Extension to job shop environment:** In this research, the MODS is considered primarily for a single machine case, either in a serial or batch processing environment. Further research direction also includes the application of MODS in a job shop environment with the ease of modelling while using conjunctive simulated scheduling approach.

References

- Aneja, Y., Kabadi, S., and Nagar, A. (1998). Minimizing weighted mean absolute deviation of flow times in single machine systems. *Naval Research Logistics*, 45:297–311. 20
- Arguello, M. (1994). Review of scheduling software. Technical report, SEMATECH, Austin, TX. Technology transfer 93091822A-XFR. 185
- Bagchi, U. (1989). Simultaneous minimization of mean and variation of flow time and waiting time in single machine systems. *Operations Research*, 37:118–125. 8, 21
- Baker, K. and Scudder, G. (1990). Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38(1):22–36. 23, 24, 127
- Blackstone, J., Philips, D., and Hogg, G. (1982). A state of art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, 20:27–45. 42, 43, 185
- Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., and Weglarz, J. (1996). *Scheduling computer and manufacturing processes*. Springer-Verlag, Berlin. 184
- Brucker, P. (1998). *Scheduling algorithms*. Springer-Verlag, Berlin. 184
- Brucker, P., Gladky, A., Hoogeveen, J., Kovalyov, M., Potts, C., Tauteenhahn, T., and van de Velde, S. (1998). Scheduling a batching machine. *Journal of Scheduling*, 1:31–54. 25, 88

REFERENCES

- Chandru, V., Lee, C., and Uzsoy, R. (1993a). Minimizing total completion time on a batch processing machine with job families. *Operations Research*, 13(2):61–65. 25, 26
- Chandru, V., Lee, C., and Uzsoy, R. (1993b). Minimizing total completion time on batch processing machines. *International Journal of Production Research*, 31(9):2097–2121. 25, 26, 86
- Chen, C.-L. and Bulfin, R. (1990). Scheduling unit processing time jobs on a single machine with multiple criteria. *Computers and Operations Research*, 17(1):1–7. 15
- Conway, R. and Maxwell, W. (1962). Network dispatching by the shortest-operation discipline. *Operations Research*, 10(1):51–73. 43, 49
- Dabbas, R., Chen, H.-N., Fowler, J., and Shunk, D. (2001). A combined dispatching criteria approach to scheduling semiconductor manufacturing systems. *Computers and Industrial Engineering*, 39:307–24. 7, 9
- Dabbas, R. and Fowler, J. (2003). A new scheduling approach using combined dispatching criteria in wafer fabs. *IEEE Transaction on Semiconductor Manufacturing*, 16(3):501–510. 9
- Das, I. and Dennis, J. (1997). A closer look at the drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69. 36
- De, P., Ghosh, J., and Wells, C. (1992). On the minimization of completion time variance with a bicriteria extension. *Operations Research*, 40(6):1148–1155. 21
- Deo, N. (1993). *System simulation with digital computer*. Prentice Hall of India, New Delhi. 185
- Dileepan, P. and Sen, T. (1988). Bicriterion state scheduling research for a single machine. *Omega*, 16:53–59. 15, 184

REFERENCES

-
- Emmons, H. (1975). One machine sequencing to minimize mean flow time with minimum number tardy. *Naval Research Logistics Quarterly*, 22:585–592. 18
- Fowler, J., Hogg, G., and Phillips, D. (2000). Control of multiproduct bulk service diffusion/oxidation processes. part 2: Multiple servers. *IIE Transactions*, 32:167–176. 10, 27, 28
- Fowler, J., Phillips, D., and Hogg, G. (1992). Real-time control of multiproduct bulk-service semiconductor manufacturing processes. *IEEE Transaction on Semiconductor Manufacturing*, 5(2):158–163. xii, 10, 27, 28, 106, 111, 112
- French, S. (1982). *Sequencing and scheduling: an introduction to the mathematics of the job-shop*. John Wiley and Sons, New York. xiv, 5
- Fry, T., Armstrong, R., and Lewis, H. (1989). A framework for single machine multiple objective sequencing approach. *Omega*, 17:595–607. 15, 184
- Garey, M., Tarjan, R., and Wilfong, G. (1988). One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research*, 13:330–348. 22
- Geoffrion, A. (1968). Proper efficiency and the theory of vector optimization. *Journal of Mathematical Analysis and Application*, 41:491–502. 29, 30, 35
- Glassey, C. and Weng, W. (1991). Dynamic batching heuristic for simultaneous processing. *IEEE Transaction on Semiconductor Manufacturing*, 4(2):77–82. xi, 10, 26, 27, 90, 106, 109, 110, 111
- Goicoechea, A., Hansen, D., and Duckstein, L. (1982). *Multi-objective decision analysis with engineering and business applications*. John Wiley & Sons, New York. 33
- Gordon, V., Proth, J.-M., and Chu, C. (2002). A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research*, 139:1–25. 24

REFERENCES

-
- Graham, R., Lawler, E., Lenstra, J., and Kan, A. R. (1979). Optimization and approximation in deterministic machine scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326. 2
- Gupta, A. and Sivakumar, A. (2002). Simulation based multi-objective schedule optimization in semiconductor manufacturing. In *Proceedings of Winter Simulation Conference*, pages 1862–1870, San Diego, California, USA. 33, 38
- Gupta, A. and Sivakumar, A. (2005). Conjunctive simulated scheduling. *International Journal of Advanced Manufacturing Technology*, 26(11):1409–1413. 32, 186
- Gupta, A. and Sivakumar, A. (2006). Job shop scheduling techniques in semiconductor manufacturing. *International Journal of Advanced Manufacturing Technology*, 27(11):1163–1169. 186
- Haupt, R. (1989). A survey of priority rule based scheduling. *OR Spectrum*, 11:3–16. 42, 185
- Heck, H. and Roberts, S. (1972). A note on the extension of a result on scheduling with secondary criteria. *Naval Research Logistics Quarterly*, 19:403–405. 17
- Holthaus, O. and Rajendran, C. (1997). Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics*, 48:87–105. 42
- Hoogeveen, J. and van de Velde, S. (1995). Minimizing total completion time and maximum cost simultaneously is solvable in polynomial time. *Operations Research Letters*, 17:205–208. 19
- Hopp, W. and Spearman, M. (2000). *Factory Physics: foundations of manufacturing management*. Irwin McGraw-Hill, Singapore. 9, 16, 185, 186
- Hung, Y. (1998). Scheduling of mask shop e-beam writers. *IEEE Transaction on Semiconductor Manufacturing*, 11(1):165–172. 25, 28

REFERENCES

-
- Ikura, Y. and Gimple, M. (1986). Efficient scheduling algorithms for a single batch processing machine. *Operations Research Letters*, 5(2):61–65. 26, 27
- Kanet, J. (1981). Minimizing variation of flow time in single machine systems. *Management Science*, 27:1453–1459. 20
- Keeney, R. and Raiffa, H. (1976). *Decisions with multiple objectives: preferences and value tradeoffs*. John Wiley and Sons, Inc. 33
- Kim, Y.-D., Kim, J.-G., Choi, B., and Kim, H.-U. (2001). Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates. *IEEE Transaction on Semiconductor Manufacturing*, 17(5):589–598. 29, 76
- Kim, Y.-D., Kim, J.-U., Lim, S.-K., and Jun, H.-B. (1998). Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility. *IEEE Transaction on Semiconductor Manufacturing*, 11(1):155–164. 52, 53, 76
- Koksalan, M. (1999). A heuristic approach to bicriteria scheduling. *Naval Research Logistics*, 46:777–789. 8, 19, 40, 57
- Koulamas, C. (1996). Single-machine scheduling with time windows and earliness/tardiness penalties. *European Journal of Operational Research*, 91:190–202. 8
- Law, A. and Kelton, W. (1984). Confidence intervals for steady-state simulations: a survey of fixed sample size procedures. *Operations Research*, 32:1221–1239. 49, 94
- Law, A. and Kelton, W. (2000). *Simulation modeling and analysis: third edition*. McGraw-Hill Book Co., Singapore. 47, 48
- Lee, C. and Choi, J. (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. 23

REFERENCES

-
- Lee, C., Uzsoy, R., and Martin-Vega, L. (1992). Efficient algorithms for scheduling burn-in operations. *Operations Research*, 40(4):764–775. 25, 26, 86
- Lu, S., Ramaswamy, D., and Kumar, P. R. (1994). Efficient scheduling policies to reduce mean and variance of cycle time in semiconductor manufacturing plants. *IEEE Transaction on Semiconductor Manufacturing*, 7(3):374–388. 42
- Merten, A. and Muller, M. (1972). Minimization in single machine sequencing problems. *Management Science*, 18:518–528. 20
- Mittenthal, J., Raghavachari, M., and Rana, A. (1996). A class of single machine central tendency - dispersion bicriteria problems. *Journal of the Operational Research Society*, 47:1355–1365. 20
- Montgomery, D. (1997). *Design and analysis of experiments: fourth edition*. John Wiley & Sons, New York. 54, 61, 187
- Morton, T. and Pentico, D. (1993). *Heuristic scheduling systems with application to production systems and project management*. John Wiley & Sons, New York. 186
- Mosheiov, G. (2000). Minimizing mean absolute deviation of job completion times from the mean completion time. *Naval Research Logistics*, 47:657–668. 20
- Nagar, A., Haddock, J., and Heragu, S. (1995). Multiple and bicriteria scheduling: a literature survey. *European Journal of Operational Research*, 81:88–104. 9, 15, 184
- Nandkeolyar, L., Ahmed, M., and Sundararaghavan, P. (1993). Dynamic single machine weighted absolute deviation problem: predictive heuristics and evaluation. *International Journal of Production Research*, 31:1453–1466. 24
- Neuts, M. (1967). A general class of bulk queues with poisson input. *Management Science*, 38(3):759–770. xi, 90, 91
- Ow, P. and Morton, T. (1989). The single machine early-tardy problem. *Management Science*, 35(2):177–191. 22

REFERENCES

-
- Panwalker, S. and Iskander, W. (1977). A survey of scheduling rules. *Operations Research*, 25:45–61. 185
- Pinedo, M. (1995). *Scheduling: theory, algorithms and systems*. Prentice Hall, Englewood Cliffs, NJ. 2, 9, 184
- Pinedo, M. and Chao, X. (1999). *Operations scheduling with applications in manufacturing and services*. Irwin McGraw-Hill, Singapore. 7, 9, 186
- Potts, C. and Kovalyov, M. (2000). Scheduling with batching: A review. *European Journal of Operational Research*, 120:228–249. 2, 26, 184
- Qi, C., Tang, T., and Sivakumar, A. (2002). Simulation based cause and effect analysis of cycle time and WIP in semiconductor wafer fabrication. In *Proceedings of Winter Simulation Conference*, pages 1423–1430, San Diego, California, USA. 151
- Qi, X. and Tu, F. (1999). Earliness and tardiness scheduling problems on a batch processor. *Discrete Applied Mathematics*, 98:131–145. 24
- Raghavachari, M. (1988). Scheduling problems with non-regular penalty functions - a review. *Opsearch*, 25(3):144–164. 184
- Ramasesh, R. (1990). Dynamic jobshop scheduling: A survey of simulation research. *OMEGA*, 16:43–57. 42, 185
- Robinson, J., Fowler, J., and Bard, J. (1995). The use of upstream and downstream information in scheduling semiconductor batch operations. *International Journal of Production Research*, 33(7):1849–1869. 10, 27
- Russell, R., Dar-el, E., and Taylor, B. (1987). A comparative analysis of the covert job sequencing rule using various shop performance measures. *International Journal of Production Research*, 25:1523–1540. 42
- Sanjay, V. and Uzsoy, R. (1998). Minimizing total tardiness on a batch processing machine with incompatible job families. *IIE Transactions*, 30:165–178. 25, 26

REFERENCES

-
- Sawaragi, Y., Nakayama, H., and Tanino, T. (1985). *Theory of multi-objective optimization*. Academic Press Inc., Orlando, Florida. 3, 33
- Sen, T. and Gupta, S. (1983). A branch and bound procedure to solve a bicriterion problem. *IIE Transactions*, 15:84–88. 18, 19
- Sidney, J. (1977). Optimal single machine scheduling with earliness and tardiness penalties. *Operations Research*, 25(1):62–69. 22
- Sivakumar, A. (2001). Multi-objective dynamic scheduling using discrete event simulation. *International Journal of Computer Integrated Manufacturing*, 14(2):154–167. 7, 9
- Sivakumar, A. and Gupta, A. (2006). On-line multi-objective pareto optimal dynamic scheduling of semiconductor back-end using conjunctive simulated scheduling. *IEEE Transactions on Electronics Packaging Manufacturing*, 29(2):99–109. 162
- Smith, W. (1956). Various optimizers for single stage production. *Naval Research Quarterly*, 3:59–66. 17, 18
- Sridharan, V. and Zhou, Z. (1996). A decision theory based scheduling procedure for single machine weighted earliness and tardiness problem. *European Journal of Operational Research*, 94:292–301. 24
- Steuer, R. E. (1986). *Multiple criteria optimization: theory, computation, and application*. Krieger Publishing Company, Malabar, Florida. 33, 40
- Sung, C., Choung, Y., Hong, J., and Kim, Y. (2002). Minimizing makespan on a single burn-in oven with job families and dynamic job arrivals. *Computers and Operations Research*, 29:995–1007. 86
- Taboun, S., Abib, A., and Atmani, A. (1995). Generating efficient points of bicriteria scheduling problem by using compromise programming. *Computers and Industrial Engineering*, 29(1–4):227–231. 40

REFERENCES

-
- Tabucanon, M. (1988). *Multiple criteria decision making in industry*. Elsevier Science Publishers B.V., Amsterdam, The Netherlands. 4, 33, 34, 39
- T'kindt, V. and Billaut, J.-C. (2001). Multicriteria scheduling problems: a survey. *RAIRO Operations Research*, 35:143–163. 9, 16, 33, 183, 184
- T'kindt, V. and Billaut, J.-C. (2002). *Multicriteria scheduling: theory, models and algorithms*. Springer-Verlag, Heidelberg. 2, 7, 16, 24, 29, 33
- Uzsoy, R. (1994). Scheduling a single batch processing machine with non-identical job size. *International Journal of Production Research*, 32(7):1615–1635. 25, 26, 86
- VanWassenhove, L. and Gelders, L. (1980). Solving a bicriterion scheduling problem. *European Journal of Operational Research*, 4(1):42–48. 5, 8, 15, 18, 19
- Ventura, J. and Radhakrishnan, S. (2003). Single machine scheduling with symmetric earliness and tardiness penalties. *European Journal of Operational Research*, 144:598–612. 8, 23, 184
- Webster, S. and Baker, K. (1995). Scheduling groups of jobs on a single machine. *Operations Research*, 43(4):692–703. 25, 184
- Weng, W. and Leachman, R. (1993). An improved methodology for real-time production decisions at batch-process work stations. *IEEE Transaction on Semiconductor Manufacturing*, 6(3):219–225. 27
- Wu, Z. and Weng, M. (2005). Multiagent scheduling method with earliness and tardiness objectives in flexible job shops. *IEEE Transaction on Systems, Man, and Cybernetics*, 35(2):293–301. 22
- Xiong, H. and Zhou, M. (1998). Scheduling of semiconductor test facilities via petri nets and hybrid heuristic search. *IEEE Transaction on Semiconductor Manufacturing*, 11(3):384–393. 185

REFERENCES

- Yang, J. and Chang, T.-S. (1998). Multi-objective scheduling for IC sort and test with a simulation testbed. *IEEE Transaction on Semiconductor Manufacturing*, 11(2):304–315. 7
- Yu, P.-L. (1973). A class of solutions for group decision problems. *Management Science*, 19:936–946. 33
- Yu, P.-L. (1985). *Multiple-criteria decision making*. Plenum press., New York. 3, 33, 34, 36
- Yu, P.-L. and Leitmann, G. (1974). Compromise solutions, dominations structures, and salukvadze’s solution. *Journal of the Optimization Theory and Applications*, 13:362–378. 36
- Zeleny, M. (1973). *Multiple criteria decision making*, chapter Compromise programming, pages 262–301. University of South Carolina Press, Columbia. 33, 36
- Zeleny, M. (1982). *Multiple criteria decision making*. McGraw-Hill Book Company, New York. 33, 34, 36, 37, 38, 39, 95
- Zhou, M. and Jeng, M. (1998). Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: a petri net approach. *IEEE Transaction on Semiconductor Manufacturing*, 11(3):333–357. 185

Appendix A

Diversity of Scheduling

Scheduling problems, models and related literature range from deterministic to stochastic models, from single machine to multiple machine problems, from static to dynamic models, from single objective to multi-objective problems and so on. The diversity of scheduling is developed and presented in breadth of the scheduling domain in Figure A.1, which is described in the following sections.

Shop configuration

Based on the number of machines available on the shop floor different shop configurations are possible, which can be broadly classified in two cases, namely, single and multiple machines. When only one machine processes all the jobs, it is known as the *single machine* problem. In this case, the problem is to find an order in which the jobs must be processed on this machine. In contrast to real world situations, this case is highly simplistic; nevertheless it lays the foundation for the more complex multiple machine case. Single machine problem is also useful to model and study bottleneck problems, which occur frequently on the shop floor. An overall shop can also be seen as a ‘single machine’ in an aggregate formulation of the problem.

More complex configurations arise when there are multiple machines on the shop floor. The configurations, in increasing order of their complexity, are *parallel machine*, *flow shop* and *job shop*. A shop is said to have parallel machine configuration when there are a number of identical machines and the single operation jobs can be processed on any of these machines. Flow shop is a configuration in which machines are arranged in a serial fashion; each job has to pass through each machine. As the jobs are processed by each machine, they are multi-stage in nature and each stage defines an operation. The most complex of all cases is the job shop configuration in which the operations of the jobs are equally likely to be assigned to any machine for processing. In this case, there is no common pattern for job routing.

Job characteristics

If the job characteristics, *e.g.* processing times (p_i), release dates (r_i) and due dates (d_i), can be ascertained with certainty, the problem is modelled as a *deterministic* scheduling

Figure A.1: Diversity of scheduling

problem. Otherwise, the problem is considered *stochastic* where the parameters are modelled as random variables with some known probability distribution. Stochastic models, in general, capture real life situations better than deterministic models as the parameters in the stochastic models are not assigned with the specific values.

Nature of job arrival

With all jobs being available at the beginning of the scheduling period (*i.e.*, $r_i = 0, \forall i$), the problem is modelled as *static*. When the jobs go on arriving discretely over time and each new job joins an existing queue in front of the machine, the model is considered as *dynamic* (T'kindt and Billaut, 2001).

Type of job processing

When a machine processes only a single job at a time and each job has its unique processing time and setup time, then it is called *discrete/serial processing*. On the other hand, when many jobs are grouped and processed together to avoid multiple setup of the machine, it is called *batch processing*.

Batch processing mode can further be categorized in two types of situations. In one situation, jobs are partitioned into families according to their similarity so that no setup is required for a job if it belongs to the same family of the previously processed job. The setup may reflect the need to change a tool or to clean the machine. A setup time is required at the start of the schedule and on each occasion when the machine

switches from processing jobs in one family to jobs in another family. This is called *serial batching*. In this model, a batch is a maximal set of jobs that are scheduled contiguously on a machine and share a setup. Large batches have the advantage of high machine utilization because the number of setups is small. On the other hand, processing a large batch may delay the processing of an important job belonging to a different family (Potts and Kovalyov, 2000).

In another situation, a batching machine may be capable of processing several jobs simultaneously. This is referred to as *parallel batching*; and is found in burn-in ovens in semiconductor back-end, oxidation furnaces in wafer fabs and kilns in chemical processes. Based on the batch processing time, parallel batching can further be divided primarily into two: burn-in oven model and furnace model (Webster and Baker, 1995). In burn-in oven model, the jobs from different families, *i.e.*, the incompatible jobs can form a batch. The batch processing time is equal to the longest processing time of the jobs in a batch. In furnace model, only the jobs from a single family, *i.e.*, the compatible jobs form a batch. The batch processing time is constant and equal to the processing time of the family.

Scheduling objective

A vast amount of the scheduling literature focuses on *single objective* scheduling, where only one scheduling objective is optimized. While, in *multi-objective scheduling*, many conflicting objectives are optimized simultaneously. Multi-objective scheduling literature is thoroughly reviewed by Dileepan and Sen (1988); Fry et al. (1989); Nagar et al. (1995); T'kindt and Billaut (2001), *etc.* Literature relevant to this thesis on multi-objective scheduling is presented in Section 2.1

Performance measure

A performance measure is said to be *regular* if it is a non-decreasing function of job completion times and the scheduling objective is to minimize the performance measure (Ventura and Radhakrishnan, 2003). Objectives such as the minimization of mean flowtime, percentage of tardy jobs, and mean tardiness are regular measures. A large number of scheduling problems have been studied with regular performance measures, mostly related to flowtime and tardiness.

With the advent of JIT philosophy, the deliveries are desired to be neither early nor late. This has shifted the focus from regular to non-regular performance measures. A *non-regular* performance measure is usually not a monotone function of the job completion times. Measures such as earliness, the sum of earliness and tardiness penalties, the completion time variance, *etc.* are non-regular measures (Raghavachari, 1988).

Applications

Scheduling problems are encountered in production lines (Pinedo, 1995), in computer systems (Blazewicz et al., 1996), in project management (Brucker, 1998), in timetabling of educational establishments, *etc.*

Scheduling approaches

The Non-deterministic Polynomial-time hard (NP-hard) problems are problems for which there is no known polynomial algorithm, so that the time to find a solution grows exponentially in problem size. Since most real-world scheduling problems fall into the NP-hard category and tend to be large, it quite literally means that it is almost impossible to solve many realistically sized scheduling problems optimally (Hopp and Spearman, 2000). Therefore, all the scheduling approaches available in commercial software systems make use of heuristics and hence none produces an optimal schedule. Hopp and Spearman (2000) address these scheduling approaches in two categories: *simulation-based methods* and *optimization-based methods*. Arguello (1994) provides an excellent survey of scheduling software (both simulation-based and optimization-based) used in the semiconductor industry.

Simulation-based methods: Simulation-based methods necessarily require a simulation model to perform the scheduling task. One way to avoid the NP-hard optimization problem is to simply ignore it by developing a detailed and deterministic simulation model of the entire system (Hopp and Spearman, 2000). To generate a schedule, the model is run forwards in time and records the arrival and departure of jobs at each station. Different schedules are generated by applying various dispatching rules at each station, which are then evaluated according to selected performance measures to find the “best” schedule.

Researchers have been using discrete event system simulation as the tool for scheduling large, complex and dynamic manufacturing systems as no other technique has been developed to solve these problems satisfactorily (Deo, 1993). A good review of simulation-based approaches for various types of environments and problems can be found in Panwalker and Iskander (1977), Blackstone et al. (1982), Haupt (1989) and Ramasesh (1990). They provide an insight into the aspects of job shop scheduling using discrete event simulation.

The advantage of the simulation-based approach is that it is easier to explain than most optimization-based methods. Since a simulator mimics the behavior of the actual system in an intuitive way, one can easily understand its logic. Another advantage is that it can quickly generate a variety of different schedules by simply changing the dispatching rules according to the specific objective requirements and then reporting performance statistics. However, the disadvantages of simulation-based approach include maintaining an enormous amount of data for simulation, no accounting for variability in the schedules, finding an effective schedule by trial-and-error process, *etc.*

Petri net model is another simulation-based approach of scheduling the shops. Zhou and Jeng (1998) presented a tutorial on Petri nets, introducing definitions and concepts of Petri nets, discussing system modeling, presenting their properties and analysis methods, and reviewing applications of Petri nets in semiconductor manufacturing automation. They also presented a case study to show how a Petri net methodology can be applied to a complex real world system, by illustrating through Petri net modeling, analysis, and simulation of the photolithography area of an IC wafer fab. Xiong and Zhou (1998) proposed two Petri net-based hybrid heuristic search strategies and their application to semiconductor test facilities scheduling with a size limit of 79 resources and 30 jobs. The limitations of Petri Net model based simulation approach are the size

of the problem and the multiple objective consideration.

Due to the vast complexity of scheduling problems in semiconductor manufacturing such as mix of serial and batch processors, reentrance, mix product type and trade-off among multiple objectives, the often used scheduling approach is simulation-based scheduling.

Optimization-based methods: Unlike classical optimization, optimization-based scheduling techniques use heuristic procedures which lack in guarantees of performance. It differs from simulation-based scheduling in that it uses some sort of algorithm to actively search for a good schedule. Morton and Pentico (1993) and Pinedo and Chao (1999) give a detailed description of these techniques. These techniques include exact methods such as branch-and-bound technique, dynamic programming, Lagrangian approximation, queuing networks, *etc.*, neighborhood search techniques such as tabu search, genetic algorithms, simulated annealing, *etc.*, artificial intelligence (AI) techniques such as neural networks, fuzzy logic, expert systems *etc.* and dispatching heuristics. For the purpose of brevity, these techniques are not reviewed here. However Gupta and Sivakumar (2006) provide a thorough review of these techniques for solving the scheduling problems in semiconductor manufacturing environment.

The primary problem with these techniques manifests in a dynamic shop environment as it requires the schedule to be modified again and again with great endeavor according to the real-time information. Secondly, in complex job shops where sequence dependent setup times, multiple product families, re-entrant process flow *etc.* exist, it becomes increasingly unachievable to optimize the schedule with a single scheduling objective. Another problem with optimization-based scheduling is that many practical scheduling problems are not really optimization problems at all. Rather, they are better characterized as *satisficing* problems (Hopp and Spearman, 2000).

However, many optimization-based scheduling methods also make use of simulation either as a tool for searching the optimum solution or as a tool to study the behavior and to evaluate the superiority of one technique over another. This leads to a confusion in the understanding of the simulation-based scheduling methods among the researchers especially from the field of Operational Research. The question arises how to clearly distinguish between the optimization-based and simulation-based scheduling methods. In an attempt to clear away this misunderstanding, the simulation-based approach of scheduling is referred to as Conjunctive Simulated Scheduling (CSS) (Gupta and Sivakumar, 2005). CSS is considered as an approach of scheduling, where discrete event simulation is conjuncted with the scheduling criteria. The scheduling decisions are simulated at each decision instance, whenever either the resource becomes available or a new job arrives at an idle resource. A task has to be selected for the next operation on this resource and then the simulated clock is moved forward for the next decision instant to repeat the procedure. Thus the effect of simulation is conjuncted with the scheduling of the shop and thus the terminology “Conjunctive Simulated Scheduling” is used (Gupta and Sivakumar, 2005). This makes it clearly distinct from the optimization-based methods.

Appendix B

Duncan's Multiple Range Test

Duncan's multiple range test is a widely used post hoc test for comparing all pairs of means (Montgomery, 1997). Post hoc tests need to be undertaken after Analysis Of Variance (ANOVA), whenever there are more than two means. The results in one way ANOVA table serve only to indicate whether the means of factors differ significantly or not. These do not indicate which means differ from others. To determine this, post hoc tests are conducted after finding the significant difference in ANOVA.

In Chapter 3, a sample illustration of Duncan's multiple range test is presented. Table 3.3 presented the RDI values of mean cycle times for all the ten runs of various priority indices at $\rho = 0.75$. ANOVA and Duncan's multiple range tests are conducted using SPSS 10 software. The results of Duncan's multiple range test were presented in Table 3.4.

The results of ANOVA test as obtained from SPSS 10 software are presented for all the three objectives in Figures B.1 and B.2 for $\rho = 0.75$ and 0.85 respectively. ANOVA result states that the null hypothesis of all means being equal does not hold true, *i.e.*, the means of various priority indices significantly differ from each other. Further, the results of Duncan's multiple range test for $\rho = 0.75$ and 0.85 were presented in Tables 3.4, 3.5 and 3.6 respectively for mean cycle time, max. tardiness and cycle time variance.

*B. Duncan's Multiple Range Test*Descriptives ($\rho = 0.75$):

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
Priority index no.						Lower Bound	Upper Bound		
Mean CT	1	10	.541520	.0599467	.0189568	.498637	.584404	.4545	.6281
	2	10	1.000000	.0000000	.0000000	1.000000	1.000000	1.0000	1.0000
	3	10	.278972	.1295658	.0409723	.186286	.371658	.0000	.4152
	4	10	.687283	.0808309	.0255610	.629460	.745106	.5695	.8189
	5	10	.256319	.0899314	.0284388	.191986	.320652	.1304	.3967
	6	10	.685612	.0802083	.0253641	.628235	.742990	.5695	.8179
	7	10	.430021	.0483388	.0152861	.395442	.464601	.3693	.5006
	8	10	.012001	.0379493	.0120006	-.015147	.039148	.0000	.1200
	Total	80	.486466	.3001583	.0335587	.419669	.553263	.0000	1.0000
CT Var	1	10	.026494	.0375260	.0118668	-.000350	.053339	.0020	.1311
	2	10	.900243	.1347680	.0426174	.803835	.996650	.6798	1.0000
	3	10	.326551	.1472732	.0465719	.221198	.431904	.0514	.4792
	4	10	.061739	.0482451	.0152565	.027227	.096252	.0192	.1903
	5	10	.924071	.0989283	.0312839	.853302	.994840	.7789	1.0000
	6	10	.061006	.0481573	.0152287	.026557	.095456	.0188	.1898
	7	10	.008287	.0262052	.0082868	-.010459	.027033	.0000	.0829
	8	10	.202004	.1329322	.0420369	.106910	.297098	.0000	.3936
	Total	80	.313799	.3729755	.0416999	.230798	.396801	.0000	1.0000
Max Tar	1	10	.000908	.0013777	.0004357	-.000077	.001894	.0000	.0043
	2	10	.773493	.2090159	.0660966	.623972	.923014	.3845	1.0000
	3	10	.372335	.1471104	.0465204	.267099	.477572	.1931	.5901
	4	10	.006221	.0122715	.0038806	-.002558	.014999	.0000	.0387
	5	10	.969161	.0830936	.0262765	.909719	1.028602	.7352	1.0000
	6	10	.006236	.0122746	.0038816	-.002545	.015017	.0000	.0387
	7	10	.035863	.0117906	.0037285	.027429	.044298	.0103	.0513
	8	10	.595498	.2568899	.0812357	.411730	.779266	.2600	1.0000
	Total	80	.344964	.3905494	.0436647	.258052	.431877	.0000	1.0000

Test of Homogeneity of Variances:

	Levene Statistic	df1	df2	Sig.
Mean CT	6.296	7	72	.000
CT Var	6.758	7	72	.000
Max Tar	13.086	7	72	.000

ANOVA:

		Sum of Squares	df	Mean Square	F	Sig.
Mean CT	Between Groups	6.711	7	.959	169.627	.000
	Within Groups	.407	72	.006		
	Total	7.118	79			
CT Var	Between Groups	10.323	7	1.475	159.323	.000
	Within Groups	.666	72	.009		
	Total	10.990	79			
Max Tar	Between Groups	10.802	7	1.543	89.025	.000
	Within Groups	1.248	72	.017		
	Total	12.050	79			

Figure B.1: Results of ANOVA test for RDI values of Mean Cycle Time, Cycle Time Variance and Max. Tardiness at $\rho = 0.75$ from SPSS 10 output window

B. Duncan's Multiple Range Test

Descriptives ($\rho = 0.85$):

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
Priority index no.						Lower Bound	Upper Bound		
Mean CT	1	10	.503783	.1014517	.0320818	.431208	.576357	.2566	.6066
	2	10	1.000000	.0000000	.0000000	1.000000	1.000000	1.0000	1.0000
	3	10	.275708	.0871636	.0275635	.213354	.338061	.1518	.4288
	4	10	.594156	.1285680	.0406568	.502184	.686128	.2713	.7340
	5	10	.245923	.0677844	.0214353	.197433	.294413	.1276	.3204
	6	10	.593938	.1284475	.0406187	.502052	.685824	.2715	.7340
	7	10	.393034	.0790644	.0250024	.336475	.449594	.2073	.4882
	8	10	.000000	.0000000	.0000000	.000000	.000000	.0000	.0000
	Total	80	.450818	.2934398	.0328076	.385516	.516119	.0000	1.0000
CT Var	1	10	.011587	.0099468	.0031454	.004472	.018703	.0002	.0275
	2	10	.938541	.1335430	.0422300	.843011	1.034072	.5833	1.0000
	3	10	.317463	.1306265	.0413077	.224018	.410907	.1071	.5652
	4	10	.025540	.0144277	.0045624	.015219	.035861	.0006	.0461
	5	10	.894486	.1028760	.0325323	.820893	.968079	.7360	1.0000
	6	10	.025499	.0144512	.0045699	.015161	.035836	.0006	.0461
	7	10	.000000	.0000000	.0000000	.000000	.000000	.0000	.0000
	8	10	.173389	.0933507	.0295201	.106610	.240168	.0275	.3112
	Total	80	.298313	.3816291	.0426674	.213386	.383240	.0000	1.0000
Max Tar	1	10	.001161	.0012071	.0003817	.000298	.002025	.0000	.0036
	2	10	.788514	.2151849	.0680474	.634580	.942448	.4597	1.0000
	3	10	.415989	.1619036	.0511984	.300170	.531808	.2033	.7274
	4	10	.001296	.0038432	.0012153	-.001453	.004045	.0000	.0122
	5	10	.995907	.0105815	.0033462	.988338	1.003477	.9665	1.0000
	6	10	.001299	.0038421	.0012150	-.001449	.004048	.0000	.0122
	7	10	.049174	.0277848	.0087863	.029298	.069050	.0090	.0918
	8	10	.623549	.2284198	.0722327	.460148	.786951	.2852	.8971
	Total	80	.359611	.3987883	.0445859	.270865	.448357	.0000	1.0000

Test of Homogeneity of Variances:

	Levene Statistic	df1	df2	Sig.
Mean CT	2.978	7	72	.008
CT Var	6.138	7	72	.000
Max Tar	16.363	7	72	.000

ANOVA:

		Sum of Squares	df	Mean Square	F	Sig.
Mean CT	Between Groups	6.247	7	.892	115.583	.000
	Within Groups	.556	72	.008		
	Total	6.802	79			
CT Var	Between Groups	11.013	7	1.573	230.056	.000
	Within Groups	.492	72	.007		
	Total	11.506	79			
Max Tar	Between Groups	11.433	7	1.633	104.025	.000
	Within Groups	1.130	72	.016		
	Total	12.564	79			

Figure B.2: Results of ANOVA test for RDI values of Mean Cycle Time, Cycle Time Variance and Max. Tardiness at $\rho = 0.85$ from SPSS 10 output window

Appendix C

Output Data Tables

Table C.1: Results of Pareto control in serial processing for Mean Cycle Time and Maximum Tardiness ($\rho = 0.75$)

	CP 0.33		CP 0.66		CP 1		CP 2	
w	Mean CT	T_max	Mean CT	T_max	Mean CT	T_max	Mean CT	T_max
0.05	60.61	664.9	63.13	309.37	68.14	232.98	70.27	217.83
0.1	60.28	671.77	61.7	387.31	65.35	254.96	70.12	217.95
0.15	60.23	673.27	61.19	449.79	63.93	275.75	69.99	219.66
0.2	60.22	673.29	60.87	485.34	63.05	288.09	69.84	219.78
0.25	60.2	669.58	60.69	506.31	62.59	310.59	69.69	219.88
0.3	60.22	669.52	60.44	541.43	62.2	332.27	69.42	221.72
0.35	60.23	693.46	60.31	585.13	61.96	355.95	69.2	221.99
0.4	60.21	687.63	60.37	621.29	61.77	381	68.96	221.55
0.45	60.19	686.39	60.27	630.96	61.49	377.2	68.69	223.89
0.5	60.15	686.39	60.22	613.86	61.34	395.85	68.5	225.09
0.55	60.16	678.55	60.14	642.15	61.05	421.43	68.22	223.87
0.6	60.15	678.55	60.13	649.35	60.91	438.94	67.92	225.54
0.65	60.16	675.06	60.07	653.12	60.73	462.02	67.59	230.72
0.7	60.16	670.59	60.12	664.7	60.65	464.89	67.22	225.3
0.75	60.16	670.59	60.13	661.21	60.57	492.27	66.78	233.46
0.8	60.16	670.59	60.15	670.6	60.35	522.57	66.21	248.39
0.85	60.16	670.59	60.16	670.59	60.17	555.73	65.5	272.05
0.9	60.16	670.59	60.16	670.59	60.26	600.22	64.53	286.74
0.95	60.16	670.59	60.16	670.59	60.22	625.47	63.28	306.33
	CP 3		CP 10		WA		Extreme Points	
w	Mean CT	T_max	Mean CT	T_max	Mean CT	T_max		
0.05	70.24	217.78	70.07	217.74	61.79	450.21	Min (p)	
0.1	70.19	217.85	70.05	217.77	61.23	523.33		
0.15	70.08	217.85	70.05	217.77	60.72	554.15	Mean CT	T_max
0.2	70.08	221.42	70.05	217.83	60.55	661.94	60.15	696.57
0.25	70.04	221.29	70.02	217.84	60.43	658.5	Min (d)	
0.3	69.96	219.57	70.01	217.85	60.48	662.34		
0.35	69.95	219.56	70.02	217.87	60.42	651.01	Mean CT	T_max
0.4	69.92	219.56	70.02	217.87	60.38	639.32	70.46	219.67
0.45	69.79	219.5	70.01	217.88	60.33	649.84		
0.5	69.78	219.51	70.01	217.88	60.32	649.85		
0.55	69.74	219.55	70	217.88	60.23	669.51		
0.6	69.67	219.56	70	217.88	60.18	656.75		
0.65	69.67	218.97	70.01	217.88	60.11	649.2		
0.7	69.59	219.08	69.93	217.95	60.06	652.12		
0.75	69.52	221.03	69.92	217.95	60.08	666.95		
0.8	69.41	221.08	69.92	217.95	60.09	666.96		
0.85	69.33	221.17	69.96	221.49	60.13	675.06		
0.9	69.07	221.1	69.92	221.31	60.15	670.6		
0.95	68.61	219.38	69.9	221.32	60.16	670.59		

C. Output Data Tables

Table C.2: Results of Pareto control in serial processing for Mean Cycle Time and Maximum Tardiness ($\rho = 0.85$)

	CP 0.33		CP 0.66		CP 1		CP 2	
w	Mean CT	T_max	Mean CT	T_max	Mean CT	T_max	Mean CT	T_max
0.05	84.29	963.27	93.37	385.79	103.31	303.77	106.47	294.35
0.1	82.24	1241	88.14	469.23	98.42	330.99	106.25	294.7
0.15	81.6	1299.13	85.99	551.18	95.05	367.71	105.8	294.06
0.2	81.31	1389.78	84.74	677.64	92.47	401.5	105.58	294.57
0.25	80.63	1440.25	84.04	748.41	90.63	416.53	105.32	293.39
0.3	80.29	1445.63	83.47	830.34	89.2	438.43	104.85	293.81
0.35	80.13	1394.27	82.96	908.61	88.12	456.1	104.41	293.46
0.4	80.2	1519.77	82.49	941.75	87.32	477.79	104.11	299.45
0.45	80.06	1587.27	82.41	1008.84	86.63	509.78	103.52	299.15
0.5	79.95	1664.29	82.12	1047.84	86.02	538.62	103.16	300.79
0.55	80.18	1748.41	81.67	1113.29	85.26	609.84	102.62	302.08
0.6	80.13	1752.86	81.68	1133.53	84.82	635.37	101.9	306.88
0.65	80.12	1752.88	81.42	1175.45	84.36	691.74	101.1	310.41
0.7	80.11	1752.86	81.21	1267.81	83.78	728.72	100.32	320.45
0.75	80.11	1752.89	80.89	1299.99	83.43	769.17	99.52	335.47
0.8	80.11	1752.89	80.62	1311.12	83	823.28	97.86	362.56
0.85	80.11	1752.91	80.29	1367.62	82.24	908.45	96.27	380.61
0.9	80.11	1752.91	79.99	1491.42	81.85	1001.58	93.89	391.21
0.95	80.11	1752.91	79.85	1704.83	81.01	1183.74	90.82	434.08
	CP 3		CP 10		WA		Extreme Points	
w	Mean CT	T_max	Mean CT	T_max	Mean CT	T_max		
0.05	106.43	293.63	106.02	294.31	88.18	593.16	Min (p)	
0.1	106.15	294.16	105.95	294.3	85.5	770		
0.15	106.09	293.36	105.86	294.37	84.4	906.33	Mean CT	T_max
0.2	105.89	293.47	105.82	294.38	83.98	1011.82	79.94	1779.28
0.25	105.83	293.55	105.79	294.41	83.31	1083	Min (d)	
0.3	105.72	293.49	105.81	294.56	82.99	1077.07		
0.35	105.59	292.25	105.75	294.45	82.82	1207.57	Mean CT	T_max
0.4	105.49	292.27	105.75	294.46	82.36	1196.72	106.66	293.01
0.45	105.47	292.37	105.82	294.58	82.23	1190.67		
0.5	105.35	292.43	105.78	294.6	81.87	1346.97		
0.55	105.15	292.52	105.74	293.65	81.87	1318.22		
0.6	105.09	292.61	105.73	293.64	81.62	1321.21		
0.65	104.93	292.71	105.7	293.64	81.31	1347.44		
0.7	104.83	293.23	105.69	293.67	81.03	1359.85		
0.75	104.74	294.41	105.63	293.69	80.81	1378.32		
0.8	104.59	296.35	105.58	292.3	80.4	1417.97		
0.85	104.43	294.08	105.52	292.31	80.21	1529.9		
0.9	103.84	297.06	105.53	292.35	80.12	1688.63		
0.95	102.74	298.08	105.49	292.38	79.85	1748.47		

C. Output Data Tables

Table C.3: Results of Pareto control in serial processing for Mean Cycle Time and Cycle Time Variance ($\rho = 0.75$)

	CP 0.33		CP 0.66		CP 1		CP 2	
w	Mean CT	Var. CT	Mean CT	Var. CT	Mean CT	Var. CT	Mean CT	Var. CT
0.05	67.73	2178.57	67.56	2156.3	67.4	2156.8	66.74	2137
0.1	67.73	2178.57	67.49	2153.06	67.14	2135.5	66.37	2140.6
0.15	67.57	2157.99	67.37	2155.55	66.93	2139.72	66.06	2139
0.2	67.56	2159.36	67.15	2136.51	66.69	2128.16	65.86	2144.6
0.25	67.52	2158.73	66.92	2133.9	66.43	2127.59	65.71	2155.99
0.3	67.42	2159.83	66.67	2122.94	66.15	2133.28	65.54	2167
0.35	67.22	2151.17	66.33	2118.76	65.94	2141.47	65.33	2170.8
0.4	66.85	2147.24	66.01	2131.71	65.56	2131.31	65.12	2157.3
0.45	66.29	2143.48	65.58	2136.93	65.26	2123.21	65.06	2169.63
0.5	65.6	2155.31	65.16	2137.74	65.03	2152.01	64.93	2187.55
0.55	64.56	2159.05	64.69	2144.76	64.77	2159.55	64.8	2192.46
0.6	63.43	2206.29	64.28	2159.12	64.39	2175.59	64.63	2177.6
0.65	62.79	2350.5	63.68	2216.58	64.02	2172.55	64.34	2166.88
0.7	61.7	2453.84	63.1	2250.62	63.62	2212.82	64.14	2159.27
0.75	60.91	2659.31	62.72	2334.05	63.09	2211.19	63.97	2166.88
0.8	60.05	2757.78	62.14	2376.61	62.72	2228.85	63.81	2207.39
0.85	59.13	2754.37	61.32	2462.33	62.39	2321.44	63.44	2213.31
0.9	58.75	2894.75	60.66	2597.02	61.87	2412.46	62.96	2248.09
0.95	58.72	2978.18	59.65	2719.02	60.87	2522.52	62.46	2267.75
	CP 3		CP 10		WA		Extreme Points	
w	Mean CT	Var. CT	Mean CT	Var. CT	Mean CT	Var. CT		
0.05	66.27	2148.43	65.35	2188.5	66.93	2132.94	Min (p)	
0.1	65.9	2151.03	65.27	2201.97	66.4	2123.09		
0.15	65.76	2175.96	65.23	2208.96	66.02	2120.36	Mean CT	Var. CT
0.2	65.56	2163.49	65.18	2207.15	65.56	2120.6	58.68	3004.58
0.25	65.39	2158.27	65.07	2201.8	65.27	2129.21	Min (p/w)	
0.3	65.24	2167.27	65.05	2198.96	64.97	2156.63		
0.35	65.23	2194.24	65.02	2199.23	64.66	2171.62	Mean CT	Var. CT
0.4	65.16	2203.21	64.95	2191.34	64.28	2152.15	67.73	2178.57
0.45	65.05	2189.24	64.98	2206.94	63.9	2197.43		
0.5	64.92	2196.74	64.93	2205.74	63.51	2220.52		
0.55	64.87	2196.1	64.93	2210.72	62.97	2231.99		
0.6	64.68	2172.02	64.92	2213.18	62.5	2262.96		
0.65	64.56	2173.58	64.82	2195.1	61.85	2336.71		
0.7	64.39	2176.39	64.81	2198.14	61.29	2381.11		
0.75	64.2	2160.71	64.78	2201.45	60.81	2449.41		
0.8	64.18	2213.89	64.69	2194.34	60.23	2547.15		
0.85	63.97	2205.25	64.69	2220.49	60.03	2665.97		
0.9	63.77	2240.47	64.63	2225.87	59.49	2685.1		
0.95	63.16	2229.19	64.46	2239.34	58.91	2766.71		

C. Output Data Tables

Table C.4: Results of Pareto control in serial processing for Mean Cycle Time and Cycle Time Variance ($\rho = 0.85$)

	CP 0.33		CP 0.66		CP 1		CP 2	
w	Mean CT	Var. CT	Mean CT	Var. CT	Mean CT	Var. CT	Mean CT	Var. CT
0.05	102.83	6682.58	102.78	6682.15	102.23	6677.92	100.45	6827.63
0.1	102.88	6683.73	102.6	6700.03	101.76	6720.31	99.66	6954.86
0.15	102.85	6686.58	102.17	6685.15	101.18	6775.05	99.1	7011.14
0.2	102.81	6692.59	101.81	6695.21	100.55	6810.21	98.64	7083.33
0.25	102.68	6694.93	101.33	6739.18	99.86	6841.91	98.2	7130.35
0.3	102.31	6695.6	100.79	6781.72	99.28	6885.08	97.62	7139.82
0.35	101.88	6716.35	99.98	6809.3	98.7	6948.37	97.29	7236.27
0.4	101.21	6748.89	99.14	6858.4	98.05	6993.37	96.79	7287.3
0.45	100.07	6794.76	98.19	6928.33	97.27	7088.36	96.43	7368.35
0.5	98.45	6952.84	97.21	7031.85	96.47	7107.28	96.09	7474.73
0.55	96.25	7183.64	96.2	7134.48	95.75	7213.96	95.84	7552.82
0.6	93.9	7492.64	94.86	7300.73	94.97	7415.42	95.49	7629.13
0.65	91.12	8248.1	93.31	7527.79	94.11	7583.37	94.92	7720.8
0.7	88.77	9079.35	92.21	7927.75	93.14	7877.32	94.37	7818.22
0.75	86.25	10218.65	90.95	8424.27	92.32	8191.19	93.91	7957.36
0.8	83.86	12251.45	89.32	9257.68	91.19	8531.71	93.62	8308.93
0.85	81.86	14796.26	87.48	10204.4	89.77	9116.24	92.63	8430.18
0.9	79.75	16554.21	85.71	11693.35	88.63	10180.65	91.59	8867.8
0.95	78.43	21972.13	83.08	14868.37	86.41	11809.07	90.54	9657.05
	CP 3		CP 10		WA		Extreme Points	
w	Mean CT	Var. CT	Mean CT	Var. CT	Mean CT	Var. CT		
0.05	99.38	7010.25	97.08	7330.16	100.54	6856.27	Min (p)	
0.1	98.59	7080.5	96.84	7380.45	98.7	7063.04		
0.15	98.11	7123.13	96.7	7445.26	97.45	7272.94	Mean CT	Var. CT
0.2	97.86	7192.74	96.57	7504.56	96.43	7554.8	78.67	29046.62
0.25	97.45	7261.97	96.52	7542.57	95.71	7922.87	Min (p/w)	
0.3	97.16	7326.15	96.36	7531.41	94.71	8007.76		
0.35	96.85	7373.36	96.28	7568.57	93.86	8576.05	Mean CT	Var. CT
0.4	96.58	7454.75	96.3	7610.88	93.01	8881.31	102.83	6682.25
0.45	96.32	7495.03	96.12	7622.46	91.87	9078.78		
0.5	95.99	7520.64	96.01	7650.15	91.17	9452.59		
0.55	95.87	7629.79	95.93	7669.71	89.92	9778.94		
0.6	95.53	7658.14	95.89	7659.49	88.98	10801.9		
0.65	95.34	7757.76	95.93	7745.21	87.73	11492.46		
0.7	95.02	7817.31	95.89	7794.51	86.08	12142.26		
0.75	94.5	7869.7	95.71	7766.57	84.59	13914.97		
0.8	94.13	8074.69	95.58	7797.29	83.84	16312.52		
0.85	93.89	8294.38	95.47	7861.92	83.29	20224.09		
0.9	93.37	8662.54	95.18	7871.05	81.33	23220.38		
0.95	92.13	8988.5	94.82	7989.9	79.84	27650.81		