ONLINE DUE DATE ASSIGNMENT IN DYNAMIC AND STOCHASTIC SCHEDULING ENVIRONMENTS WITH FAMILY SETUPS UNDER TARDINESS AND QUOTED LEAD TIME CONSIDERATIONS

by

Zehra Düzgit

B.S., Systems (Industrial) Engineering, Yeditepe University, 2003

M.S., Engineering and Technology Management, Boğaziçi University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Department of Industrial Engineering

Boğaziçi University

2015

ACKNOWLEDGEMENTS

This PhD dissertation is the result of a challenging journey, upon which many people have given their support. Undertaking this PhD has been a truly life-changing experience for me. Without the support of the following people, this study would not have been completed.

I would like to express my deepest gratitude and thanks to my advisor Ali Tamer Ünal for all the support, guidance and encouragement. He has been a tremendous mentor for me helping to shape up my vision. His wisdom, knowledge and expertise inspired and motivated me. I appreciate all his generous contributions of time and ideas. This dissertation would not have been possible without his supervision and contribution. I am grateful to my supervisor for always keeping his doors wide open. It was a great pleasure working with him.

I would also like to thank my committee members for their guidance and support. I am very grateful to Barış Selçuk and Taner Bilgiç for their time, interest, brilliant suggestions and providing insightful discussions and helpful comments about the research. I would also like to thank the other two members of my defense committee, Zeki Caner Taşkın and Ayhan Özgür Toy. I am indebted to Zeki Caner Taşkın for all his support, interest and generously sharing servers. My profound thanks to Ayhan Özgür Toy for his endless motivation, interest, continuous support, guidance and believing in me.

I would also like to thank my friends and colleagues Ayşenur, Cemre, Hande, Sedef, Haluk, Ufuk and Korhan for their motivation and companionship. A special thanks to my life-long friend Gizem Bacaksızlar for her sincere motivation, interest and support.

Last but foremost, I would also like to say a heartfelt thank you to my family for always believing in me and encouraging me to follow my dreams. Words cannot express how grateful I am to my mother Yurdanur, my father Ercan, my brother Mustafa and my aunt Aynur who supported and helped me at every stage of my personal and academic life. I owe a lot to my family for their sacrifice, unconditional love, incredible support, understanding, motivation and patience during this challenging and long-lasting period; especially my brother for being the best friend and the source of my life. I am proud of them and love them very much for brightening my life. This study would not have been completed without them.

ABSTRACT

ONLINE DUE DATE ASSIGNMENT IN DYNAMIC AND STOCHASTIC SCHEDULING ENVIRONMENTS WITH FAMILY SETUPS UNDER TARDINESS AND QUOTED LEAD TIME CONSIDERATIONS

In this study, due date assignment problem is addressed in a single machine dynamic and stochastic environment with family setups. A due date is to be assigned for each new job immediately at the time of its arrival. While assigning a due date for a new arrival, non-complete jobs that have arrived before that job and whose due dates are already assigned, are also considered and rescheduled. While assigning a short due date for a new arrival as close as possible, compliance to assigned due dates of non-complete jobs is necessary. These two objectives conflict with each other. In order to solve this problem, a two-phase solution methodology is proposed. In the first phase, a capacity allocation takes place for families before observing any actual jobs arrivals, based on expected work load and arrival estimation, in a periodic and static manner. In this phase, families can be assigned to batches and a batching structure is formed. In the second phase, a due date is assigned for the new arrival immediately in an online fashion, based on the outputs of the first phase. A mixed integer programming model and a heuristic algorithm are developed for each phase. Simultaneously, a discrete event simulation is carried out to imitate a real production system. The performance of the designed batching policy is measured through the developed simulation model and results are reported under different system parameters.

ÖZET

DİNAMİK VE STOKASTİK ORTAMLARDA GECİKME VE TESLİMAT SÜRESİ ETKİLERİ ALTINDA AİLE HAZIRLIK ZAMANLI TERMİN ATAMA

Bu çalışmada tek makinalı dinamik ortamda rassal olarak gelen ve farklı ailelere ait işlere termin tarihi atama problemi ele alınmıştır. Her yeni gelen işe geldiği anda hemen bir termin tarihi atanmalıdır. Yeni gelen bir işe termin tarihi atanırken, bu işten önce gelmiş ve termin tarihleri çoktan atanmış olan işler de dikkate alınmalı ve yeniden çizelgelenmelidirler. Yeni gelen işlere olduğunca kısa termin süresi atarken termin tarihleri atanmış işlerin atanan termin tarihlerine uyumlarını sağlamak gerekmektedir. Bu iki amaç birbiriyle çelişmektedir. Bu problemin çözümü için iki aşamalı bir çözüm yöntemi sunulmuştur. Birinci aşamada, henüz hiç iş gelmeden, beklenen iş yükü ve geliş zamanlarına dayanarak periyodik ve statik bir şekilde aileler için kapasite ataması yapılır. Bu aşamada aileler için parti ataması yapılabilir ve bir parti yapısı oluşturulur. İkinci aşamada, birinci aşamanın sonuçlarına dayanarak, yeni gelen işe hemen bir termin tarihi atanır. Her iki aşama için de bir karışık tamsayı programlama modeli ve bir sezgisel algoritma geliştirilmiştir. Tasarlanan parti üretim politikası, geliştirilen benzetim modeli vasıtasıyla test edilmiş ve farklı sistem parametreleri çerçevesinde performans raporlanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	Allowance factor
a_j	Arrival time of job j
$\mathcal{A}_f$	Mean of interarrival time distribution of family f
b	Batch position index
B	Maximum number of batches to be opened within two scheduling periods in batch allocation phase
BB	Total number of opened batches within two scheduling periods in batch allocation phase
c_j	Completion time of job j
C_{kb}	Completion time of the job on position k in batch position b
CT_b	Completion time of batch on position b
CT'_b	Completion time of opened batch on position b
d_j	Due date of job j
e_j	Random number for job j in RND due date assignment method
f	Family index
$\bar{f}$	Mean flow allowance for performance comparison
F	Set of families
FT_j	Flow time of job j
$\mathcal{F}(b)$	Family assigned to batch on position b
$\mathcal{F}(j)$	Family of job j
$g_j(a_j, d_j)$	Function regarding job j's competitive power
G_{fb}	Binary decision variable denoting whether family f is assigned to batch on position b which is completed in second scheduling period, or not
$h_j(c_j, d_j)$	Function regarding job j's late delivery
H_b	Binary decision variable denoting whether batch b is completed in second scheduling period or not
i	New arriving job
j	Job index

j^p	Job currently being processing on the machine
J	Set of jobs
J_t^C	Set of completed jobs by time t
J_t^{CF}	Set of jobs waiting to be processed from the current family being processed before time t, where $J_t^{CF} \subseteq J_t^{NC}$
J_t^{NC}	Set of non-complete jobs by time t
J_t^{NF}	Set of jobs waiting to be processed from the next family for which a setup is planned at time t, where $J_t^{NF} \subseteq J_t^{NC}$
J_t^S	Set of scheduled jobs before and on time t, where $J_t^S = J_t^{NC} \cup \{i\}$
$J_{T_{begin}}^{NC}$	Set of non-complete jobs by time T_{begin}
$\mid J_t^C \mid$	Number of jobs completed by time t
$\mid J_t^{CF} \mid$	Number of jobs waiting to be processed from the current family being processed before time t
$\mid J_{T_{begin}}^{NC} \mid$	Number of non-complete jobs by time T_{begin}
$\mid J_t^{NF} \mid$	Number of jobs waiting to be processed from the next family for which a setup is planned at time t
k	Job position index within each batch
K	Maximum number of positions within each batch
L_j	Lateness of job j
M_1	Sufficiently large positive number in optimization models, equals $EST + 2 \cdot SP$
M_2	Sufficiently large positive number in optimization models, equals $2 \cdot SP$
n	Number of opened batches for a family in batch allocation phase
$O(x)$	Running time of an algorithm in Big O notation in computational complexity theory where x refers to the size of the problem
p_j	Processing time of job j
$\bar{p}$	Mean processing time for performance comparison
$\mathcal{P}_f$	Mean of processing time distribution of family f

PT_b	Expected total processing time requirement for new arrivals assigned to batch on position b
Q_j	Quoted lead time of job j
QLT_{fbn}	Expected quoted lead time for family f assigned to batch b with a total of n opened batches
RCT_b	Rescheduled completion time of batch on position b
$\mathcal{S}_f$	Mean of setup time distribution of family f
ST_b	Start time of batch on position b
ST'_b	Start time of opened batch on position b
t	Time instance
T	Scheduling period index
T_{begin}	Beginning of scheduling period
T_{end}	End of scheduling period
$TARD_j$	Tardiness of job j
U_{fn}	Binary decision variable denoting whether there are n opened batches for family f or not
w_j	Weight (priority) of job j
W_{fb}	Total work content requirement of family f assigned to batch on position b
X_{jkb}	Binary decision variable denoting whether job j is assigned to position k in batch on position b or not
Y_{fb}	Binary decision variable denoting whether family f is assigned to batch on position b or not
Z_0	Value of initial objective function
Z_0^*	Optimum value of initial objective function
Z_1	Value of first major objective function
Z_1^*	Optimum value of first major objective function
Z_2	Value of second major objective function
α	Due date tightness parameter in TWK and PPW due date assignment methods
β	Due date tightness parameter in SLK and PPW due date assignment methods

γ	Due date tightness parameter in CON due date assignment method
δ_j	Flow allowance of job j
Δ^I_{fb}	Extra amount of capacity allocated to new arrivals from family f assigned to batch on location b
Δ^{II}_f	Extra amount of capacity allocated to new arrivals from family f
Δ^{III}	Extra amount of capacity allocated to new arrivals from all families on all locations
ϵ	Slack within two scheduling periods in batch allocation phase
θ_f	Probability of belonging to family f
κ	Due date tightness parameter in RND due date assignment method
λ	Job arrival rate
μ	Mean job processing time
π	Setup time multiplier
ρ	Machine utilization rate
ρ_{eff}	Effective machine utilization parameter
ρ^{low}_{eff}	Effective machine utilization parameter for low utilization
ρ^{high}_{eff}	Effective machine utilization parameter for high utilization
τ	Mean job interarrival time
υ	Mean family setup time
ϕ	Scheduling period multiplier
ψ	Setup time variability parameter
ω	Processing time variability parameter

LIST OF ACRONYMS/ABBREVIATIONS

AFT	Average Flow Time
AQLT	Average Quoted Lead Time
ATC	Apparent Tardiness Cost
ATCS	Apparent Tardiness Cost with Setups
ATRDY	Average Tardiness
CDF	Cumulative Distribution Function
CON	Common (constrained) due date assignment method
CR	Critical Ratio
DIF	Unrestricted due date assignment method
DSL	Decreasing Service Level
EDD	Earliest Due Date
EFT	Earliest Finish Time
ERD	Earliest Release Date
ES	End of Simulation
EST	Earliest Start Time
FCFS	First Come First Served
FEL	Future Event List
FIFO	First In First Out
G/G/1	Single server queuing system where arrivals and service times follow a general distribution
JA	Job Arrival
JC	Job Completion
JIT	Just In Time
KPI	Key Performance Indicators
LEPT	Largest Expected Processing Time
LPT	Longest Processing Time
MDD	Modified Due Date
M/G/1	Single server queuing system with Poisson arrivals where service times follow a general distribution

MIP	Mixed Integer Programming
M/M/1	Single server queuing system with Poisson arrivals and Exponential service times
MS	Minimum Slack
MTO	Make To Order
MTS	Make To Stock
NP-hard	Non-deterministic Polynomial-time hard
NR	Number of Replications
NTJ	Number of Tardy Jobs
PPW	Process Plus Wait due date assignment method
RND	Random due date assignment method
SC	Setup Completion
SLK	Slack due date assignment method
SNS	Start of Next Setup
SOH	Start of Horizon
SP	Length of Scheduling Period
SPT	Shortest Processing Time
SPTA	Shortest Processing Time among Available jobs
SSC	Statistical Service Control
SST	Shortest Setup Time
TC	Tardiness Coefficient
TPT	Total Processing Time requirement after warm-up period
TST	Total Setup Time requirement after warm-up period
UCB	Update Current Batch
TWK	Total WorK content due date assignment method
WP	Warm-up Period
WSEPT	Weighted Shortest Expected Processing Time
WSPT	Weighted Shortest Processing Time
ZI	Zero Inventory

1. INTRODUCTION

Due date is the date a job or an order is promised to be delivered to customers. Quoted lead time is the difference between the assigned due date and arrival time of a job. Due date assignment and lead time quotation have been receiving considerable attention in recent years with the transition from make-to-stock (MTS) companies to make-to-order (MTO) ones which is a result of the paradigm shift from mass production to customized production.

Especially, due to the introduction of inventory management systems such as Zero Inventory (ZI) and the production philosophies such as lean manufacturing or Just-in-Time (JIT) systems, as the name implies, jobs are to be completed on time; neither early nor late. When a job is completed before the promised due date and the company cannot deliver the job, the company may encounter earliness penalties such as inventory carrying and storage costs, product deterioration cost and tied-up capital or opportunity cost, according to [1]. This situation also signifies ineffective usage of resources. When a job is completed and delivered after the promised due date, the company may encounter tardiness penalties such as monetary penalties for late delivery, customer dissatisfaction, loss of customer goodwill, possibility of losing customers, damaged reputation and potential loss of future business or lost market share in the long run, according to [2]. Therefore, providing products and services on time, is a critical factor as well as cost and quality.

Within the framework of due date assignment, in a production system at any time instance, there can be a set of jobs whose due dates are assigned and already completed, a set of scheduled non-complete jobs whose due dates are assigned but not completed and a set of jobs whose due dates are not assigned yet. While assigning due dates, depending on the closeness to job arrival times, due dates can be set as tight or loose. However, there is a trade-off for companies between assigning tight and loose due dates, equivalently short and long quoted lead times, respectively.

Quoting relatively longer due dates gives additional production flexibility to manufacturers in order to assure delivery reliability. Nevertheless, assigning longer due dates causes loss of competitive power with respect to responsiveness to customer demand, particularly in highly competitive markets. Especially, within the scope of supply chain, quoting long lead times may cause large downstream stock levels in MTS systems and it may lead to customer dissatisfaction in MTO organizations, according to [3]. Hence, keeping delivery dates too far may be unacceptable for customers and they may even cancel their orders and search for a competitive firm which promises earlier delivery.

On the other hand, setting shorter due dates is more attractive in terms of retaining customers but orders may not be delivered on time if the promised due dates leave a very short production lead time. In that sense, tardiness is an indicator of the compliance to the assigned due dates. Thus, firms must quote shorter due dates than their competitors' in competitive markets but at the same time, the quoted due dates must be reliable and attainable. As a result, it is very important for companies to promise realistic delivery dates and due date setting plays an important role in scheduling and supply chain management.

Within the scope of this study, due date assignment problem is considered for batch processing companies. The important attribute of batch processing companies is that they have product family structures and family setups. A significant setup time is required before switching to another product family. Metal, automotive supplier, glass, paper and textile manufacturers are examples of such companies.

In order to assign due dates, in literature, one of the frequently used methods is to add some flow allowance in addition to arrival time of jobs. During the determination of flow allowances, some take job related data (such as processing times) into account and some consider shop related information (such as work congestion) whereas others either consider or ignore both types of data. Most of these flow time estimation methods are parametric and the parameters are tuned usually via simulation experiments. These methods work in general, however, they do not involve family setups.

Under setup considerations, in practice there are two due date approaches used in industry. The first one is the periodic due date assignment. In this type of due date assignment strategy, planning horizon is divided into fixed scheduling periods and all jobs that arrive in between two scheduling periods are gathered. Due dates of these jobs are quoted at the beginning of the next scheduling period under the light of the characteristics (family types and processing time requirements) that reveal at their arrival times. The advantage of this method is to be able to assign due dates based on known job related data. This type of an assignment enables to share a single setup for the jobs that belong to the same family and to process them consecutively within the single batch allocated for their family. Due to batching, operational efficiency is the major gain of this method. The downside of this method is that customers have to wait to get a due date in addition to the flow time of jobs, in other words the time spent in the system. This method may be acceptable for monopolistic structures in low competitive markets where the company has full control on its customers and usually produces in large volume of batches. But it is not a valid procedure for every market, especially in highly competitive markets where companies usually produce in small volumes.

As an alternative method, due dates can be assigned at the time of job arrivals instantly. This policy is known as immediate due date assignment. By this way, the waiting time to get a due date is removed for customers. But this time, since job related data is not known in advance at the time of due date quotation, gaining operational efficiency through batching cannot be possible in a straightforward way from the point of manufacturer, except first-come first-served (FCFS) dispatching. However, under such circumstances, straightly assigning a new arrival to the first available position in schedule is not reasonable in terms of family setups since a new setup will be required every time before processing a job from a different family and it will lead to excessive number of setups in the long run.

In this study, a due date is to be assigned for a new arriving job immediately at the time of its arrival based on its expected completion time under two conflicting objectives. These are keeping quoted due dates of new arrivals as tight as possible to attract customers and to gain or sustain competitive power while avoiding tardiness of non-complete jobs as much as possible. The goal is to gain operational efficiency in spite of uncertainty under these two conflicting objectives through an immediate due date assignment strategy.

To the best of our knowledge, there is no study that combines dynamic and stochastic environment with family setups so as to minimize quoted lead time for new arrivals and tardiness of non-complete jobs, simultaneously. Therefore, setting due dates in dynamic and stochastic environment together with family setups under these considerations is an interesting and fertile area for research in order to fill part of this gap which constitutes the motivation behind this study.

The production environment under consideration can be defined as follows. There is a single machine that can process one job at a time. The machine is not subject to any breakdown or maintenance. It is assumed that each job is composed of a single operation. Preemption is not allowed. Every job belongs to a product family. A sequence-independent setup time is required before processing a job from a different family where setup time depends only on the current family to be processed, independent of the immediately preceding family. The environment is dynamic, jobs are not readily available at the beginning and arrive over time. Moreover, job related data, namely interarrival times, processing times and family types are stochastic variables.

Dynamic arrivals represent a more realistic environment since all jobs to be processed may not be simultaneously available at the beginning of scheduling process in most of the real life cases. The rationale behind selecting a single machine environment is the possibility to extend its results to apply in multi machine environments.

As a solution approach, a two-phase methodology is proposed. In the first phase, the process starts with dividing the planning horizon into equidistant scheduling periods. As a proactive plan for operational efficiency, a capacity allocation for families takes place based on expected workload and arrival estimation, before observing actual job arrivals. This forward-looking family allocation is managed in an offline and periodic manner. However, before passing to the next phase, non-complete jobs with assigned due dates from previous periods must be taken into consideration and rescheduled according to the constructed family allocation and their tardiness must be tracked. This phase is designed in such a way that, each family can be allocated to more than one batch by allowing family splitting. This first phase is called batch allocation.

The motivation behind applying a batching structure is to take the advantage of sharing a single setup and processing a group of expected future arrivals from the same family consecutively, instead of performing a setup every time before processing a job from a different family. The number and sequence of formed batches, batch sizes and family-batch allocations are to be determined in the batch allocation phase.

The second phase is the due date assignment phase. Due date quotation takes place as soon as a new job arrives in an online fashion, by assigning the job into the most appropriate position, based on the family-batch configuration obtained in the first phase. Hence, the first phase serves as a pre-arrangement with respect to families for the second phase.

For each phase, a mixed integer programming (MIP) formulation and a heuristic algorithm are constructed. There are two objectives in the mathematical model developed for batch allocation phase: to minimize quoted lead time for expected new arrivals and to minimize tardiness of non-complete jobs. Likewise, there are two objectives in the mathematical model constructed for due date assignment phase: to minimize the due date to be assigned for a new arrival and to minimize tardiness of non-complete jobs. Two mathematical models are quite interlinked to each other such that the outputs of the first phase become inputs for the second phase. The optimization model for the first phase is solved at the beginning of each scheduling period whereas the optimization model in the second phase is solved every time a new job arrives.

While loading batches into the schedule in the batch allocation phase, large batches have the advantage of less number of setups and high machine utilization. On the other hand, producing in small batches causes more number of setups but shorter due dates can be assigned accordingly.

While assigning a due date for a new arrival in the due date assignment phase, there is a trade-off to be considered. The challenge is to integrate every new arriving job into an already existing batch. The difficulty arises from the consideration of conflicting objectives with respect to the new arrival and non-complete jobs present in the system. A closer due date can be quoted for the new arrival by assigning it to the left hand side of non-complete jobs in the schedule, by shifting the completion times of non-complete jobs to the right hand side, at the expense of causing or increasing their tardiness. On the other hand, a further due date can be assigned for the new arrival by assigning it to the right hand side of non-complete jobs, in order not to violate the promised due dates and to prevent tardiness of non-complete jobs.

In order to experiment and measure the performance of the proposed solution methodology, a simulation environment is constructed to mimic a real production system. Every time when batch allocation problem is solved, the current situation of the simulation is taken into consideration to reflect the current circumstance of the virtual single machine shop, through a mapping between the proposed solution (plans) and simulation (realizations).

Different dispatching policies are designed to represent different sequencing schemes for simulation, which take the outputs of the batch allocation as their guideline. Performance measures in terms of average quoted lead time, average tardiness, average flow time, number of setups and machine utilization are analyzed and reported under different production environments.

The study is organized as follows. In Chapter 2, a literature survey is given regarding due date related problems, mostly in single machine environments. Problem definition is introduced in Chapter 3 which is followed by the proposed solution methodology in Chapter 4. In Chapter 5, MIP models are presented. Heuristic algorithms take place in Chapter 6. Dispatching policies are given in Chapter 7. In Chapter 8, the framework of the experimental design is described. Computational tests are reported in Chapter 9. Chapter 10 contains the performance comparison. Conclusions are drawn in the last chapter.

2. LITERATURE REVIEW

There is a noteworthy amount of research regarding due date assignment and lead time quotation problems. Problems can be classified as static and dynamic depending on the nature of job arrivals. In static problems, all jobs are readily available at the beginning and the number of jobs to be processed is known in advance whereas in dynamic problems, jobs may arrive over time and job characteristics become known when jobs arrive.

Problems can be further classified as deterministic and stochastic, based on job characteristics. In deterministic problems, job related data (such as arrival times, processing times) are known with certainty, while in stochastic problems, it is generally assumed that job related data follows a probability distribution.

One of the pioneering studies in the area of due date assignment problems is [4]. A single machine problem is considered where all jobs have a common but unknown due date. The objective is to find the optimal value of this common due date and the optimal sequence of jobs in order to minimize a penalty function involving due date assignment, earliness and tardiness costs. A polynomial time algorithm is proposed for the problem. [5] is another basic paper which deals with due date assignment problem. The environment and objective function are the same as in [4]. The distinction is that, a different due date can be assigned for each job. A scheduling procedure is proposed to determine optimal due dates and processing sequence. These two studies triggered considerable research in the domain of due date assignment.

[4] and [5] are the first examples of static and deterministic due date assignment problems. Similarly, [6] examines single machine simultaneous due date assignment and scheduling problem in static and deterministic environments. A polynomial time algorithm is suggested for determination of optimal due dates and optimal sequence of jobs in order to minimize earliness, tardiness and due date assignment penalties. [7] proposes a linear programming model to find an optimal common due date for the

same environment and objective function with [6]. Duality theory is used to solve the problem optimally. However, [8] states that [7] ignores the scheduling aspect of the problem and through the same examples given in [7], it is shown that it is possible to find lower objective function values based on a certain sequencing principle.

[9] suggests an $O(n^2)$ polynomial time dynamic programming algorithm for static and deterministic single machine due date assignment problem where n is the number of jobs. The objective is to minimize due date assignment cost and weighted number of tardy jobs, for which an $O(n^4)$ polynomial time algorithm was already proposed by [10]. [11] shows that the dynamic algorithm proposed by [9] can be applied to due date assignment problems under the effects of learning and sequence-dependent setup times in order to minimize the weighted number of tardy jobs, at no additional computational cost.

[12] studies static and deterministic single machine due date assignment problem for the minimization of two distinct objective functions: one is a cost function involving earliness, tardiness and due date assignment penalties and the other is a function containing penalties due to number of tardy jobs and due date assignments. It is shown that, if linear type of penalty functions are replaced with more general type of penalty functions, more robust optimization algorithms can be obtained with no additional computational effort. [13] examines single machine due date assignment problem where due dates are linear functions of job waiting times and the objective is to minimize the maximum lateness. An optimal sequence is determined to obtain optimal due dates.

[14] analyzes static deterministic single and identical parallel machine due date assignment problems with equal-size jobs where all jobs require the same processing time. The objective is to minimize total weighted earliness, tardiness and due date assignment cost. An optimal $O(n^3)$ polynomial time algorithm is suggested for the single machine case for which an $O(n^4)$ polynomial time algorithm was already presented by [15].

[16] integrates due date assignment and scheduling problems in a job-shop environment. A double layered optimization model is proposed to solve the combined problem so as to minimize total weighted tardiness. The set of unscheduled jobs waiting to be scheduled at the beginning is divided into two subsets based on their due date quotation status as the jobs whose due dates are already assigned (reflecting the orders from important customers and/or carrying strategic importance for the company whose due dates cannot be modified once set based on customers' preferences and/or marketing department) and the remaining set of jobs whose due dates are to be specified. In the first layer, the due dates for the unassigned jobs are roughly determined whereas in the second layer, the solution's quality is aimed to be improved by means of perturbing the due dates.

[17] considers a due-window concept instead of considering an upper bound on the acceptable lead time. There is an acceptable time interval for lead times and due dates assigned out of this interval are assumed to be tardy or early due dates, depending on the upper and lower bound of the intervals, respectively. The objective is to minimize the maximum cost among all the jobs where the cost function is composed of job earliness, job tardiness, due date earliness and due date tardiness costs. A polynomial time solution is presented to solve the minmax problem.

Computational complexities of static and deterministic due date assignment problems are investigated in the following studies. [18] analyzes single machine scheduling problems where due dates are decision variables. Computational complexity results are examined for a variety of due date related objectives. [19] and [20] explore various scheduling environments including single machine cases. Maximum tardiness and due date assignment cost are considered as the scheduling criteria to be minimized in [19] whereas a cost function including earliness, tardiness and due date assignment is minimized in [20]. The computational complexity of the problems are stated by proving that they are NP-hard for most single (and multi) machine environments. Polynomial time solutions are presented for some special cases in single machine setting.

Examples of static and stochastic single machine problems are as follows. [21] considers static stochastic problem on a single machine where the objective is to minimize total weighted deviations of job completion times from a common due date. Processing times and the due date are exponentially distributed random variables and each job has an associated weight. Job weights are proportional to processing times, with the claim that, a job with longer processing time always has a higher selling price and consequently higher priority. It is shown that largest expected processing time (LEPT) job sequence solves the problem optimally.

[22] investigates static stochastic single machine scheduling problem with exponentially distributed processing times and due dates in order to minimize expected maximum lateness. A deterministic equivalent of the problem is derived and its results are extended for the stochastic case. A dynamic programming algorithm is suggested to solve the problem. Also heuristic policies are suggested to find near-optimal solutions.

[23] examines static stochastic single machine due date setting problem where there is no idle time between jobs and preemption is not allowed. The goal is to minimize total expected earliness and tardiness penalties. Two due date setting procedures are developed. One of them is shown to be asymptotically optimal under certain conditions while the other is not. Moreover, under normal distribution, sufficient optimality conditions and precedence relationships, that the optimal sequence must obey, are also provided.

[24] studies a static and stochastic single machine sequencing and due date assignment problems in order to minimize the expected total earliness and tardiness cost where due dates are considered as decision variables and processing times are random variables. An analytical model is constructed for the determination of optimal due dates. Two heuristics are proposed to determine the optimal sequence due to the combinatorial nature of the sequencing problem.

[25] addresses due date assignment problem in a job-shop environment under both deterministic and stochastic settings by considering the trade-off between short lead times and missed due dates. A method is proposed for accurately estimating job

flow times which is followed by a cost-based model constructed for due date quotation, based on the proposed flow time estimation method.

In some sources, which explore static and stochastic due date assignment problems, learning effect and deterioration are taken into consideration. Under learning effect, the processing time of a job gets shorter which can be related with the experience gained by an operator over time. Under deterioration effect, the processing time of a job gets longer which can be associated with the fatigue of an operator or wearing down of tools of a machine over time.

[26] extends the problem introduced in [4] by considering the effects of learning and deterioration, simultaneously. In [4], it is shown that the problem can be solved in $O(n\ log\ n)$ polynomial time. [26] proves that the problem still remains polynomially solvable in $O(n\ log\ n)$ polynomial time by the proposed algorithm under the new setting.

[27] considers learning and deterioration effects for the determination of optimal job sequence and due dates in a static stochastic single machine environment. The objective is to minimize earliness, tardiness, due date assignment costs and makespan. A unified optimization algorithm is proposed to solve the problem under different due date assignment methods.

[28] takes only the deterioration effect into account for single machine simultaneous due date assignment and scheduling problem in static and stochastic setting. A common due date is to be assigned for all jobs in order to minimize the sum of earliness, tardiness and due date assignment penalties. A simpler polynomial time algorithm is proposed to solve the problem for which an $O(n\ log\ n)$ time algorithm was already presented by [29]. [30] addresses static due date assignment and sequencing problem simultaneously with deteriorating jobs on a single machine so as to minimize weighted number of tardy jobs, earliness and due date assignment penalties. Two different polynomial time algorithms are proposed.

[31] examines static and stochastic single machine scheduling problem with simultaneous due date assignment decisions under position-dependent learning effects. Past-sequence-dependent delivery times are considered where the delivery time of a job is proportional to the job's waiting time. Four different versions of the problem are analyzed which differ with regards to the objective function and due date assignment method. The objective functions include total earliness, total tardiness and weighted number of tardy jobs components. The properties of the optimal schedules for each are stated and polynomial time algorithms are proposed.

[32] extends the problem introduced in [5] by considering processing times as random variables and switching to stochastic setting, with a known mean and variance but no knowledge of the entire distribution. A heuristic algorithm is constructed by utilizing the variances of job processing times in addition to the means of job processing times, in order to find the optimal due dates and sequence to minimize earliness, tardiness and due date assignment penalties in a single machine shop.

[33] focuses on the single machine due date assignment and scheduling problem where processing time of a job depends on its position in a processing sequence. Manufacturer has an option of discarding some of the orders if they cannot be completed on time. The objective is to minimize the cost of changing due dates, total cost of discarded jobs and possible total earliness of scheduled jobs. [34] also analyzes stochastic single machine scheduling and due date assignment problem with position-dependent processing times. [34] extends the study of [33] by considering positional weighted earliness penalties in the objective function and it is shown that the problem yet remains polynomial time solvable based on the new objective function.

[35] studies single machine scheduling and due date assignment problem under two different functions, where a job's processing time is a linear and convex decreasing function of the amount of resource allocated to the processing of the job. The goal is to minimize a cost function that includes the costs of earliness, tardiness, due date assignment, makespan and resource consumption. A polynomial time algorithm is developed to obtain the optimal job sequence, due dates and resource allocation.

[36] considers static stochastic single machine scheduling problem where job processing times are assumed to be normally distributed. The objective function to be minimized is the expected number of jobs finished after a given deterministic common due date, equivalently expected number of tardy jobs. A mathematical programming model is constructed for the problem. Then, the stochastic optimization problem is transformed into equivalent non-linear deterministic integer programming models to obtain optimal or approximate solutions.

[37] examines single machine static stochastic scheduling problem with exponentially distributed processing times. A general stochastic cost function, that includes a random due date and weight, is to be optimized and an optimal job sequence is to be determined. The distribution of the due dates is not restricted to a particular distribution. It is proven that, under certain conditions, a sequence with the weighted shortest expected processing time (WSEPT) first structure is optimal.

[38] provides a survey about papers concerning due date assignment problems under special conditions on job processing such as precedence constraints, controllable processing times based on resource allocation, deterioration and learning effects and maintenance activity.

From a different point of view, [39] investigates single machine static stochastic problem under safe scheduling concept where processing times are random variables with given probability distributions and due dates are decision variables. While trying to minimize the sum of assigned due dates and expected tardiness cost (with and without weighting factors), service level constraints are taken into consideration within the framework of safe scheduling approach. Heuristic algorithms are proposed based on sorting procedures to compute optimal due dates. [40] gives an overview of stochastic versions of single machine scheduling problems involving due dates within the scope of safe scheduling notion.

[41] considers simultaneous due date assignment and scheduling problem in a static and stochastic single machine environment to minimize the maximum due date

subject to no tardy jobs. It is shown that sequencing the jobs in decreasing service level (DSL) order solves the problem optimally.

When dynamic and deterministic scheduling problems in single machine environments are explored, the followings are encountered in the domain of due date assignment. [42] examines dynamic rescheduling problem where the existing schedule is updated when new jobs arrive into the system. Jobs are classified as old or new. Old jobs are the ones already arrived to the system whose due dates are assigned but not processed yet. Meanwhile, new jobs arrive for which due dates must be assigned. The due dates of old jobs are assumed as given parameters whereas the due dates for new jobs are decision variables. The objective is to minimize the maximum weighted tardiness penalty and the due date assignment cost. It is shown that the problem is NP-hard but solvable in polynomial time if all old jobs have an equal weight and all new jobs also have another equal weight.

[43] investigates dynamic deterministic single machine problem where there are already scheduled jobs in the system and a new job arrives. The postponement of some of the scheduled jobs is allowed without changing the execution order for a given schedule. Each job has an associated due date which is revealed upon arrival. The objective is to insert the new job in the schedule under real time without violating its due date and also, minimizing the increase of total tardiness that results from postponing some of the scheduled jobs. A two-level algorithm is proposed to solve the problem. In the first level, all the possibilities for inserting a new job are taken into account by relaxing the real-time constraint with no knowledge about the job characteristics before arrival. In the second level, the real-time constraint is considered when a new job arrives and its characteristics become known.

[44] studies a dynamic deterministic due date scheduling problem with the objectives to minimize the average quoted lead time and to maximize the profit obtained from the jobs completed by their due dates, respectively. Preemption is permitted in the setting. An online algorithm having two components is proposed for the problem. The first component sets the due dates and the second component schedules the jobs

in order to meet the assigned due dates.

[45] focuses on the dynamic deterministic single machine scheduling problem in a distributed environment where both the machine and jobs are involved in decision making to create a schedule. Each job is to be completed either on a given distinct due date or within a prescribed due-window. The objective is to minimize the sum of job completion costs within a period. An integer programming formulation is presented which utilizes forecasts of future job arrivals and updates the information on rolling basis. Also, Lagrangian relaxation is used to decompose the formulation.

[46] proposes two new dynamic due date assignment rules which use shop status information together with job related data in a single machine shop under dynamic and deterministic setting. [47] considers a dynamic deterministic single machine scheduling and due date assignment problem to minimize earliness, tardiness and due date related penalties. A common due date is to be assigned for all jobs. It is shown that the problem is strongly NP-hard and an algorithm is constructed for finding the optimal due date under two special cases: predetermined job sequence and equal processing times.

Problems concerning due date assignment under dynamic and stochastic setting in single machine shops are relatively less in number when compared to others. [48] introduces a dynamic priority rule in order to assign due dates for arriving jobs at their arrival time and sequence them. Processing times are assumed to follow exponential or normal distribution. The objective is to minimize average tardiness. The suggested rule is a combination of shortest processing time (SPT) and earliest due date (EDD) priority rules. It is based on ranking the available jobs in queue and picking up the one according to this rule when the machine becomes idle. The robustness of the proposed algorithm is demonstrated using simulation experiments.

[49] suggests a new method for setting due dates to meet a target service level in a dynamic stochastic environment from the point of safe scheduling. The method is called statistical service control (SSC) and used to observe the performance of the

quoted due dates by making use of a control chart approach in order to detect the time when the system is out of control and parameters need to be adjusted. The method does not rely on any distributional assumption and provides a mechanism for quoting the shortest possible due date consistent with a given service level constraint.

Some researchers consider lead time quotation and due date assignment problems within a queuing framework under dynamic setting. [50] studies dynamic deterministic due date setting problem in single machine shop that is modeled as an M/M/1 queuing model operating under FCFS priority scheme. A new due date setting procedure is proposed as a solution to minimize mean tardiness and fraction tardy.

[51] proposes a robust lead time quotation policy to be applied to all customers for minimizing average lead time subject to customer service constraints in stochastic dynamic M/M/1 models (with exponential interarrival times and exponentially distributed processing times) and M/G/1 models (with exponential interarrival times and normally distributed processing times). Three different measures of customer service are considered: fraction of tardy jobs, average tardiness and average relative tardiness (average tardiness as a percent of the lead time). It is shown that the solution to the lead time quoting problem depends strongly on the measure used to represent customer service and also flow times, which in turn depend on processing times.

[52] examines a dynamic stochastic lead time quotation problem in an M/M/1 base stock inventory queue system by jointly considering order acceptance, lead time quotation and inventory decisions simultaneously in order to maximize expected profit subject to lateness penalties incurred for missed due dates and on-hand inventory holding costs. An analytical model is proposed to find the optimal base stock level (the minimum level of inventory required) and the corresponding optimal lead time quotation policy. A two-stage solution procedure is proposed. The first stage corresponds to the determination of the optimal lead time quotation policy for a given base stock level and in the second stage, the optimal base stock level is determined. It is shown that the optimal base stock level increases as the flexibility of the lead time quotation decreases or as the customer's sensitivity to the lead time quotes increases.

[53] deals with simultaneous due date quotation and sequencing problems in the setting of a multiclass M/G/1 queuing system where there are multiple job types and these differ in their arrival times, processing times and cost-profit structures. New due date management policies are identified which are shown to outperform conventional due date setting policies as a result of simulation experiments.

[54] considers the combined problem of due date quotation and sequencing for various customer classes having different price and lead time preferences with regard to a single product. The problem is modeled as a multiclass M/G/1 queue. A heuristic is developed that considers customer preferences during due date quotation process.

[55] studies dynamic stochastic single machine problem in order to minimize average quoted lead time. A class of polynomial time online heuristics are developed for due date quotation problem which is followed by a probabilistic analysis of the performance of these algorithms. Then the conditions, under which these heuristics are asymptotically optimal, are characterized.

The studies examined up to so far do not involve family setups. The following studies include family setups and/or batching (family splitting) issues. In these studies, setup time is considered in two different kinds. If setup time is said to be sequence-dependent, then the setup duration depends both on the family to be processed next and the family of the immediately preceding batch. On the contrary, if setup time is assumed to be sequence-independent, then setup duration depends only on the family of the next batch to be processed. The models can also be categorized as batch or job (item) availability models. In the former case, all the jobs of the same batch become available for processing and leave the machine together, hence all jobs in the same batch have the same start and completion times with each other. In the latter case, each job's start and completion times are independent of other jobs in its batch and can be delivered as soon as they have been processed.

[56] studies a static deterministic batch delivery single machine problem where jobs are to be delivered in batches to the downstream customer in the supply chain so

as to minimize earliness, tardiness, holding, due date assignment and delivery costs. There is an associated acceptable lead time for each customer. No due date assignment penalty is charged if the quoted lead time is not greater than the acceptable lead time. The problem is proved to be NP-hard. However, a polynomial time dynamic programming algorithm is proposed for two special cases: the case of identical processing times and the case where the acceptable lead times are all equal to zero and the holding penalty is less than the tardiness or due date assignment penalty.

[57] considers the problem where several batches of jobs need to be processed by a single machine in static deterministic environment. All jobs within the same batch have a common due date which is either externally given as an input or internally determined as a decision variable. Two problems are investigated. The first problem is to find an optimal schedule so that the total earliness and tardiness penalties are minimized, provided that the common due date is externally given. It is shown that this problem is NP-hard even if there are only two batches of jobs and the two due dates are given unrestrictively large. The second problem is to find an optimal schedule and an optimal due date for each batch so that total earliness, total tardiness and total due date penalties are minimized, provided that each due date is a decision variable. A polynomial time dynamic programming algorithm is proposed for solving both of the problems when the common due dates for different batches are all equal.

[58] examines static batching problem in a single machine environment where the processing time of a job depends on the batch to which it is assigned. Batch availability model is applied in which jobs within the same batch have the same completion time which is equal to the completion time of the last job in that batch. If a job is completed before this time, it must wait until delivery which causes an inventory holding cost. There is also a batch delivery cost which is an increasing function that depends on the number of batches formed. The problem is to determine the optimal number of batches, the assignment of jobs to the batches and the job sequence in order to minimize batch delivery and total earliness cost. Given an upper bound for the number of batches, two pseudo-polynomial algorithms are proposed under dynamic programming approach. A polynomial algorithm is also presented for equal processing times.

[59] considers static deterministic single machine batch delivery scheduling and common due date assignment problem where an additional rate-modifying activity is allowed and completed jobs are delivered in batches. Once this type of an activity is performed, the machine can be said to become more efficient leading to shorter processing times of jobs. However, this activity requires a certain execution time. The objective is to minimize the sum of earliness, tardiness, holding, due date and delivery costs. Polynomial time algorithms are proposed for some special cases of the problem including the case with identical modifying rates and the case with identical processing times.

[60] focuses on the combined static deterministic scheduling and batch sizing problem by considering lower and upper bounds on the batch sizes on a single machine, assuming batch availability model and multi-pegging case. As opposed to mono-pegging case where each customer order is to be satisfied from a single batch, in multi-pegging case orders can be met from different batches. The aim is to minimize the sum of tardiness costs and setup costs arise from creating a new batch. A dynamic programming algorithm is proposed for the special case where a common due date is to be determined in multiple order problem. Moreover two special cases, the case without setup cost and unconstrained batch sizes and the case without setup cost and fixed batch sizes, are shown to be solved in polynomial time.

[61] explores static deterministic single machine batch sequencing problem where a sequence-dependent setup time is required before processing different part types. A polynomial time heuristic and an exact algorithm running in polynomial time given a fixed upper bound on the number of setups are suggested in order to find a sequence of batches of parts.

[62] combines scheduling, batching and due date assignment decisions for a single machine environment in static and deterministic setting. A sequence-independent setup time is required between different batches. A common due date is to be determined for all jobs to minimize the sum of the due date assignment penalty and the weighted number of tardy jobs. Computational complexities of various special cases are

examined. Some of them are found to be NP-hard and polynomial time algorithms are proposed for some special cases. However, the case with equal setup times and equal job processing times is indicated to be open. Yet, [63] proves that the computational complexity of this special case is NP-hard.

[64] addresses a static deterministic scheduling problem for minimizing total tardiness in identical parallel machine environment where job splitting is allowed by simultaneously processing small sub-batches of a job on more than one machine and sequence-dependent major and minor setup times and due dates are taken into consideration. A minor setup time is required between processing jobs that belong to the same family whereas a major setup is incurred before processing a job from a different family. Heuristic algorithms that produce near optimal solutions are proposed.

[65] considers static deterministic sequence-independent class setup scheduling problem on a single machine with the objective to minimize the maximum lateness. A mathematical programming model is presented to attack the problem but since the problem is NP-hard, a heuristic algorithm is proposed to find near optimal solutions.

[66] studies static deterministic single machine family scheduling problem. A common due date is to be assigned for all jobs while trying to minimize the weighted earliness and tardiness penalties. A branch and bound algorithm is proposed for finding an optimal solution that works when given enough time and a faster beam search procedure is presented for finding near optimal solutions.

[67] deals with static deterministic integrated due date assignment and scheduling problem with sequence-dependent setups on a single machine. A common due date is to be assigned for all jobs so as to minimize the sum of earliness, tardiness and due date assignment penalties. An MIP model is presented. A branch and bound algorithm is proposed for small sized instances whereas heuristic algorithms are proposed for large sized instances to solve the problem within a reasonable computation time.

[68] studies static and deterministic batch scheduling problem involving family

setups in single and identical parallel machine environments to minimize the total of completion or weighted completion times. Computational complexities are explored under different cases.

[69] investigates a batch scheduling problem for a single machine that processes parts of a single item in which processing time of a part increases proportionally with its waiting time. The objective is to minimize total actual flow time. The problem is formulated as a non-linear programming model. The number of batches, batch sizes and sequence of the batches are decision variables. In order to find the optimal solution, a relaxation is applied on the number of batches by considering it to be a parameter. Based on the relaxation approach, the model is solved through a set of iterations. The model is initially solved for taking the number of batches as one and the value is increased by one gradually. In each step, the current solution is compared to the previous solution and if the solution is improved, the process is continued; otherwise, the iterations stop.

[70] addresses a dynamic deterministic scheduling problem involving ready dates, due dates and family setup considerations on a single machine to minimize the maximum lateness. A branch and bound algorithm is proposed that gives optimal solutions for instances of reasonable size.

[71] analyzes dynamic deterministic rescheduling problem for a single machine environment where a setup time is incurred between jobs of different part types. In the problem under consideration, there is a set of jobs whose due dates are already assigned and must be met. The goal is to integrate a set of new arriving jobs into an existing schedule with the least disruption. The objective is to minimize total weighted completion time or maximum completion time of new jobs subject to incurring no additional setups and preventing existing jobs to become tardy. A polynomial time algorithm is proposed for the maximum completion time objective. It is also proved that the total weighted completion time problem with arbitrary weights is NP-hard. Hence, two heuristics are provided for this objective.

[72] studies dynamic stochastic single machine scheduling problem with sequence-dependent setups. A multi-objective is considered that concerns the average and standard deviation of two conflicting performance measures: cycle time and delivery accuracy. Eight dispatching rules are evaluated and a simulation is carried out under different configurations in order to minimize four individual scheduling objectives. The results are tabulated according to the number of problem instances in which the dispatching rule gives the best value and outperforms the others in terms of the objective to be minimized.

[73] presents a rescheduling algorithm for a dynamic stochastic single machine system with setup times by grouping jobs of similar types according to first in first out (FIFO) dispatching rule under periodic and event-driven rescheduling strategies. In the problem under consideration, different job types arrive dynamically and job processing times are exponentially distributed. Schedule must be updated in order to include new arrivals. In the periodic strategy, rescheduling is performed at constant times and rescheduling period is fixed. In the event-driven rescheduling strategy, rescheduling is triggered by a job arrival that brings the queue of unscheduled jobs to a specific threshold.

[74] considers a dynamic stochastic batch processing problem in a MTO environment where the situation at the supplier is modeled as a single machine with order acceptance and rejection decisions. Any order that cannot be delivered by its due date has to be rejected at arrival. Under the setting, acceptance of an order results in a family-dependent reward and the objective is to maximize supplier's long-run profit via expected rewards for acceptance orders. A method is constructed to estimate family-dependent lead times and it is shown that quoting lead times based on the new proposed method leads to significant gains in profit compared to standard lead time estimation methods, independent of the families.

Under group technology setting, [75] examines a single flexible machine problem in a group technology environment under static deterministic setting. A major sequence-independent setup is required before processing a job from another group. A common due date is to be quoted for all jobs so as to minimize due date related cost and the cost of tardy jobs. Two special cases are considered. In the former case, jobs are only allowed processing in groups and in the latter case, jobs from different groups can be processed together in order to take the advantage of short processing times of some jobs. Polynomial time optimization algorithms are presented to solve the problem.

[76] studies due date assignment problem in a static deterministic single machine group technology environment where a sequence-independent setup time is required before processing a different family. The objective is to find the optimal family sequence and job sequence within each family so as to minimize the sum of earliness, tardiness and due date assignment penalties. A hybrid due date assignment method is proposed by restricting all jobs within the same group (family) to be assigned the same due date but different due dates are quoted for different families.

[77] considers simultaneous due date assignment and scheduling problem in a group technology environment under static deterministic setting. Due dates for all jobs within the same group are assigned based on three due date setting rules. The objective is to minimize a cost function that includes earliness, tardiness, due date assignment and flow time penalties. A unified polynomial time algorithm is proposed for all of the three due date assignment methods.

Static single machine due date related problems are tabularized in Table 2.1 with respect to the objective to be minimized, solution approach and consideration of family setup. Likewise, dynamic single machine due date related problems are tabularized in Table 2.2.

Table 2.1. Static Single Machine Due Date Related Problems in Literature.

Static/ Dynamic	**Deterministic/ Stochastic**	**Reference**	**Objective to be minimized**	**Solution approach**	**Setup? Yes/No**
Static	Deterministic	[4]	earliness, tardiness and due date assignment costs	optimization algorithm	No
		[5]	earliness, tardiness and due date assignment costs	optimization algorithm	No
		[6]	earliness, tardiness and due date assignment costs	optimization algorithm	No
		[7]	earliness, tardiness and due date assignment penalties	linear programming	No
		[8]	earliness, tardiness and due date assignment penalties	sequencing algorithm	No
		[9]	due date assignment cost and weighted number of tardy jobs	dynamic programming	No
		[10]	due date assignment cost and weighted number of tardy jobs	optimization algorithm	No
		[12]	1) earliness, tardiness and due date assignment penalties 2) penalties due to number of tardy jobs and due date assignments	optimization algorithm	No
		[13]	the maximum lateness	optimization algorithm	No
		[14]	total weighted earliness, tardiness and due date assignment cost	optimization algorithm	No
		[15]	total weighted earliness, tardiness and due date assignment cost	optimization algorithm	No
		[56]	earliness, tardiness, holding, due date assignment and delivery costs	optimization algorithm	Yes
		[57]	earliness, tardiness and due date assignment penalties	dynamic programming	Yes
		[58]	batch delivery and earliness cost	dynamic programming	Yes
		[59]	earliness, tardiness, holding, due date and delivery costs	optimization algorithm	Yes
		[60]	the sum of the tardiness and setup costs	optimization algorithm	Yes
		[62]	due date assignment penalty and weighted number of tardy jobs	dynamic programming	Yes
		[65]	the maximum lateness	heuristic algorithm	Yes
		[66]	weighted earliness and tardiness penalties	branch and bound	Yes
		[67]	earliness, tardiness, due date assignment penalties	branch and bound	Yes
		[75]	the costs of tardy jobs and due date related costs	optimization algorithm	Yes
		[76]	earliness, tardiness and due date assignment penalties	optimization algorithm	Yes
		[77]	earliness, tardiness, due date assignment and flow time penalties	optimization algorithm	Yes
	Stochastic	[21]	total weighted deviations of job completion times from a common due date	optimization algorithm	No
		[22]	expected maximum tardiness	dynamic programming	No
		[23]	expected earliness and tardiness penalties	heuristic algorithm	No
		[24]	expected total earliness and tardiness cost	optimization algorithm	No
		[26]	earliness, tardiness and due date assignment penalties	optimization algorithm	No
		[27]	earliness, tardiness, due date assignment costs and makespan	optimization algorithm	No
		[28]	earliness, tardiness and due date assignment penalties	optimization algorithm	No
		[29]	earliness, tardiness and due date assignment penalties	optimization algorithm	No
		[30]	earliness and due date assignment penalties and weighted number of tardy jobs	optimization algorithm	No
		[31]	total earliness, total tardiness, weighted number of tardy jobs	optimization algorithm	No
		[32]	earliness, tardiness and due date assignment penalties	heuristic algorithm	No
		[33]	cost of changing due dates, total cost of discarded jobs, earliness penalties of the scheduled jobs	dynamic programming	No
		[34]	cost of changing due dates, total cost of discarded jobs, total cost of positional weighted earliness	optimization algorithm	No
		[35]	earliness, tardiness, due date assignment and resource consumption costs	optimization algorithm	No
		[36]	expected number of tardy jobs	mathematical programming	No
		[37]	general stochastic cost function including a random due date and weight	sequencing algorithm	No
		[39]	the sum of assigned due dates and expected tardiness cost (with and without weights)	heuristic algorithm	No
		[41]	the maximum due date	sequencing algorithm	No
		[69]	total actual flow time	non-linear programming	Yes

Table 2.2. Dynamic Single Machine Due Date Related Problems in Literature.

Static/ Dynamic	**Deterministic/ Stochastic**	**Reference**	**Objective to be minimized**	**Solution approach**	**Setup? Yes/No**
Dynamic	Deterministic	[42]	maximum weighted tardiness penalty and due date assignment cost	optimization algorithm	No
		[43]	total tardiness of postponing already scheduled jobs	optimization algorithm	No
		[44]	1) average quoted lead time	online algorithm	No
			2) total profit obtained from the jobs completed by their due dates		
		[45]	total job completion costs	integer programming	No
		[46]	1) average flow time	dynamic due date assignment policy	No
			2) average lateness		
			3) average tardiness		
			4) percent tardy jobs		
		[47]	earliness, tardiness and due date assignment penalties	optimization algorithm	No
		[50]	mean tardiness and fraction tardy	due date assignment policy	No
		[70]	the maximum lateness	branch and bound	Yes
		[71]	1) total weighted completion time	heuristic algorithm	Yes
			2) maximum completion time		
	Stochastic	[48]	average tardiness	dynamic dispatching policy	No
		[49]	mean and standard deviation of earliness, lateness and missed due dates	statistical service control	No
		[51]	average lead time	dynamic lead time quotation policy	No
		[53]	weighted average quoted lead time	due date assignment policy	No
		[55]	average quoted lead time	heuristic algorithm	No
		[72]	average and standard deviation of cycle time and tardiness	discrete event simulation	Yes

From the point of due date setting rules, generally due dates are assigned by specifying a flow allowance in addition to job arrival times. Then the due date of job j can be denoted as $d_j = a_j + \delta_j$ where a_j indicates the arrival time and δ_j denotes the flow allowance of job j. In Table 2.3, studies are categorized based on due date setting rules where p_j represents the processing time requirement of job j. Low flow allowances lead to tight due dates whereas high flow allowances cause loose due dates. The tightness of due dates depend on parameters of due date assignment rules. The selection of appropriate parameters are not straightforward and generally chosen via simulation experiments based on the objective function to be optimized.

Due date assignment rules can be classified as three types: the rules ignoring job and shop related information (such as RND and SLK), the rules considering only job related data but ignoring shop status (such as TWK and PPW) and the rules that consider both job characteristics and shop congestion.

Table 2.3. Due Date Setting Rules.

Due date assignment method	**Explanation**	**Flow allowance**	**Due date representation**	**References**
DIF (Unrestricted)	Each job can have a different due date based on distinct flow allowances.	not available	not available	[5], [6], [9], [10], [12], [13], [18], [19], [20], [22], [23], [27], [32], [33], [34], [35], [39], [40], [41], [42], [43], [44], [45], [49], [52], [53], [55], [56], [58], [60], [61], [71], [77]
RND (Random allowance)	Each job receives a random due date.	$\delta_j = \kappa \cdot e_j$ where e_j is a random number and κ is a decision variable	$d_j = a_j + \kappa \cdot e_j$	[37], [72]
CON (Common/Constrained)	A constant flow allowance is assigned for all jobs.	$\delta_j = \gamma$ where γ is a decision variable	$d_j = a_j + \gamma$	[4], [7], [8], [12], [14], [15], [21], [26], [27], [28], [29], [30], [33], [35], [36], [40], [41], [47], [57], [59], [60], [62], [66], [69], [75], [77], [78], [79], [80]
SLK (Slack)	A common slack (equal waiting time allowance) is assigned to all jobs.	$\delta_j = p_j + \beta$ where β is a decision variable	$d_j = a_j + p_j + \beta$	[27], [30], [34], [35], [48], [77], [78], [80], [81]
TWK (Total Work Content)	Flow allowances are proportional to job processing times.	$\delta_j = \alpha \cdot p_j$ where α is a decision variable	$d_j = a_j + \alpha \cdot p_j$	[46], [48], [50], [82], [78], [80], [81]
PPW (Process Plus Wait)	Combines SLK and TWK in one model, due dates are linear functions of job processing times.	$\delta_j = \beta + \alpha \cdot p_j$ where α and β are decision variables	$d_j = a_j + \beta + \alpha \cdot p_j$	[81]

[83] investigates due date assignment rules in single and parallel machine models by analyzing relevant papers in literature. The methods of how to determine the optimal values of controllable parameters in due date setting rules are discussed.

Different due date setting rules perform differently under various dispatching rules. Dispatching rules provide reasonably good solutions in a relatively short time by prioritizing all jobs waiting to be processed based on a ranking procedure and select the job with the highest priority to be scheduled when the machine becomes free.

Table 2.4. Priority Dispatching Rules.

Static/ Dynamic	**Dispatching rule**	**Explanation**	**Sequencing principle**
Static	FCFS ≡ FIFO ≡ ERD	First Come First Served ≡ First In First Out ≡ Earliest Release Date	Nondecreasing order of a_j
	SPT	Shortest Processing Time	Nondecreasing order of p_j
	WSPT	Weighted Shortest Processing Time	Nondecreasing order of (p_j/w_j)
	LPT	Longest Processing Time	Nonincreasing order of p_j
	WLPT	Weighted Longest Processing Time	Nonincreasing order of (p_j/w_j)
	EDD	Earliest Due Date	Nondecreasing order of d_j
	EFT	Earliest Finish Time	Nondecreasing order $(a_j + p_j)$
Dynamic	MS	Minimum Slack	Min $(d_j - a_j - t)$
	CR	Critical Ratio	Min $(d_j - t)/p_j$
	MDD	Modified Due Date	Max $(d_j, p_j + t)$
	SPTA	Shortest Processing Time among Available Jobs	Minimum processing time in the set of released but not processed jobs yet
	ATC	Apparent Tardiness Cost	Combines WSPT, MS
	ATCS	Apparent Tardiness Cost with Setups	Combines WSPT, MS, SST

In Table 2.4, commonly used dispatching rules are shown where t denotes the time when the machine becomes idle and a job is to be selected for processing from the queue of waiting jobs, a_j is the arrival time, p_j is the processing time, d_j is the due date and w_j is the weight or priority associated with job j.

Dispatching rules can be classified as static and dynamic. Static dispatching rules are not time dependent whereas dynamic dispatching rules are time dependent. In static dispatching rules, once the jobs are ranked, the priority of jobs stays the same whereas in dynamic dispatching rules, priorities of jobs might change during time.

Revisiting the studies examined before, [27] minimizes its objective function under three different due date assignment methods (CON, SLK and DIF) by considering learning and deterioration effects. [28] proposes two different polynomial time algorithms for two due date assignment rules (CON and SLK) by taking only the deterioration effect into account.

[46] develops two different due date assignment rules by assigning flow times based on current shop information and states that newly developed shop information based due date assignment rules outperform job information based rules such as TWK.

[48] assesses the performance of the proposed dynamic priority rule, namely modified due date (MDD), by comparing it with two pure dispatching rules (SPT and EDD) under two different due date assignment methods (SLK and TWK). It is shown that the suggested modified rule dominates two pure dispatching rules in terms of average tardiness.

[50] compares the new proposed heuristic due date assignment rule with TWK and shows that the new method exhibits superior performance in comparison to TWK in terms of percent tardy and mean tardiness under four priority dispatching rules (FCFS, SPT, EDD and MDD) for different utilization levels.

[77] sets due dates for all jobs within the same group based on three due date setting rules (CON, SLK and DIF). A unified optimization algorithm is proposed which works under all of the three due date assignment methods.

Additionally, [82] considers due date assignment and scheduling problem on a single machine under TWK due date assignment method in static and deterministic environments. The stochastic case is also analyzed as an extension by relaxing deterministic processing time assumption.

[84] examines SLK due date assignment rule in a single machine setting. The optimal flow allowance is to be determined to minimize the sum of flow allowance and the weighted earliness and tardiness costs.

[78] compares three basic strategies (CON, SLK and TWK) for due date assign-

ment in single machine environment where preemption is allowed under both static and dynamic models.

[80] explores the relationship between due date assignment rules (CON, SLK and TWK) and the priority dispatching rules (ERD, SPT, EDD, MS and EFT) in a dynamic single machine model. Under the setting, due date of an arriving job is quoted at its arrival time. Via simulation experiments, the study provides insights about the conflict between tight and loose due dates based on different combinations with the objective of making due dates as tight as possible while avoiding tardiness as much as possible.

[85] examines due date assignment problem in real time in MTO environments under dynamic arrivals. A non-parametric flow time estimation method is proposed that considers current system state in terms of resource availability, existing commitments and routings. It is shown that, the proposed method outperforms some of the commonly used parametric flow time estimation methods used for due date assignment such as CON and TWK.

[86] overviews elementary dispatching rules and suggests a composite dispatching rule called Apparent Tardiness Cost (ATC) heuristic that combines the WSPT rule and the MS rule. Whenever the machine gets idle, the job with highest ranking index is picked up to be processed next on the machine. The ranking index contains a parameter which is a function of due date tightness factor and due date range factor. The suggested rule is also extended to include release dates and sequence-dependent setup times and called Apparent Tardiness Cost with Setups (ATCS). ATCS combines Shortest Setup Time (SST) rule with WSPT and MS. In this rule, while calculating the ranking index, an additional setup time related parameter is considered, namely setup time severity factor.

There are also surveys in the scope of due date assignment problems. [79] introduces a review on due date related scheduling problems in a single machine environment. In the problems under consideration, the common due date is a decision variable and the aim is to find the optimal value of the common due date and the optimal schedule. The complexities of the proposed algorithms in the examined articles are summarized.

[81] provides a study about various due date assignment methods and scheduling decisions under static deterministic setting for single machine case. The properties of optimal solutions and the complexity of the proposed algorithms are examined. [87] presents a comprehensive review of literature on static and deterministic problems involving setup times as of main concern.

[88] supplies a comprehensive review for static problems considering setup times since the mid 1960s. The problems under review are classified as sequence-independent and sequence-dependent based on setup type; batch and non-batch depending on grouping of jobs and also categorized according to the shop environment. [89] can be seen as an extension to [88] which covers about three hundred additional papers from then on. The survey classifies scheduling problems as problems with batching and non-batching, sequence-dependent and sequence-independent setup time considerations and job and batch availability models and shop environments. This research addresses stochastic scheduling problems with setup times as a worthy direction of research which has great potential.

[90] gives an extensive survey about problems which combine scheduling with batching decisions. Job availability and batch availability models are examined. Two different environments are considered. The first one is classical single machine that can handle at most one job at a time. The second is the batching machine which can process several jobs simultaneously. The research classifies problems as polynomially or pseudo-polynomially solvable, NP-hard in the ordinary or strong sense or open.

[91] presents a survey of scheduling problems containing due date assignment decisions in static and dynamic jobshops and classifies related papers based on performance measures and due date assignment methods.

[92] provides a survey about scheduling problems involving batch setup times in single and identical parallel machines. The study extends numerous scheduling models to include batch setup times so as to minimize maximum lateness, total weighted completion time and number of late jobs. The problems are classified as efficiently solvable or NP-hard, depending on the objective to be minimized and type of setup times.

[93] addresses the due date management problem by focusing on articles which propose analytical inferences and examine the proposed algorithms and outcomes in those articles. Problems are divided into a class of models as single machine static models with common and distinct due dates, single machine dynamic models and parallel machine, jobshop and flowshop models.

[94] reviews a broad range of articles regarding due date assignment problems. The authors categorize due date management problems according to objectives, solution approaches, commonly used due date setting rules and accompanying priority dispatching rules, offline and online models and models in the presence of service level constraints.

[95] surveys outcomes on batching models that determine whether or not to schedule jobs with similar characteristics consecutively and lot-sizing models that specify how and when to split a job in single and multi stage systems. The computational complexities for a variety of problems are investigated.

[96] addresses elementary results for problems with due dates regarding lateness criteria, the number of tardy jobs and mean tardiness by illustrating theorems and algorithms used for related problems.

In the light of the studies examined above, the majority of research is carried out within static and deterministic setting and family setups are not considered. To the best of our knowledge, so far there is no study that concerns due date assignment problem including family setups under dynamic and stochastic environment so as to minimize the aforementioned conflicting objectives. In that sense, this study will be a contribution in order to fill part of this gap.

The detailed problem description under consideration is given in the next chapter.

3. PROBLEM DEFINITION

Due date is the date when a job is promised to be delivered to customers. Within the framework of due date assignment, in a dynamic production system, at any time instance t, there can be: (i) a set of jobs whose due dates are assigned and already completed by t, (ii) a set of non-complete jobs whose due dates are assigned but not completed by t and, (iii) a set of jobs whose due dates are not assigned yet.

Depending on the closeness to the arrival times, quoted due dates can be set as tight or loose. Tight due dates mean shorter quoted lead times and fast response to customer orders which can lead to gain competitive advantage in the market for a company by delivering orders earlier than its competitors. Therefore, tight due dates are more attractive for customers. However, each assigned due date means a commitment between the manufacturer and the customer. Hence, tight due dates may cause tardiness of non-complete jobs if the orders are not delivered on the promised due date. In addition to direct costs of tardiness, the company may face with loss of customer goodwill, bad reputation and loss of market share in the long run due to missed due dates, especially in highly competitive markets.

If the company sets loose due dates through long quoted lead times to prevent tardiness and gain production flexibility, and if the firm is not monopole and there is high competition, then the company may lose its customers to a rival which promises earlier delivery. Moreover, if the assigned due dates are too loose, it leads to pile up inventory in stock until delivery which in turn will result in holding and storage costs, tied-up capital or opportunity costs and possible spoilage costs. Thus, there is a trade-off between assigning tight and loose due dates.

Batch processing companies that take place in high competitive markets are our special interest. This type of companies have product families and a significant setup time is required before processing a job from another family. Metal, automotive supplier, glass, paper and textile manufacturers are examples of such companies. During due date assignment process, these family setup times must be taken into consideration.

In literature, one of the most frequently used methods to solve due date assignment problem is to add some flow allowance in addition to the arrival time of jobs. These methods work in general but family setups are usually ignored within the scope of them.

In practice, generally two different due date assignment methods are used basically in the presence of family setups. The first one is the "periodic due date assignment". In this type of due date assignment strategy, due dates are quoted at the end of fixed scheduling periods (for instance, every Monday). In order to visualize periodic due Consider Figure 3.1. Time horizon is divided into fixed scheduling periods. Suppose, there are three families and each family is represented with a different color. Colorful arrows denote job arrivals from these families. The thick yellow arrow shows the time instance that due date quotation takes place. Small patterned boxes indicate family setups whereas larger boxes depict processing time requirement of jobs.

In Figure 3.1, all jobs that arrive within a scheduling period are gathered whose job characteristics reveal at their arrival times. This enables to share a single setup for jobs that belong to the same family and to process them in the same batch where a batch is a group of jobs from the same family which can be processed consecutively by sharing a common setup based on the production similarities. Due to batching, operational efficiency is the major gain of this method. The downside of this method is that customers have to wait to get a due date in addition to the flow time of the jobs, in other words time spent in the system. This method may be acceptable for a certain segment of customers or markets but this is not a valid procedure for every industry, especially for highly competitive ones where there may be competitors that offer closer due dates.

As an alternative approach, due dates are quoted at the time of job arrivals immediately. This policy is known as "immediate due date assignment". By this way, the waiting time to get a due date is removed for customers. But this time, since job

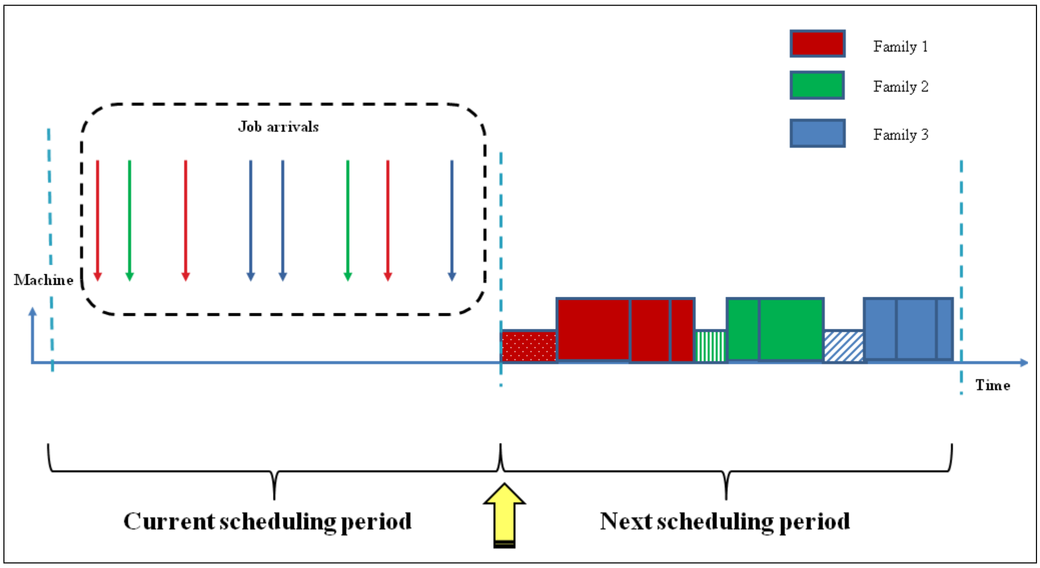


Figure 3.1. Periodic Due Date Assignment.

related data is not known in advance at the time of due date quotation, batching is not possible in a straightforward way, except FSFC rule. However, straightly assigning a new arrival to the first available position requires a setup if two consecutive jobs do not belong to the same family. Hence, assigning the new arrival to the first available position is not reasonable in the presence of family setups.

According to FCFS rule, jobs can be assigned to the machine individually according to the sequence they arrive. There will not be a setup in between two consecutive jobs if they belong to the same family. Otherwise, there will be a setup every time before processing a new job from a different family. In this case, total setup time requirement of such a schedule will be extremely high. The number of setups can be at most as high as the number of jobs if no two consecutive jobs belong to the same family. Consider Figure 3.2 in order to illustrate immediate due date assignment, based on FCFS principle where thick yellow arcs represent the times when due date quotations take place. The increase in the number of setups can be noticed, due to FCFS processing. Hence, instead of sequencing jobs one by one, a smart procedure is required for gaining operational efficiency under family setups.

In immediate due date quotation process, as it can be seen in Figure 3.3, at the arrival time of a new job, say t, there may be a set of completed jobs, there may be

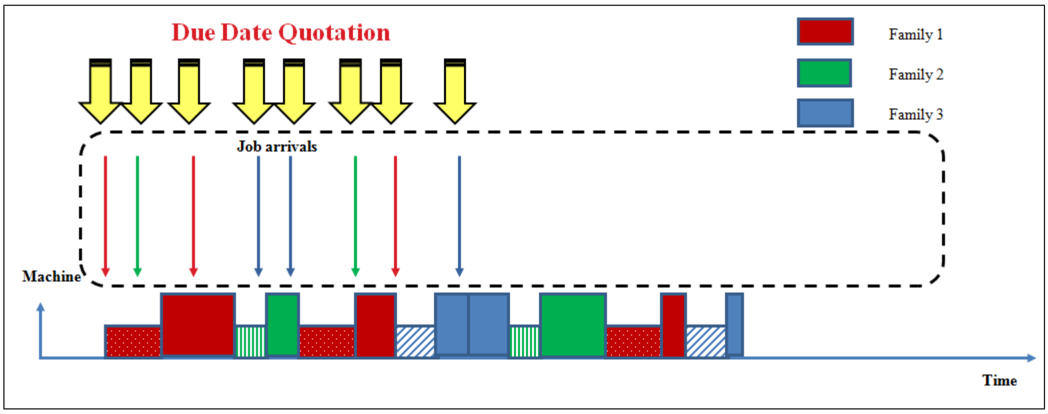


Figure 3.2. Immediate Due Date Assignment.

a set of non-complete jobs, a setup operation may be carried out or there may be a single job being processed, currently. At any time instance, there may be at most one job whose due date is to be assigned instantly, as opposed to the periodic due date assignment policy.

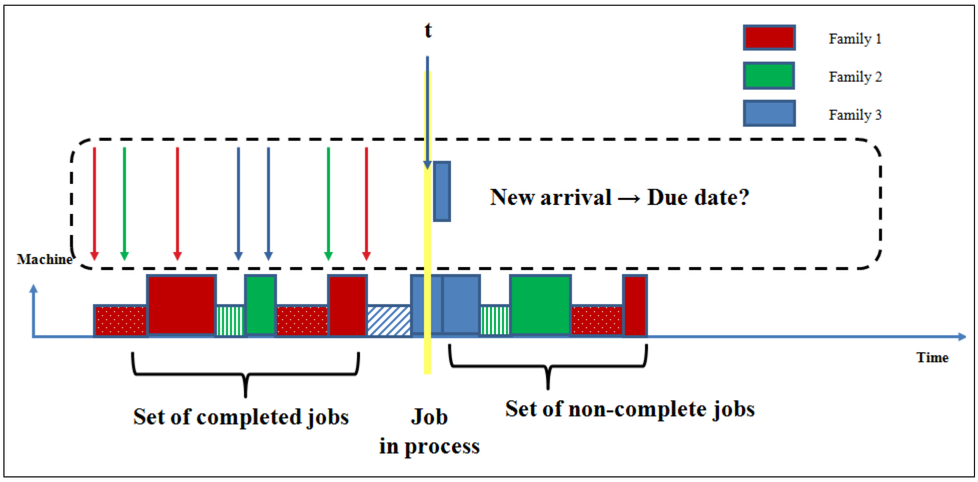


Figure 3.3. Set of Jobs in Production System.

While assigning a due date for a new arrival, although possible cases are not limited with these, there are two basic cases to be considered. As a first option, the new arrival can be assigned to the final position in the existing schedule. In this case, if the last sequenced job is from the same family, no setup is required. However, if the family of the last sequenced job is from a different family, a setup is necessary. As a second alternative, the new arriving job can be appended to the end of the last scheduled job of its family. However, inserting the new job into an existing schedule can be done by shifting the current schedule to the right hand side by the processing time of the new arrival. In the former case, a further due date can be assigned for the new arrival whereas a shorter due date can be assigned in the latter case. Yet, in the latter case, shifting the existing schedule to the right hand side means shifting the completion times of non-complete jobs present in the system.

On the other hand, once the due dates are set through commitments, they cannot be altered. Therefore, if their completion times are increased due to a right shift of the schedule, they may become tardy or their tardiness may increase if they are already tardy. Hence, the decision maker has to decide whether to quote a shorter due date for a new arrival at the expense of causing tardiness for non-complete jobs or assigning a further due date for the new arrival by keeping the promised due dates for the non-complete jobs and preventing their tardiness.

In order to demonstrate the aforementioned two cases, consider Figure 3.4. Suppose eight jobs ($J1$ to $J8$) are already scheduled and the current time is t. $J9$ has just arrived from Family 2 at time t. Instead of incurring a new setup for Family 2 at the end of the last scheduled job $J8$ and assigning a further due date for $J9$, the job could be appended to the end of $J5$ which is the last scheduled job that belongs to Family 2. In this case, the due date assigned for $J9$ would be shorter than the due date that would be assigned as a result of incurring a new setup at the end of the schedule. However, the completion times of the non-complete jobs following $J5$ (namely $J6$, $J7$ are $J8$) would increase due to the right shift of the schedule. Since the due dates for $J6$, $J7$ are $J8$ already promised, they would likely become tardy or their tardiness would get increased if they were already tardy, due to increased completion times.

In order to prevent excessive number of setups, another approach could be allocating families into the schedule as single blocks, instead of processing them according to FCFS basis. Then, a job could be processed only in the portion of the schedule allocated for the family it belongs. A sample allocation can be seen in Figure 3.5. This type of an assignment would decrease the number of setups since a single setup is required only at the beginning of each family once. Yet, due dates for the jobs,

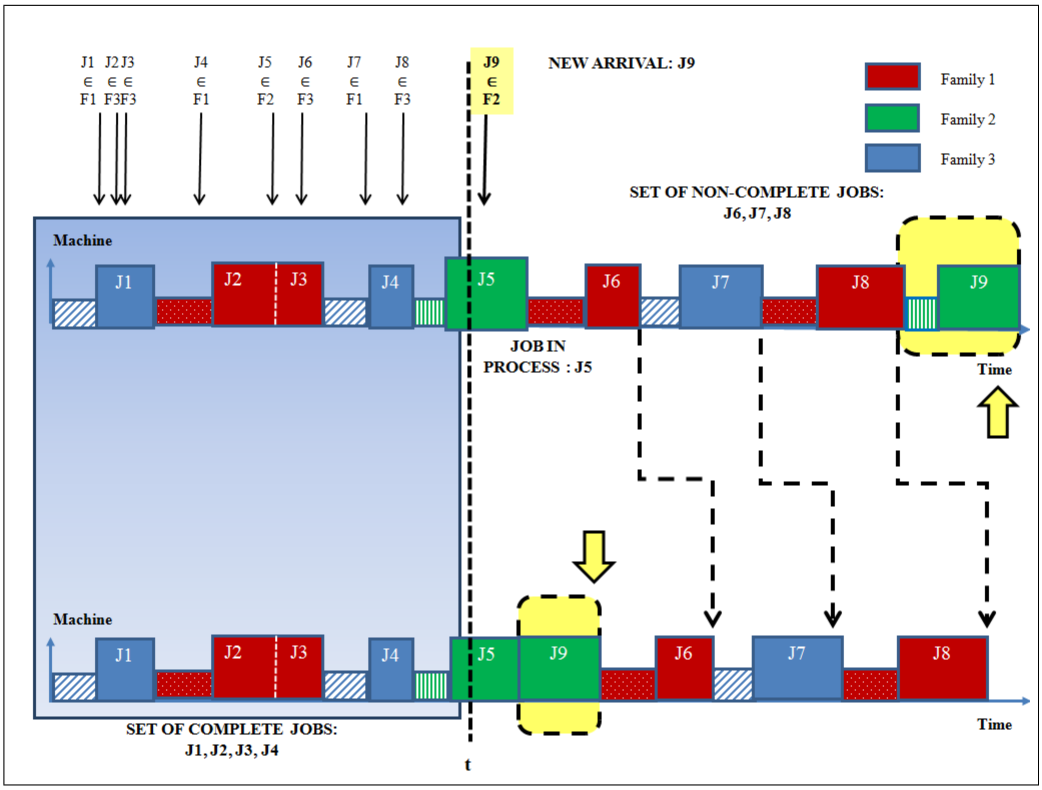


Figure 3.4. Inserting New Arriving Job into an Existing Schedule.

whose families are allocated later in the schedule, will be accordingly later since a job must wait until the start time of the batch allocated to its family, regardless of its arrival time. Hence, maximizing the machine utilization by committing machines to long production runs for families in order to reduce setup time through clustering jobs with similar setup characteristics may lead to longer quoted lead times. Moreover, if jobs arrive after the completion of the single batch allocated for their family, they are supposed to wait until the next cycle. Thus, this method is also not reasonable, especially for batch processing companies in highly competitive markets.

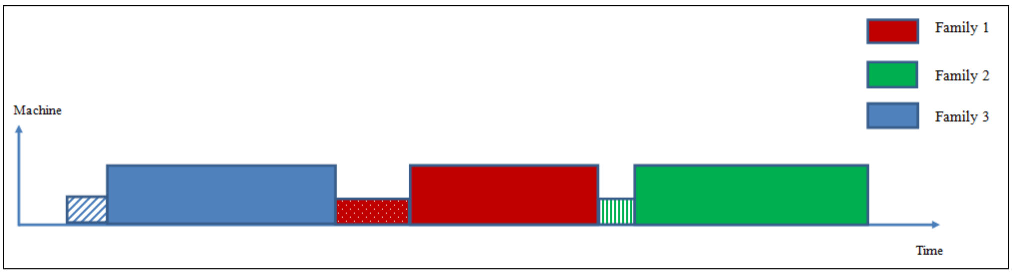


Figure 3.5. Allocation of Families into Schedule as Single Batches.

From manufacturers' point of view, tardiness minimization is vital. On the other hand, from customers' perspective, assignment of a shorter quoted lead time is critical since everyone wants to get their orders as soon as possible. Hence, the objective of the problem should be defined in such a way that it captures the individual objectives of both parties simultaneously. Consequently, a sophisticated and rational way is required to solve due date assignment problem.

In this study, the aim is to assign a due date for new arriving jobs immediately at their arrival, but at the same time, to gain operational efficiency through an immediate due date assignment policy. There are two conflicting objectives to be considered: keeping quoted due dates for new arrivals as tight as possible while avoiding tardiness of non-complete jobs as much as possible. Minimizing due dates ensures attracting and retaining customers. However, due dates must be reliable and attainable as well as being short. In that sense, tardiness measures the compliance to the assigned due dates and is an indicator of the reliability of the quoted due dates.

The problem under consideration can be defined as follows. There is a dynamic single machine production system where jobs are not readily available at the beginning and arrive over time. No job related information (namely arrival times, processing times, family types) is known until arrival. The environment is also stochastic. Job related data is not known certainly. Interarrival times, processing times and family types are stochastic variables that follow probability distributions. The machine can process at most one job at a time. No preemption is allowed. It is assumed that the machine is not subject to any breakdown or maintenance. The manufacturer does not have the option to reject an order. An order is composed of a single job for each customer so order and job are used interchangeably. It is assumed that each job is composed of a single operation and once that operation is processed, the job can be said to be completed.

Let J be the set of jobs to be processed. Associated with each job $j \in J$, there is an arrival time a_j, processing time p_j and an assigned due date d_j. Family types, interarrival times and processing times of jobs are random variables. Also, there is an actual completion time c_j associated with each job j. Jobs belong to different families based on their process similarities in production. For each job j, the family it belongs is characterized by $f \in F$ where F is the set of all families. A family is a set of jobs that require similar setup characteristics. A setup is needed for changing the tools and settings of the machine. Because of production similarities, the jobs from the same family can be processed consecutively on the machine by sharing a single setup. A sequence-independent setup time $\mathcal{S}_f$ is required before processing a job from family f where setup time depends only on the family to be processed, independent of immediately preceding family. Moreover, job (item) availability model is applied where each job has its own start and completion time and a job leaves the machine just after it has been processed, irrespective of other jobs in the batch it belongs.

A due date is to be quoted for every job as soon as it arrives and it is the date by which a job is promised to be delivered to customers. Quoted lead time of job j is the difference between the assigned due date and the arrival time of a job and represented as $Q_j = d_j - a_j$. However, assigned due dates and actual completion times may be different if the promised due dates cannot be met. Flow time (cycle time) of job j is the time spent in the system, or equivalently, the difference between the actual completion time and the arrival time of job j and can be defined as $FT_j = c_j - a_j$. Lateness of job j is the difference between actual completion time and the assigned due date of job j and shown as $L_j = c_j - d_j$. A job is said to be tardy if it is completed after its promised due date and tardiness of job j is denoted as $TARD_j = max\{0, L_j\} = max\{0, c_j - d_j\}$.

In order to generalize the problem, at any time instance t, when a new job i arrives, the system appears as in Figure 3.6. There is a set of jobs which are already completed. There is a set of jobs whose due dates are assigned but not completed yet. Presently, there may be a single job being processed or a setup could be in process.

At any time instance t, if a snapshot of the system is taken and jobs are observed, the jobs fall into one of these categories below:

(i) The jobs that have already arrived and due dates are already assigned and com-

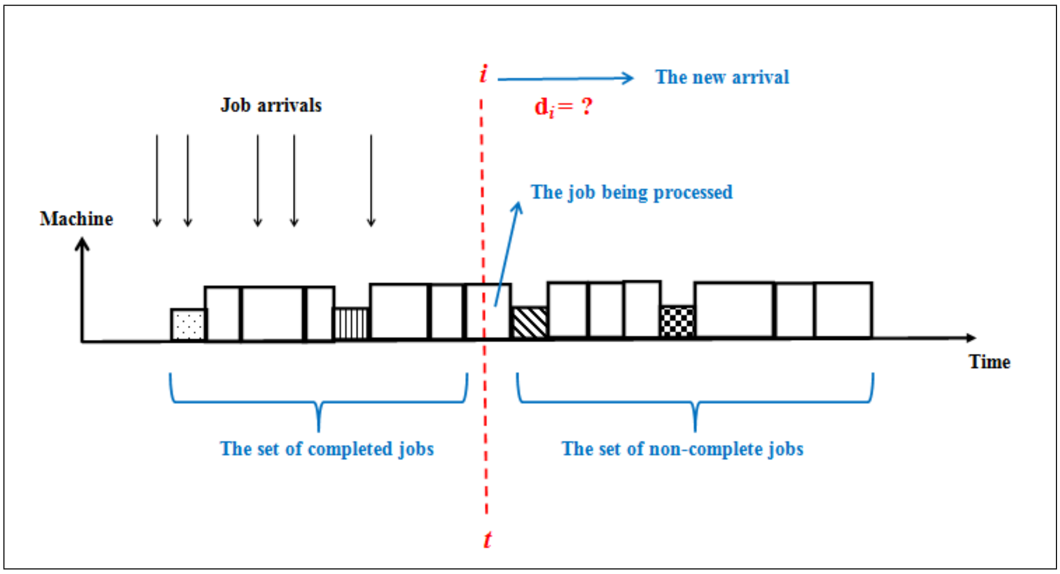


Figure 3.6. System Snapshot at any Time Instance t.

pleted by time t. Let this set be denoted by $J_t^C = \{j \in J : a_j < t, c_j < t\}$.

(ii) The jobs that have already arrived and due dates are already assigned but waiting to be processed and not completed by time t. Let this set be denoted by $J_t^{NC} = \{j \in J : a_j < t, c_j > t\}$.

(iii) The job that is being processed on the machine at time t. Let this set be denoted by j^p where $a_{j^p} < t$ and $c_{j^p} > t$ considering that $j^p \in J_t^{NC}$.

(iv) The job that arrives exactly at time t. Let this single job be denoted by i where $a_i = t$ and $c_i > t$.

The trade-off between assigning tight and loose due dates reflects the trade-off between competitive power and delivery performance. In relation to this trade-off, two cost functions are defined:

There is a cost related with competitive power which is a function of a job's arrival time and assigned due date and can be denoted by: $g_j(a_j, d_j)$

There is another cost related with late delivery which is a function of a job's completion time and assigned due date and can be denoted by: $h_j(c_j, d_j)$

The ultimate goal in this study is to assign a due date immediately for a job at its arrival time by considering the mentioned trade-offs and family setups.

In the long run, two objective functions are simultaneously to be minimized through a due date assignment policy:

(i) minimize asymptotic competitive power cost per job

$$\lim_{t\to\infty} \frac{\sum_{j\in J_t^C} g_j(a_j, d_j)}{\mid J_t^C \mid}$$

where t denotes time, $g_j(a_j, d_j)$ represents the cost function for competitive power. a_j is the arrival time and d_j is the assigned due date of job j. $\mid J_t^C \mid$ is the number of jobs completed by time t.

(ii) minimize asymptotic late delivery cost per job

$$\lim_{t\to\infty} \frac{\sum_{j\in J_t^C} h_j(c_j, d_j)}{\mid J_t^C \mid}$$

where t denotes time, $h_j(c_j, d_j)$ represents the cost function for late delivery of job j. c_j is the actual completion time and d_j is the assigned due date for job j. $\mid J_t^C \mid$ is the number of jobs completed by time t.

A two-phase solution methodology is proposed to solve the due date assignment problem under consideration which is given in the next chapter. For each phase, an MIP model and a heuristic algorithm are constructed.

4. SOLUTION APPROACH

A due date assignment policy is to be developed for immediate due date quotation and operational efficiency. As a solution methodology, a two-phase decision support mechanism is proposed. A forward-looking family allocation problem is solved in the first phase and families are allocated to schedule before observing actual job arrivals. During this phase, as a proactive plan for operational efficiency, a capacity allocation takes place based on expected workload and interarrival pattern of families. This problem is solved in an offline and periodic manner. The first phase is called "batch allocation phase" and serves as a preparation for the second phase. Then, due date quotation takes place in the second phase in an online fashion for each arriving job, based on the structure obtained in the previous phase. The second phase is called "due date assignment phase".

In order to illustrate the process in the first phase, consider Figure 4.1. As a starting step, time horizon is divided into fixed scheduling periods at the beginning of which the batch allocation problem is solved for two scheduling periods ahead. After one scheduling period, batch allocation problem is revised and solved again to cover two scheduling periods. Due to the consideration of non-complete jobs from previous periods and expected new arrivals during current period, more than one period must be covered and two periods are the minimum number of periods to be covered. By this way, a schedule of one period is always guaranteed.

After having the scheduling periods, a capacity allocation takes place based on expected work congestion and arrivals for each family as it can be seen in Figure 4.2.

While allocating the capacity for families, in addition to expected new arrivals, non-complete jobs with already assigned due dates from previous periods, must also be taken into consideration and rescheduled according to the new family-batch structure

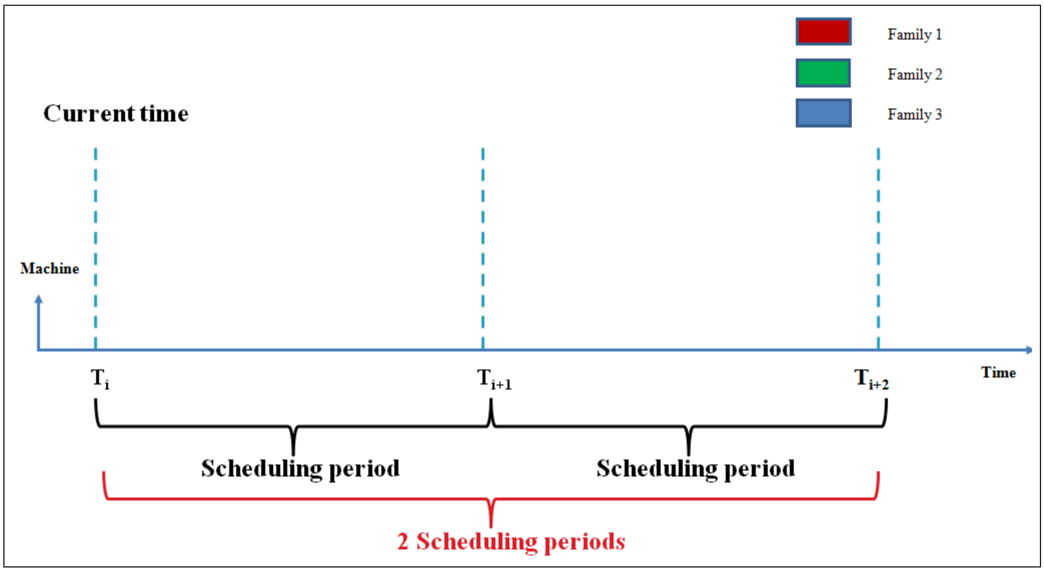


Figure 4.1. Time Frames for Batch Allocation Phase.

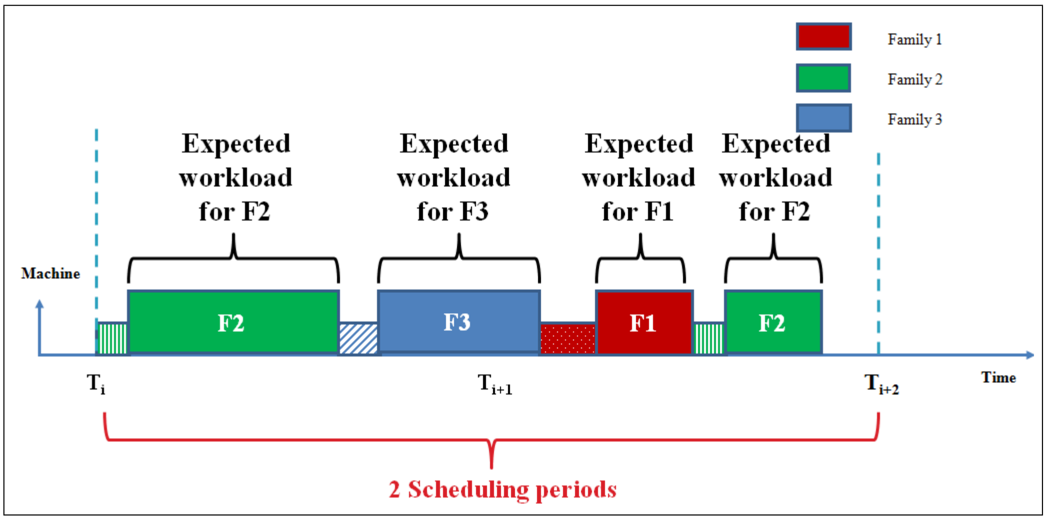


Figure 4.2. Pre-allocation of Families in Batch Allocation Phase.

after solving batch allocation problem, as it can be seen in Figure 4.3.

As the schedule is being organized by loading families in the first phase, a batching procedure is introduced as the major concern of the solution approach. A batch is a group of jobs from the same family which can be processed consecutively by sharing a common setup, based on production similarities. Therefore, for the first phase, since the environment is dynamic and no job related data is known until arrival, a prospective family based batching policy is suggested for potential future jobs.

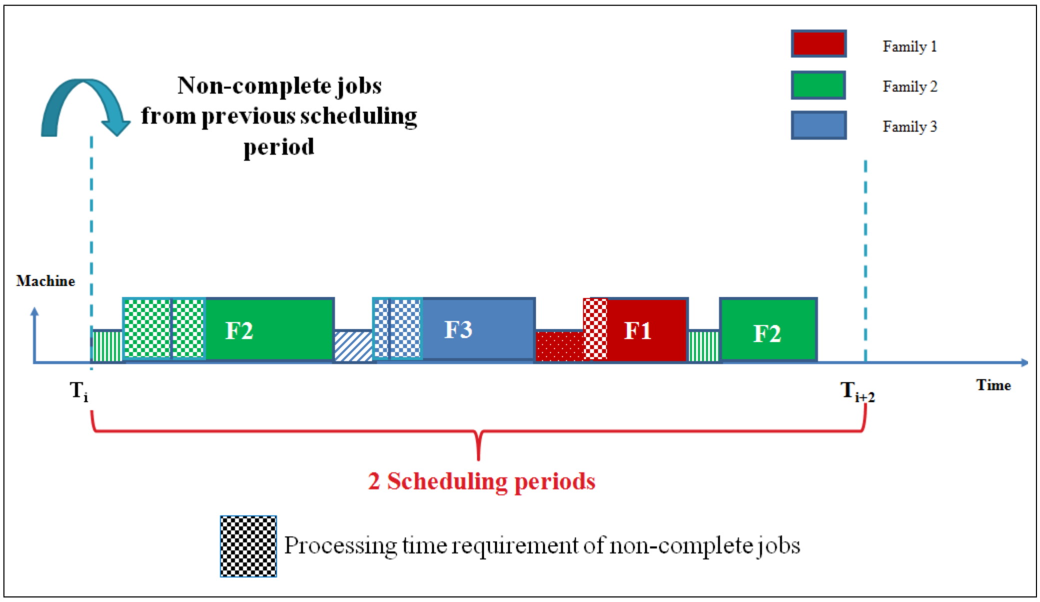


Figure 4.3. Carrying Non-complete Jobs into New Schedule.

The solution is designed in such a way that jobs from each family can be processed in one of the several batches formed for their family, instead of processing the jobs one by one or only within a single block of the schedule allocated for their family. The motivation for applying such a batching strategy is to take the advantage of sharing a single setup for a number of jobs belonging to the same family, to prevent the loss of capacity due to setups and to be able to deliver the completed jobs as early as possible.

In the proposed batching methodology, a family can be split and processed in several batches. Two successive batches cannot belong to the same family. Jobs from different families cannot coexist in a batch. Job (item) availability model is applied, thus a job within a batch can be delivered as soon as it is processed, independent of other jobs inside the batch that it takes place.

While allocating batches into schedule, the trade-off between setup time and due date assignment criteria must be considered simultaneously. Two different batch allocations can be seen in Figure 4.4. On the upper loading scheme, larger batches are allocated into the schedule whereas in the lower loading scheme, batches are smaller in

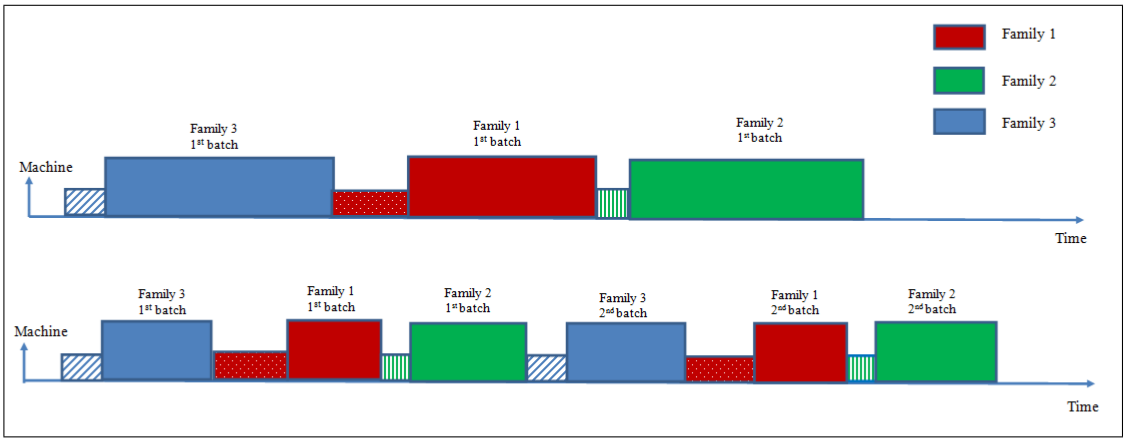


Figure 4.4. Possible Batch Allocations.

size and the number of batches is more.

Generally, large batches have the advantage of low number of setups and high level of efficiency in terms of machine utilization. However, processing a large batch may delay the processing of a job whose family is allocated later in the schedule. On the contrary, producing in smaller batches causes more number of setups and loss of productive efficiency, yet customer satisfaction and service level can be improved since shorter due dates can be quoted.

In order to visualize this trade-off, consider Figure 4.5. Suppose, at time t, a new job i has just arrived which belongs to Family 1. The allocation above is better in terms of setup but worse in terms of due date assignment considerations since the due date assigned for this new job is shorter in the second allocation.

In order to solve the family allocation problem, a mathematical model and a heuristic algorithm are proposed.

After loading batches into the schedule, in the second phase, due date quotation for new arriving jobs takes place. Whenever a new job arrives, a due date is assigned instantly by assigning the job to the first available position, based on the results of the batch allocation structure from the first phase. A sample assignment for three jobs can be observed in Figure 4.6. In that sense, two phases are strongly interrelated. Outputs

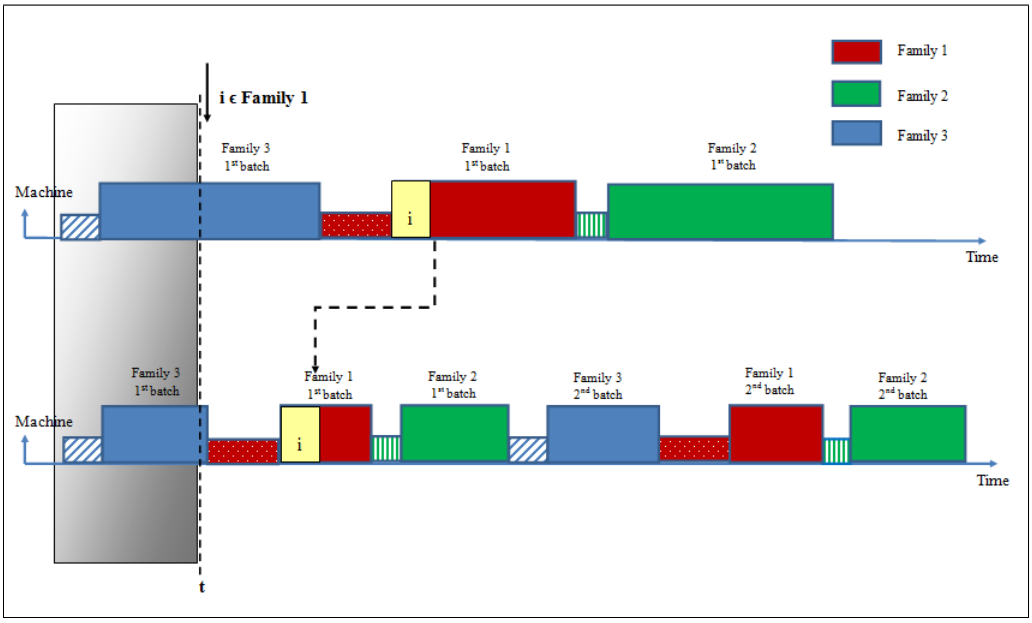


Figure 4.5. Trade-off between Setups and Due Dates.

of the first phase become inputs for the second phase.

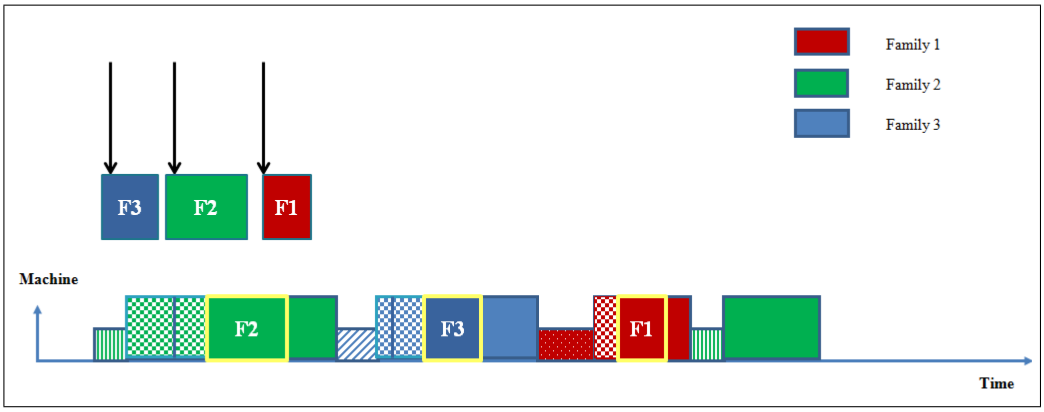


Figure 4.6. Scheduling New Arrivals in Due Date Assignment Phase.

In order to solve the due date assignment problem, a mathematical model and a heuristic algorithm are developed.

The mathematical models are described in the next chapter which is followed by the heuristic algorithms.

5. MATHEMATICAL FORMULATIONS

5.1. Phase 1: Optimization Model for Batch Allocation

The following MIP model constitutes the first phase of the proposed two-phase solution approach. Due to dynamic environment, since no job related data is known until arrival, a forward-looking family allocation is constructed based on expected workload and arrival estimation for families, before observing job arrivals. The family-batch structure obtained in the first phase serves as a guideline for the second phase during due date assignment.

The planning horizon is split into time frames that correspond to scheduling periods. The model aims to solve the batch allocation problem optimally at the beginning of every scheduling period. However, batch allocation model is solved at the beginning of every scheduling period to cover two scheduling periods ahead and solved at the midpoint of that interval again to cover two periods as it can be seen in Figure 5.1. The purpose is always to be able to have an available position inside a batch and assign a due date for a new arrival immediately.

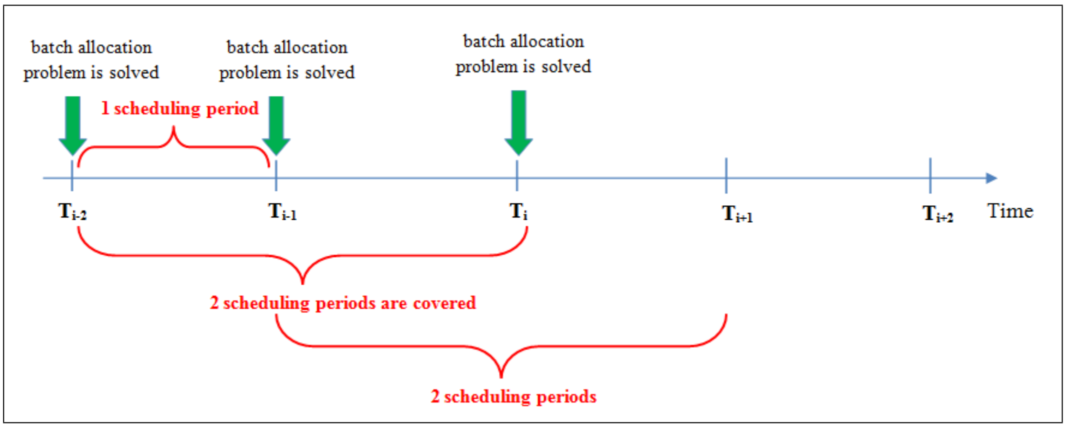


Figure 5.1. Scheduling Period.

In the model, in addition to the new arrivals that are expected during current scheduling period, non-complete jobs whose due dates are assigned in previous period(s), are also taken into consideration and rescheduled according to newly constructed family-batch structure. The family type and processing time of non-complete jobs reveal at their arrival times and their due dates are assigned immediately. Hence, non-complete job related data is already known. Rescheduled completion times of non-complete jobs in the new schedule need to be computed in order to use in tardiness calculations and find the deviations from the assigned due dates.

Batch allocation problem is solved in an offline and periodic manner. While solving it, the current state of the machine is taken into consideration. There are three possible cases: firstly, the machine may be idle (no setup is carried out or no job is being processed), secondly, the machine may be in a setup operation or thirdly, the machine may be processing a job currently. Hence the concept of Earliest Start Time (EST), current batch, current job and current time concepts are introduced into the model. The details are explained in the related constraints.

When the optimization model is solved, the outputs are the number and sequence of opened batches, family-batch assignments, batch sizes in terms of planned start and completion times, rescheduled completion times and tardiness of non-complete jobs.

The sets, indices, parameters, decision variables, constraints and objective functions of the mathematical model to solve batch allocation problem are given as follows.

5.1.1. Sets

$J^{NC}_{T_{begin}}$ is the set of non-complete jobs whose due dates are assigned in previous scheduling periods but not completed by the beginning of current scheduling period and to be rescheduled according to new batch allocation.

5.1.2. Indices

Let b be batch position index, $b = 1, 2, \ldots, B$ (where B is a parameter); f be family index, $f = 1, 2, \ldots, F$ (where F is a parameter); j be (non-complete) job index, $j = 1, 2, \ldots, |J^{NC}_{T_{begin}}|$; k be job position index within each batch , $k = 1, 2, \ldots, K$ (where

K is a parameter); n be the number of opened batches for a family , $n = 1, 2, \ldots, B$; T be scheduling period index, $T = 1, 2, \ldots$ and $T = i \Rightarrow [T_{i-1}, T_i]$ where $T_{i-1} \equiv T_{begin}$; $T_i \equiv T_{end}$ and $T_0 = 0$.

5.1.3. Parameters

B is the maximum number of batches that can be opened within two scheduling periods. F is the number of families. K is the maximum number of positions in a batch. $\mathcal{A}_f$ is the mean of interarrival time distribution of family f and $\mathcal{P}_f$ is the mean of processing time distribution of family f. $\mathcal{S}_f$ is setup time required for family f. d_j is the due date assigned for (non-complete) job j and p_j is processing time of (non-complete) job j. EST represents earliest start time (time that the machine becomes available for processing). $\mathcal{F}(j)$ is the family of job j and $\mathcal{F}(current\_batch)$ is the family of current batch being processed on the machine at EST. SP is the length of scheduling period ($SP = T_{i+1} - T_i \equiv T_{end} - T_{begin}$). M_1 is a sufficiently large positive number and equals ($EST + 2 \cdot SP$) and M_2 is also a sufficiently large positive number that is equal to ($2 \cdot SP$). TC is tardiness coefficient.

5.1.4. Decision Variables

Binary decision variables are as follows. Y_{fb} is a binary decision variable which takes 1, if family f is assigned to batch on position b; 0, otherwise. X_{jkb} is a binary decision variable that takes 1 if job j is assigned to position k in batch on position b; 0, otherwise. G_{fb} is a binary decision variable that takes 1, if family f is assigned to batch on position b which is completed in the second scheduling period; 0, otherwise. H_b is a binary decision variable which takes 1, if batch b is completed in the second scheduling period; 0, otherwise. U_{fn} is a binary decision variable that takes 1, if there are n opened batches for family f; 0, otherwise.

Continuous decision variables are as follows. ST_b is the start time of batch on location b, CT_b is the completion time of batch on location b. PT_b represents expected total processing time requirement of new arriving jobs in current scheduling period to be processed in batch on location b (processing time requirement for non-complete jobs is not included). W_{fb} denotes total work content for family f to be processed within batch on location b. c_j is the completion time of (non-complete) job j in the new schedule after rescheduling and $TARD_j$ is the tardiness of (non-complete) job j. QLT_{fbn} shows expected quoted lead time for family f within batch b, based on the number of opened batches n. ST'_b is the start time of (opened) batch on location b, after the assignment of non-complete jobs. CT'_b is the completion time of (opened) batch on location b. Δ^I_{fb} represents extra amount of capacity allocated for family f assigned to batch on location b, with respect to new arrivals, due to stochasticity of job interarrival and processing times, Δ^{II}_f represents extra amount of capacity allocated for family f, with respect to new arrivals, due to stochasticity of job interarrival and processing times and Δ^{III} represents extra amount of capacity allocated with respect to new arrivals from all families on all locations, due to stochasticity of job interarrival and processing times. ϵ is the slack within two scheduling periods

5.1.5. Mathematical Model for Phase 1

$$\text{Minimize} \qquad Z_0 = \sum_{b=1}^{B}\sum_{f=1}^{F}\Delta^I_{fb} + \sum_{f=1}^{F}\Delta^{II}_f + \Delta^{III} \tag{5.1}$$

$$\text{Minimize} \qquad Z_1 = \sum_{j\in J^{NC}_{T_{begin}}} TARD_j \tag{5.2}$$

$$\text{Minimize} \qquad Z_2 = \sum_{f=1}^{F}\sum_{b=1}^{B}\sum_{n=1}^{B} QLT_{fbn} \tag{5.3}$$

Subject to:

$$\sum_{f=1}^{F} Y_{fb} \leq 1 \qquad \forall b \tag{5.4}$$

$$\sum_{b=1}^{B} Y_{fb} \geq 1 \qquad \forall f \tag{5.5}$$

$$Y_{fb} + Y_{f,b+1} \leq 1 \qquad \forall f, \forall b\backslash B \tag{5.6}$$

$$\sum_{f=1}^{F} Y_{fb} \geq \sum_{f=1}^{F} Y_{f,b+1} \quad \forall b\backslash B \tag{5.7}$$

$$\sum_{j\in J^{NC}_{T_{begin}}} X_{jkb} \leq 1 \qquad \forall k, \forall b \tag{5.8}$$

$$\sum_{k=1}^{K}\sum_{b=1}^{B}X_{jkb} = 1 \qquad \forall j \in J^{NC}_{T_{begin}} \tag{5.9}$$

$$\sum_{j\in J^{NC}_{T_{begin}}} X_{jkb} \geq \sum_{j\in J^{NC}_{T_{begin}}} X_{j,k+1,b} \qquad \forall k\backslash K, \forall b \tag{5.10}$$

$$\sum_{k=1}^{K}X_{jkb} \leq Y_{\mathcal{F}(j),b} \qquad \forall j\in J^{NC}_{T_{begin}}, \forall b \tag{5.11}$$

$$ST_1 \geq EST + \sum_{\substack{f=1\\ f\neq\mathcal{F}(current\_batch)}}^{F} \mathcal{S}_f \cdot Y_{f1} \tag{5.12}$$

$$ST_b \geq CT_{b-1} + \sum_{f=1}^{F}\mathcal{S}_f \cdot Y_{fb} \qquad \forall b\backslash 1 \tag{5.13}$$

$$ST'_b \geq ST_b + \sum_{j\in J^{NC}_{T_{begin}}}\sum_{k=1}^{K}\sum_{b=1}^{B}X_{jkb}\cdot p_j - M_1(1-Y_{fb}) \qquad \forall f, \forall b \tag{5.14}$$

$$ST'_b \leq ST_b + \sum_{j\in J^{NC}_{T_{begin}}}\sum_{k=1}^{K}\sum_{b=1}^{B}X_{jkb}\cdot p_j + M_1(1-Y_{fb}) \qquad \forall f, \forall b \tag{5.15}$$

$$ST'_b \leq M_1\left(\sum_{f=1}^{F}Y_{fb}\right) \qquad \forall b \tag{5.16}$$

$$CT_b = ST_b + \sum_{j\in J^{NC}_{T_{begin}}}\sum_{k=1}^{K}X_{jkb}\cdot p_j + PT_b \qquad \forall b \tag{5.17}$$

$$CT'_b \geq CT_b - M_1(1-Y_{fb}) \qquad \forall f, \forall b \tag{5.18}$$

$$CT'_b \leq CT_b + M_1(1-Y_{fb}) \qquad \forall f, \forall b \tag{5.19}$$

$$CT'_b \leq M_1\left(\sum_{f=1}^{F}Y_{fb}\right) \qquad \forall b \tag{5.20}$$

$$\sum_{l\leq b}W_{fl} + \Delta^{I}_{fb} \geq \left[\left(\frac{CT_b - EST}{\mathcal{A}_f}\right)\cdot\mathcal{P}_f\right] - M_2(1-Y_{fb}) \qquad \forall f, \forall b \tag{5.21}$$

$$\sum_{b=1}^{B}W_{fb} + \Delta^{II}_{f} \geq \left(\frac{SP}{\mathcal{A}_f}\right)\cdot\mathcal{P}_f \qquad \forall f \tag{5.22}$$

$$W_{fb} \geq PT_b - M_2(1-Y_{fb}) \qquad \forall f, \forall b \tag{5.23}$$

$$W_{fb} \leq PT_b + M_2(1-Y_{fb}) \qquad \forall f, \forall b \tag{5.24}$$

$$W_{fb} \leq M_2\left(Y_{fb}\right) \qquad \forall f, \forall b \tag{5.25}$$

$$PT_b \leq M_2\left(\sum_{f=1}^{F}Y_{fb}\right) \qquad \forall b \tag{5.26}$$

$$\left(\sum_{f=1}^{F} Y_{fb}\right) \leq M_2 \left(\sum_{j \in J^{NC}_{T_{begin}}} \sum_{k=1}^{K} X_{jkb} \cdot p_j + PT_b\right) \quad \forall b \tag{5.27}$$

$$\sum_{f=1}^{F}\sum_{b=1}^{B} \mathcal{S}_f \cdot Y_{fb} + \sum_{j \in J^{NC}_{T_{begin}}} \sum_{k=1}^{K}\sum_{b=1}^{B} X_{jkb} \cdot p_j + \sum_{b=1}^{B} PT_b + \epsilon = 2 \cdot SP + \Delta^{III} \tag{5.28}$$

$$c_j \geq ST_b + \sum_{j \in J^{NC}_{T_{begin}}} \sum_{l=1}^{k} X_{jlb} \cdot p_j - M_1(1 - X_{jkb}) \quad \forall j \in J^{NC}_{T_{begin}}, \forall k, \forall b \tag{5.29}$$

$$c_j \leq ST_b + \sum_{j \in J^{NC}_{T_{begin}}} \sum_{l=1}^{k} X_{jlb} \cdot p_j + M_1(1 - X_{jkb}) \quad \forall j \in J^{NC}_{T_{begin}}, \forall k, \forall b \tag{5.30}$$

$$TARD_j \geq c_j - d_j \quad \forall j \in J^{NC}_{T_{begin}} \tag{5.31}$$

$$\sum_{b=1}^{B} G_{fb} \geq 1 \quad \forall f \tag{5.32}$$

$$G_{fb} \leq Y_{fb} \quad \forall f, \forall b \tag{5.33}$$

$$H_b \geq \sum_{f=1}^{F} G_{fb} \quad \forall b \tag{5.34}$$

$$CT_b \geq (EST + SP + 1) - M_1(1 - H_b) \quad \forall b \tag{5.35}$$

$$\sum_{b=1}^{B} Y_{fb} = \sum_{n=1}^{B} n \cdot U_{fn} \quad \forall f \tag{5.36}$$

$$\sum_{n=1}^{B} U_{fn} = 1 \quad \forall f \tag{5.37}$$

$$QLT_{fbn} \geq \frac{(ST'_b + CT'_b)}{n} - 2M_1(2 - U_{fn} - Y_{fb}) \quad \forall f, \forall b, \forall n \tag{5.38}$$

$$CT_b, CT'_b, PT_b, ST_b, ST'_b \geq 0 \quad \forall b \tag{5.39}$$

$$c_j, TARD_j \geq 0 \quad \forall j \in J^{NC}_{T_{begin}} \tag{5.40}$$

$$W_{fb} \geq 0 \quad \forall f, \forall b \tag{5.41}$$

$$\epsilon, \Delta^{III} \geq 0 \tag{5.42}$$

$$QLT_{fbn} \geq 0 \quad \forall f, \forall b, \forall n \tag{5.43}$$

$$\Delta^{II}_f \geq 0 \quad \forall f \tag{5.44}$$

$$G_{fb}, Y_{fb}, \Delta^{I}_{fb} \in \{0,1\} \quad \forall f, \forall b \tag{5.45}$$

$$X_{jkb} \in \{0,1\} \quad \forall j \in J^{NC}_{T_{begin}}, \forall k, \forall b \tag{5.46}$$

$$H_b \in \{0,1\} \quad \forall b \tag{5.47}$$

$$U_{fn} \in \{0,1\} \quad \forall f, \forall n \tag{5.48}$$

There are two major objective functions to be minimized. However, an additional objective function is introduced to prevent any kind of infeasibility, due to the stochastic nature of the problem. This newly introduced objective function (5.1) minimizes amount of extra capacity allocated regarding workload of new arrivals. A necessity may occur due to stochastic job processing times if job processing times deviate from the mean processing time a lot. Firstly, the optimization model is solved under objective (5.1) and then, the following constraint (5.49) is introduced into the model:

$$\sum_{b=1}^{B}\sum_{f=1}^{F}\Delta^{I}_{fb} + \sum_{f=1}^{F}\Delta^{II}_{f} + \Delta^{III} \leq Z^*_0 \tag{5.49}$$

The first major objective function (5.2) minimizes tardiness of non-complete jobs. Secondly, the optimization model is solved under the objective (5.2). Then, the following constraint (5.50) is introduced into the model to reflect an upper bound on total tardiness of non-complete jobs which is defined as a multiplier of the optimal objective function value obtained with the objective function (5.2):

$$\sum_{j \in J^{NC}_{T_{begin}}} TARD_j \leq Z^*_1 \cdot TC \tag{5.50}$$

The second major objective function (5.3) minimizes expected quoted lead time for new arrivals. Finally, the model is solved under objective (5.3) with all constraints.

Constraint (5.4) ensures that each batch can be occupied by at most one family and all batches are not necessarily opened. In the model, the number of batches to be opened is also a decision variable which can be determined at the end by computing: $\sum_{f=1}^{F}\sum_{b=1}^{B} Y_{fb}$ where $\sum_{f=1}^{F}\sum_{b=1}^{B} Y_{fb} \leq B$. Constraint (5.5) states that each family must be assigned to at least one batch and a family can be split into more than one batch. According to constraint (5.6), two consecutive batches cannot be allocated to the same family. Constraint (5.7) guarantees that if a batch is not opened, the next batch(es) following this batch will not be opened either. In other words, batches are sequenced from left to right.

The set of non-complete jobs transferred from previous periods must be rescheduled. The family type and processing time of these jobs are known and their due dates are already assigned before. Constraints (5.8), (5.9), (5.10) and (5.11) define assignment constraints regarding non-complete jobs that are not completed by T_{begin}. According to constraint (5.8), at most one non-complete job can be assigned to a position in a batch. Constraint (5.9) guarantees that each non-complete job is assigned to a position in a batch. Constraint (5.10) ensures that a non-complete job must be assigned to the first available position and if a position is not occupied within a batch, the next position(s) cannot be occupied before that position. In other words, non-complete jobs are assigned from left to right. Constraint (5.11) assures that a non-complete job can be assigned to a position in a batch if and only if the family it belongs is assigned to that batch in the new schedule.

Every time, when batch allocation model is solved, start time of the first batch in optimization model depends on a parameter called Earliest Start Time (EST). EST is the earliest time the machine becomes available if it is busy. As it can be seen in Figure 5.2, there are three possible cases when calculating EST. Firstly, the machine may be idle, in other words, no setup operation is in process or no job is being processed. Then, EST is taken as the current time (the time when the model is solved, equivalently T_{begin}). Secondly, the machine may be under a setup operation. In this case, EST is the completion time of the setup for the current batch. Thirdly, the machine may be processing a job currently, namely current job. In such a situation, EST is equal to the completion time of this current job.

Constraint (5.12) ensures that the start time of the batch on the first location is greater than or equal to Earliest Start Time (EST) plus setup time if necessary. A setup is required if the assigned family for the first batch in optimization model is different than the family of the current batch on the machine. Constraint (5.13) shows the start time of other batches except the batch on the first location. Each is greater than or equal to the completion time of the previous batch plus setup time required for the family assigned to that batch.

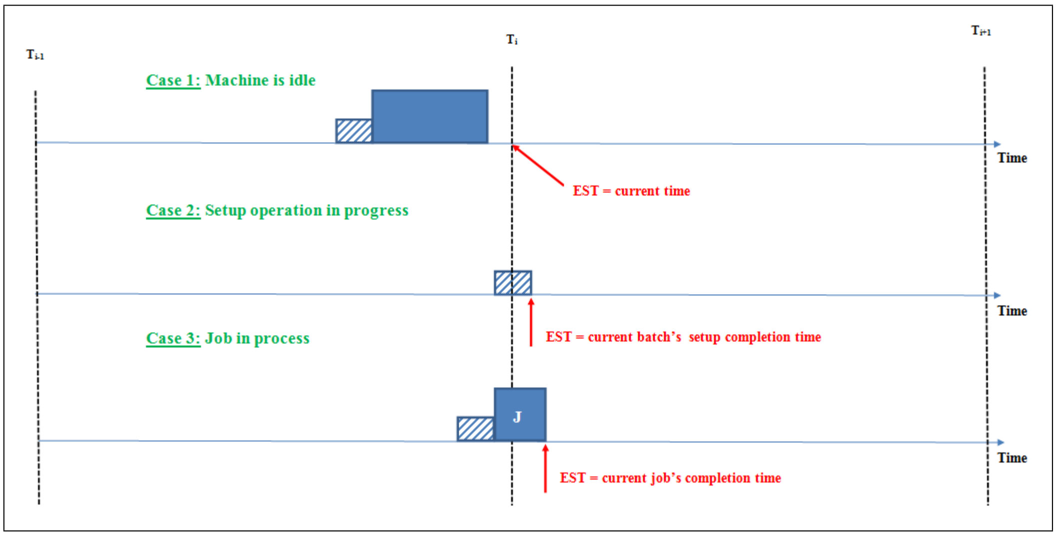


Figure 5.2. EST for Batch Allocation Phase.

In order to minimize expected quoted lead time through families for expected new arrivals, a rough estimation method is developed. The idea is as follows. By forcing the batches to take place on the left-hand-side of the schedule, start and completion times of opened batches can be minimized. This leads to quoting closer due dates, which in turn, can minimize the difference between quoted due dates and arrival times of jobs inside these opened batches. To be used in expected quoted lead time computations, it is necessary to distinguish opened batches and non-opened ones, among maximum available number of batches.

Constraints (5.14) and (5.15) make sure that start time of an opened batch is equal to the sum of the start time and processing time requirement of non-complete jobs assigned to that batch if that batch is opened (equivalently, if a family is assigned to that batch). Otherwise, constraint (5.16) forces the start time of a non-opened batch (a batch to which no family is assigned) to be zero together with the non-negativity constraint (5.39). These start times are then used in the calculation of second objective function (5.3) through constraint (5.38) in order to minimize expected quoted lead time.

Constraint (5.17) defines the completion time of all batches. It is equal to the summation of the start time of the batch, total processing time requirement of non-complete jobs assigned to that batch and expected total processing time of new jobs during current scheduling period to be assigned to that batch. Constraints (5.18) and (5.19) make sure that completion time of an opened batch is equal to the completion time of that batch if that batch is opened. Otherwise, constraint (5.20) forces completion time of a non-opened batch to be zero together with the non-negativity constraint (5.39). Likewise, these completion times are then used in the calculation of objective function (5.3) together with (5.38) so as to minimize expected quoted lead time.

Constraint (5.21) ensures that total work content of a particular family up to the completion time of a batch is satisfied through the work content of all batches assigned to that family, including the batch itself. Constraint (5.22) declares that, for each family, total work content within a scheduling period is met. Constraints (5.23) and (5.24) provide that work content for new arrivals from a family assigned to a batch is equal to the expected processing time requirement of that batch for new arrivals.

Constraint (5.25) guarantees that if a family is not assigned to a batch, no work content can be assigned to that batch. Similarly, constraint (5.26) ensures that if a batch is not opened (if no family is assigned to it), then no processing time can be assigned to that batch. On the other hand, if a family is assigned to a batch (if the batch is opened), then the assigned processing time takes a non-negative value.

Constraint (5.27) provides that if sum of processing time requirement for non-complete jobs and new arriving jobs to be processed in a batch is zero, then that batch cannot be opened (no family can be assigned to it). Constraint (5.28) states that the sum of setup times and total processing time assigned for non-complete jobs and allocated for new arrivals and slack (if exists) must be equal to the twice of the scheduling period plus extra capacity (if required).

Completion times of non-complete jobs must be computed according to their assignments in the new schedule. Constraints (5.29) and (5.30) calculate completion time of a non-complete job after rescheduling, according to the new schedule. Constraint (5.31) computes tardiness of non-complete jobs using rescheduled completion times obtained from constraints (5.29) and (5.30) and already assigned due dates of these non-complete jobs. Since tardiness of job j can be written as: $TARD_j = Max(0, L_j) = Max(0, c_j - d_j)$, constraints (5.31) and (5.40) together linearize tardiness of job j. Then, using these tardiness values, the objective function (5.2) minimizes the sum of tardiness of all non-complete jobs.

Quoted lead time estimation methodology must be clarified before explaining the next set of constraints and the related objective function. Without the following constraints, the number of opened batches is prone to be kept very limited and take place on the left-hand-side (mostly in the 1^{st} scheduling period) of the schedule to keep the difference between the assigned due date and arrival time of jobs as short as possible. Moreover, intuitively, if more batches could be opened, shorter due dates could be assigned accordingly on the average. Thus, the following constraints are introduced into the model.

To be able to have an available position in the 2^{nd} scheduling period, constraints (5.32), (5.33), (5.34) and (5.35) together guarantee that a family is certainly assigned to a batch which has its completion in the 2^{nd} scheduling period. This does not mean that the families will not be assigned to batches which have their completion times in the 1^{st} scheduling period. If families are already assigned to such batches, in addition to those batches, they must be assigned to at least one batch which is completed in the 2^{nd} scheduling period.

Constraint (5.32) ensures that each family must be assigned to at least one batch whose completion time takes place in the 2^{nd} scheduling period. Hence, in the 2^{nd} scheduling period, some capacity is allocated for each family. Constraint (5.33) states that a family can be assigned to a batch which is completed in the 2^{nd} scheduling period if that family is assigned to that batch. Constraint (5.34) assures that if there is a family assigned to a batch which is completed in the 2^{nd} scheduling period, then that batch's completion takes place in the 2^{nd} scheduling period. Constraint (5.35) guarantees that if a batch is completed in 2^{nd} scheduling period, then its completion time must be at least one time unit greater than the end of 1^{st} scheduling period.

Before explaining the final set of constraints, in order to understand the logic behind quoted lead time estimation, consider the illustration in Figure 5.3. The average due date can be roughly estimated as the average of the start time of an opened batch after the assignment of non-complete jobs and completion time of a batch. A similar way is applied to compute expected quoted lead times since quoted lead times depend on assigned due dates.

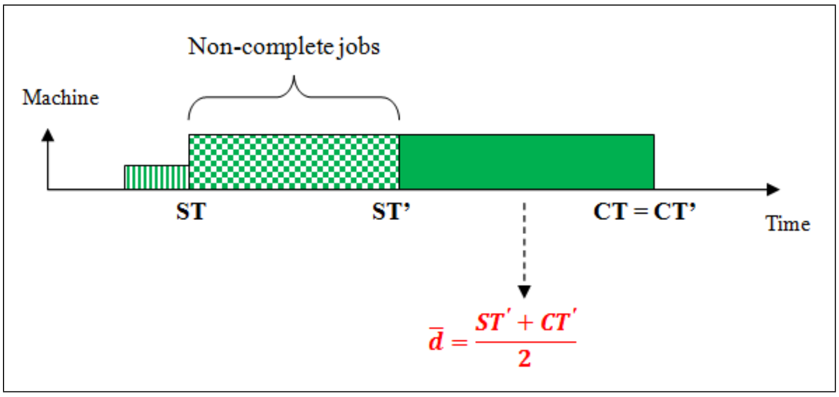


Figure 5.3. Average Due Date.

Constraints (5.36), (5.37) and (5.38) are introduced to prevent the model to open a single batch for a family and to motivate opening more than one batch for families so as to minimize expected quoted lead time. By this way, every job is expected to find a closer position in potentially opened closer batches, compared to the possibility of being assigned to a single further batch in the schedule. Constraint (5.36) counts the number of opened batches for each family through family-batch allocations. Constraint (5.37) states that the number of opened batches can take a single value (n) for each family f. Constraint (5.38) forms a function that takes the start and completion times of an opened batch and divides this sum by the number of opened batches if family f is assigned to batch b among n opened batches. This constraint is then used in the objective function (5.3).

The remaining are non-negativity and binary restrictions.

5.2. Phase 2: Optimization Model for Due Date Assignment

Whenever a new job i arrives, a due date d_i has to be assigned immediately upon arrival. Let t be the arrival time of the new job i. Then, t is the point where the due date assignment takes place.

Due date assignment model takes some of its inputs from the outputs of batch allocation model, namely the family-batch pairs (Y_{fb}), the number of opened batches $BB = \sum_{f=1}^{F}\sum_{b=1}^{B} Y_{fb} \leq B$, planned start and completion times of the batches (ST_b) and (CT_b). The processing time p_j and assigned due date d_j of non-complete jobs are also revealed at their arrival time, hence known in this phase. The relationship between the first and the second phase, regarding their inputs and outputs, is illustrated in Figure 5.4.

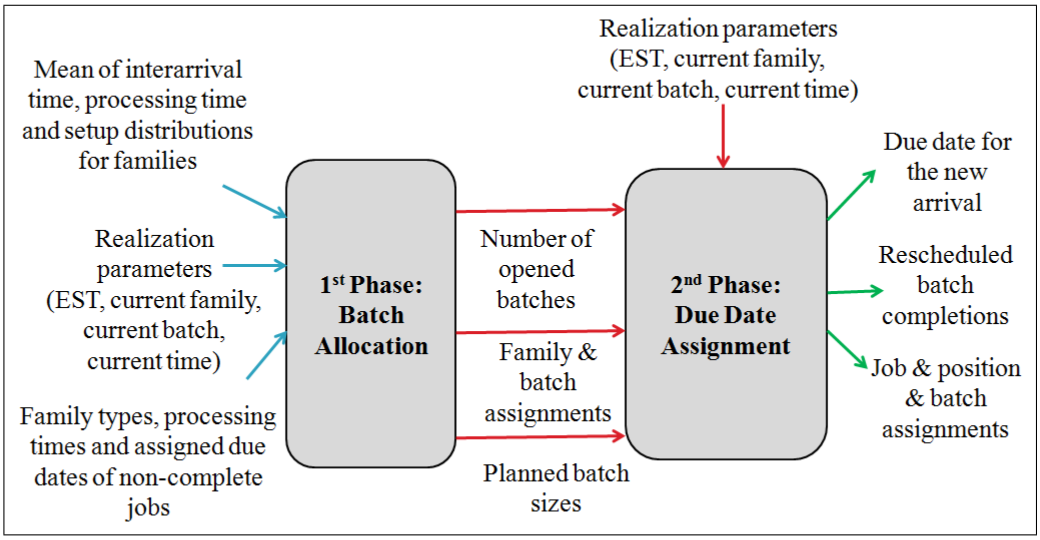


Figure 5.4. Relationship between the First Phase and the Second Phase in terms of Inputs and Outputs.

At every t, that is, at each new job arrival, the mathematical model is to be solved for the due date assignment process.

5.2.1. Sets

There are two sets to be considered. J_t^{NC} is the set of jobs whose due dates are assigned but not completed by time t. J_t^S is the set of jobs scheduled before and at time t where $J_t^S = J_t^{NC} \cup \{i\}$.

5.2.2. Indices

Let b be batch location index, $b = 1, 2, \ldots, BB$ (where BB is a parameter); f be family index, $f = 1, 2, \ldots, F$; i be the job that arrives exactly at time t where $i = J_t^S \setminus J_t^{NC}$; j be job index, $j = 1, 2, \ldots, |J_t^S|$ and k be job position index within each batch, $k = 1, 2, \ldots, K$.

5.2.3. Parameters

BB is the number of opened batches in 1^{st} (batch allocation) phase. a_i is the arrival time of new job i. d_j is the assigned due date for non-complete job j and p_j is the processing time of non-complete job j. EST represents earliest start time (time that the machine becomes available for processing). $\mathcal{F}(j)$ is the family that job j belongs. ST_b is the start time of batch on location b and CT_b is the completion time of batch on location b. M_1 is a sufficiently large positive number and equals $(EST + 2 \cdot SP)$. SP is the length of scheduling period. $\mathcal{S}_f$ is the setup time required for family f. Y_{fb} denotes whether family f is assigned to batch on position b or not.

5.2.4. Decision Variables

X_{jkb} is a binary decision variable that takes 1, if job j is assigned to position k in batch on location b; 0, otherwise. In terms of continuous decision variables, C_{kb} is the completion time of job on position k in batch on location b. c_j is the completion time of job j. d_i is the assigned due date for new job i. RCT_b shows rescheduled completion time of batch on location b in the new schedule. $TARD_j$ is the tardiness of non-complete job j.

5.2.5. Mathematical Model for Phase 2

$$\text{Minimize} \qquad Z_1 = \sum_{j \in J_t^{NC}} TARD_j \tag{5.51}$$

$$\text{Minimize} \qquad Z_2 = d_i \tag{5.52}$$

Subject to:

$$\sum_{j \in J_t^S} X_{jkb} \leq 1 \qquad \forall k, \forall b \tag{5.53}$$

$$\sum_{k=1}^{K} \sum_{b=1}^{BB} X_{jkb} = 1 \qquad \forall j \in J_t^S \tag{5.54}$$

$$\sum_{j \in J_t^S} X_{jkb} \geq \sum_{j \in J_t^S} X_{j,k+1,b} \qquad \forall k \backslash K, \forall b \tag{5.55}$$

$$\sum_{k=1}^{K} X_{jkb} \leq Y_{\mathcal{F}(j),b} \qquad \forall j \in J_t^S, \forall b \tag{5.56}$$

$$C_{kb} \geq EST + \sum_{j \in J_t^S} \sum_{l=1}^{k} X_{jlb} \cdot p_j \qquad \forall k, \forall b \tag{5.57}$$

$$C_{kb} \geq ST_b + \sum_{j \in J_t^S} \sum_{l=1}^{k} X_{jlb} \cdot p_j \qquad \forall k, \forall b \tag{5.58}$$

$$C_{kb} \geq RCT_{b-1} + \sum_{f=1}^{F} \mathcal{S}_f \cdot Y_{fb} + \sum_{j \in J_t^S} \sum_{l=1}^{k} X_{jlb} \cdot p_j \qquad \forall k, \forall b \tag{5.59}$$

$$c_j \geq C_{kb} - M_1(1 - X_{jkb}) \qquad \forall j \in J_t^S, \forall k, \forall b \tag{5.60}$$

$$c_j \leq C_{kb} + M_1(1 - X_{jkb}) \qquad \forall j \in J_t^S, \forall k, \forall b \tag{5.61}$$

$$TARD_j \geq c_j - d_j \qquad \forall j \in J_t^{NC} \tag{5.62}$$

$$RCT_b \geq CT_b \qquad \forall b \tag{5.63}$$

$$RCT_b \geq C_{kb} \qquad \forall k, \forall b \tag{5.64}$$

$$d_i \geq c_i \qquad \text{for } i = \left\{ i \in J_t^S : a_i = t \right\} \tag{5.65}$$

$$X_{jkb} \in \{0, 1\} \qquad \forall j \in J_t^S, \forall k, \forall b \tag{5.66}$$

$$C_{kb} \geq 0 \qquad \forall k, \forall b \tag{5.67}$$

$$c_j \geq 0 \qquad \forall j \in J_t^S \tag{5.68}$$

$$TARD_j \geq 0 \qquad \forall j \in J_t^{NC} \tag{5.69}$$

$$RCT_b \geq 0 \quad \forall b \tag{5.70}$$

$$d_i \geq 0 \qquad \text{for } i = \left\{ i \in J_t^S : a_i = t \right\} \tag{5.71}$$

There are two objective functions. The first objective function (5.51) minimizes tardiness of non-complete jobs. The second objective function (5.52) minimizes the due date assigned for the new arrival. The optimization model is first solved under only (5.51) and then, the following constraint (5.72) is introduced into the model to reflect an upper bound for the total tardiness of non-complete jobs, based on the objective function value obtained with (5.51):

$$\sum_{j \in J_t^{NC}} TARD_j \leq Z_1^* \cdot TC \tag{5.72}$$

Finally, the model is solved under the objective (5.52) with all constraints.

Constraint (5.53) ensures that each position in every batch can be occupied by at most one job from the set J_t^S. Constraint (5.54) states that each job in the set J_t^S is assigned to only one position in a batch. Constraint (5.55) guarantees that jobs are assigned from left to right. Constraint (5.56) assures that a job can be assigned to a position in a batch if and only if the family it belongs is assigned to that batch.

Constraints (5.57), (5.58) and (5.59) define completion times of the jobs through assigned positions in batches. The completion time of a job assigned to a position within a batch will be greater than or equal to the maximum of these three constraints. In (5.57), the start time of a particular batch is greater than or equal to EST plus total processing time of all jobs assigned to that batch, including the job itself. In (5.58), the start time of a batch is greater than or equal to the planned start time (which is an output of the batch allocation model) plus total processing time of all jobs assigned to that batch, including the job itself. In (5.59), the start time is computed by summing the rescheduled completion time of the preceding batch in the new schedule (which is a decision variable of due date assignment model) and the setup time required for the family assigned to that batch (which is an output of batch allocation phase) plus total processing time of all jobs assigned to that batch, including the job itself.

Likewise the case in batch allocation model, EST is a realization parameter about the status of the machine at time t. However, in due date assignment phase, there are six possible cases as illustrated in Figure 5.5.

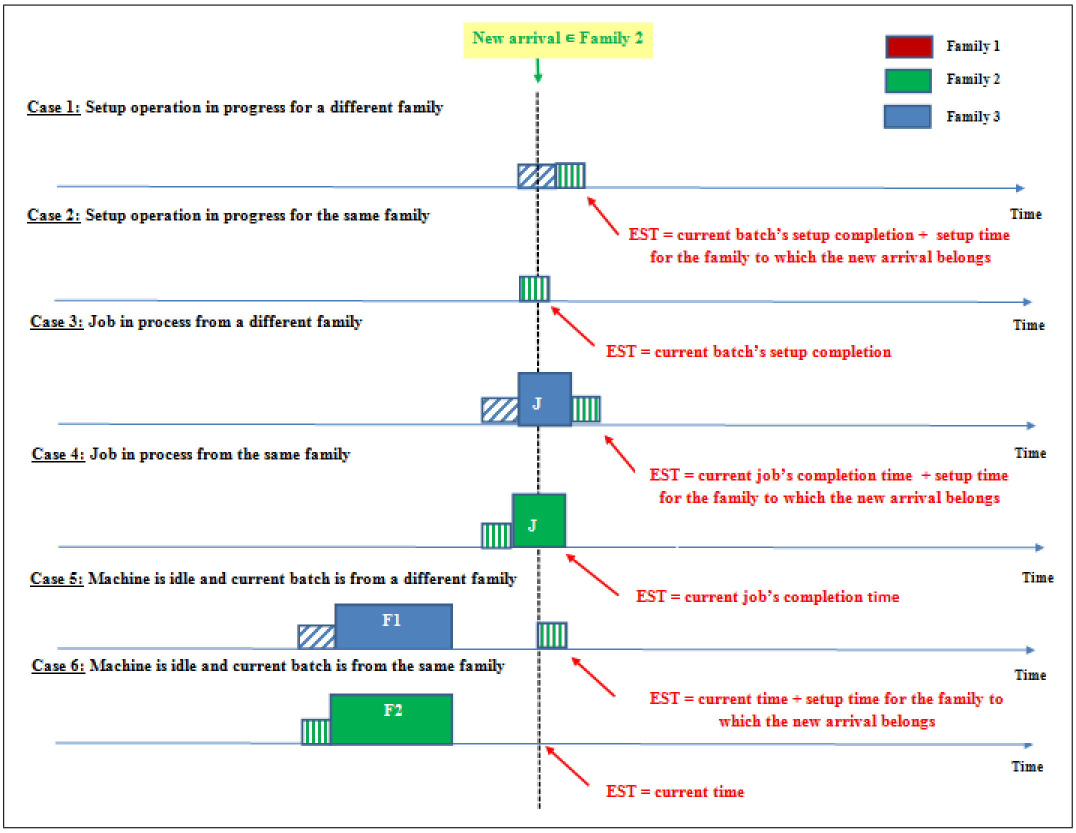


Figure 5.5. EST for Due Date Assignment Phase.

According to Figure 5.5, as a first case, there may be a setup being incurred under current batch for a different family than the family of the new arrival. Then, next to this setup, a setup must incur for the family that the new arrival belongs. In this case, EST is the last setup completion. As a second case, there may be a setup being incurred under current batch for the same family of the new arrival. Then, EST is the setup completion of the current batch. As a third case, there may be a job being processed within current batch from a different family, other than the family of the new arrival. Then, after the completion time of this current job, a setup must incur for the family that the new arrival belongs. In this case, EST is the setup completion.

As a fourth case, there may be a job being processed under current batch from the same family of the new arrival. Then, EST is the completion time of the current job. As a fifth case, the machine may be idle but the family of the current batch may be different than the family of the new arrival. Then, a setup must incur for the family that the new arrival belongs. In this case, EST is the setup completion. As a sixth case, the machine may be idle but the family of the current batch may be the same as the family of the new arrival. Then, EST is the current time.

Constraints (5.60) and (5.61) together state that the completion time of job j is equal to the completion time of the job on position k in batch b if and only if the job is assigned to position k in batch on location b. Constraint (5.62) defines the tardiness of the non-complete jobs in the set J_t^{NC}. Constraints (5.63) and (5.64) express the rescheduled completion time of batch on location b in the new schedule. Constraint (5.65) regulates the due date assigned to a new job i over its expected completion time.

The remaining are the non-negativity and binary restrictions.

6. HEURISTIC ALGORITHMS

The proposed heuristic algorithms are utilized while deciding on the length of the warm-up period due to their fastness in computing time, when compared to the time required by solving the optimization models. Moreover, effective machine utilization parameters are determined by the help of these heuristics. These also provide an upper bound for the proposed optimization models by means of performance comparison.

6.1. Phase 1: Batch Allocation Heuristic

The most important contribution of batch allocation heuristic is that it provides an initial feasible solution for the batch allocation optimization model. The model initially takes the outputs of this heuristic algorithm and improves the solution proposed by the heuristic. By this way, the time to find an initial solution for batch allocation model is removed. The outputs of batch allocation heuristic algorithm are listed below:

- Number of opened batches
- Family sequence for opened batches
- Non-complete job assignments

In this heuristic, the following properties hold in terms of number of opened batches, family sequence and non-complete job assignments:

- Exactly two batches are opened for each family within two scheduling periods. Total number of opened batches is equal to $2F$. Therefore, batches on locations $\{2F+1, 2F+2, \ldots, B\}$ are not opened.
- Families in the 1^{st} scheduling period are sequenced as follows. Instead of a fully cyclic pattern, the families are sequenced in the schedule by shifting the sequence of every family to the next sequence every time the batch allocation heuristic is solved. The sequence for all cases can be seen in Table 6.1.

Table 6.1. Family Sequence in Batch Allocation Heuristic.

F	Number of times heuristic is solved	Family sequence
2	$1, 3, 5, 7, 9, \dots$	$\{F1, F2\}$
	$2, 4, 6, 8, 10, \dots$	$\{F2, F1\}$
3	$1, 4, 7, 10, 13, \dots$	$\{F1, F2, F3\}$
	$2, 5, 8, 11, 14, \dots$	$\{F2, F3, F1\}$
	$3, 6, 9, 12, 15, \dots$	$\{F3, F1, F2\}$
4	$1, 5, 9, 13, 17, \dots$	$\{F1, F2, F3, F4\}$
	$2, 6, 10, 14, 18, \dots$	$\{F2, F3, F4, F1\}$
	$3, 7, 11, 15, 19, \dots$	$\{F3, F4, F1, F2\}$
	$4, 8, 12, 16, 20, \dots$	$\{F4, F1, F2, F3\}$

The logic behind applying such a sequencing principle is not to pile up non-complete jobs from a particular family and to give a chance for each family to take place in the first batch of the schedule respectively.

- The same family sequence is duplicated in the 2^{nd} scheduling period.
- Non-complete jobs from each family are assigned to the first available batch in terms of their family types. There is no restriction regarding batch sizes.
- When the batch allocation heuristic is applied at the beginning of the scheduling period, if the current batch of the machine is the same as the first family of the heuristic, no setup is required at the beginning; otherwise, a setup is required.

6.2. Phase 2: Due Date Assignment Heuristic

A due date is assigned for a new arriving job immediately based on the outputs of the batch allocation problem which is solved in the first phase. Having assigned non-complete jobs in the schedule, a new arrival is assigned to the first available position within opened batches according to the previously solved batch allocation problem in order to assign a due date.

Inside the heuristic, all opened batches during batch allocation phase are scanned respectively to find an available position for a new arrival with respect to its arrival time, family type and processing time requirement. Two different cases are considered when there is an available opened batch for the new arrival in terms of family type and arrival time but the processing time requirement cannot be met within that batch. These two cases can be explained as below:

(i) Batch size remains as it is planned and no shift is allowed (in this case, next possible batches are examined to assign the job)

(ii) Batch size can be extended infinitely as long as family type and arrival time concerns are satisfied (in this case, the job is assigned to that batch)

If there is no available batch in terms of arrival time and processing time requirements within opened batches allocated for the family of the new arrival, a new batch is opened at the end of the schedule, in addition to ones opened in the first phase and the job is assigned to that batch in order to assign a due date.

Related notation:

- i: new arriving job at time t
- a_i: arrival time of job i
- d_i: assigned due date for job i
- t: current time
- BB: number of opened batches in 1^{st} (batch allocation) phase
- b: batch location index, $b = 1, 2, \ldots, BB$
- $\mathcal{F}(i)$: family type of job i
- $\mathcal{F}(b)$: family assigned to batch on location b
- ST_b: planned start time of batch on location b
- CT_b: planned completion time of batch on location b
- |OPENED_BATCHES|: total number of opened batches in first phase for the two scheduling periods ahead
- |EXTRA_BATCHES|: total number of extra batches opened in heuristic in case

of not finding an available position in any of the batches opened in first phase

- |TOTAL_FORMED_BATCHES|=|OPENED_BATCHES| + |EXTRA_BATCHES|

Initialization:

- Let |EXTRA_BATCHES| = 0 .
- Get necessary inputs from the outputs of first (batch allocation) phase such that:
  - (i) |OPENED_BATCHES| = BB
  - (ii) Family and batch assignments, i.e. $\mathcal{F}(b)$ for $b = 1, 2, \ldots,$ |OPENED_BATCHES|
  - (iii) ST_b and CT_b values for $b = 1, 2, \ldots,$ |OPENED_BATCHES|

    (Then, calculate batch sizes by taking the difference between the planned completion and start times of each opened batch)
  - (iv) Non-complete job assignments and rescheduled completion times for non-complete jobs

The pseudocode for due date assignment heuristic is given in Figure 6.1.

```
for all (New arrival i where a_i = t) do
    Scan all batches respectively in terms of family type and arrival time;
    Find the first available batch, say fab;
    if (Processing time requirement of i can be met directly or through shifting) then
        Assign a due date for i, say d_i;
        if (d_i > CT_fab) then
            Shift the planned completion time of fab;
            Update the planned start & completion times of all batches following fab;
            for all (j ∈ J_t^NC) do
                Update rescheduled completion times of non-complete jobs inside the batches following fab;
            end for
        end if
    end if
    if (There is no available batch in terms of arrival time and processing time concerns) then
        if (F(i) = F( |TOTAL_FORMED_BATCHES| )) then
            Expand the last opened batch in the first phase, i.e. |TOTAL_FORMED_BATCHES|;
            Update planned completion time of |TOTAL_FORMED_BATCHES|;
            Assign a due date for i;
        else
            Form a new batch in addition to the ones opened in first phase, say nb;
            Let nb = |TOTAL_FORMED_BATCHES| + 1 ;
            Let F(nb) = F(i) ;
            Assign a due date for i;
            |EXTRA_BATCHES| ++ ;
            |TOTAL_FORMED_BATCHES| ++ ;
        end if
    end if
end for
```

Figure 6.1. Pseudocode for Due Date Assignment Heuristic.

7. EXPERIMENTAL DESIGN

After developing the solution framework, in order to experiment and evaluate the performance of the proposed solution methodology, a simulation is carried out in order to imitate a real single machine shop production environment. There is a mapping from the simulation (real world) to the proposed solutions (plans) as it can be seen in Figure 7.1. A discrete-event simulation environment is constructed. An event-driven approach is adopted. Simulation clock jumps from event to event. The clock always jumps to the event with the minimum event time from the list of scheduled events, so called Future Event List (FEL). The event list is updated as time evolves dynamically. Termination criterion for simulation is a predetermined number of job completions, a part of which is utilized during warm-up period and the rest is used for statistics collection. The details are given in the following sections.

The current state of the machine is taken into consideration every time batch allocation and due date problems are solved. In simulation, jobs are processed on the machine continuously. However, batch allocation problem is solved at the beginning of fixed scheduling periods and due date assignment model is solved in case of a new job arrival.

Three different dispatching policies are constructed for job processing sequence in simulation. Firstly, without considering the proposed solution, jobs are processed according to FCFS rule. Secondly, by taking the proposed solution directly, jobs are processed according to planned family-batch structure in a strict manner. Thirdly, by taking the proposed solution as a guideline, jobs are processed according to planned family-batch structure in a flexible fashion, by allowing modifications on the proposed solution when necessary.

To be able to test the performance in different production environments, several system parameters are utilized in terms of number of families, family setup structure, and production system busyness level.

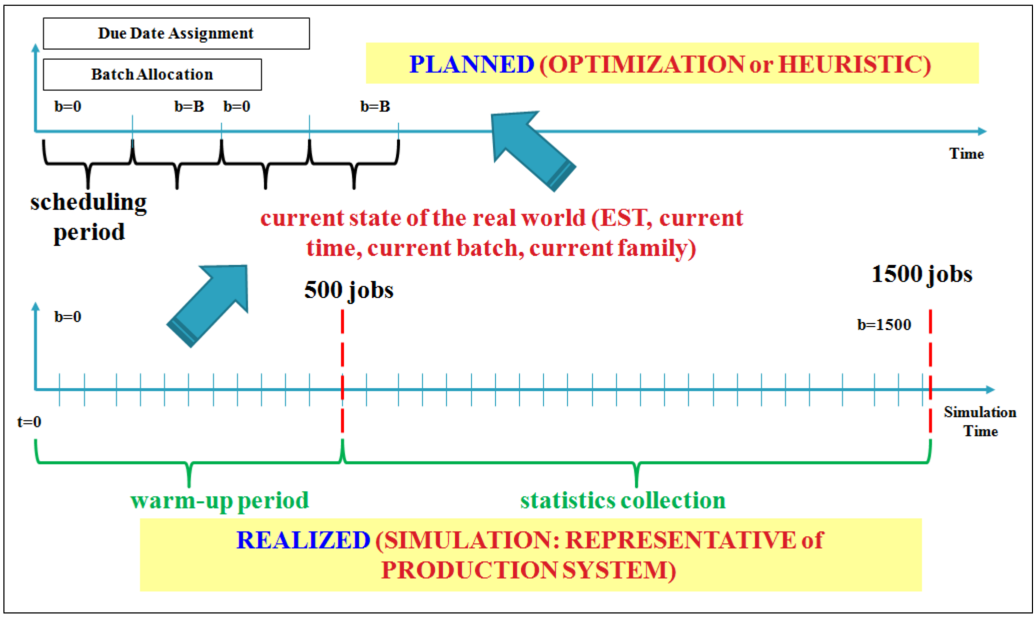


Figure 7.1. Correspondence between Proposed Solution and Simulation.

7.1. Simulation Parameters

Parameters to be determined for simulation are as follows:

- ρ_{eff}: effective machine utilization parameter
- τ: mean job interarrival time
- μ: mean job processing time
- υ: mean family setup time
- θ_f: probability of belonging to family f, $\forall f \in \{1, 2, ...F\}$
- ω: processing time variability parameter
- π: setup time multiplier
- ψ: setup time variability parameter
- ϕ: scheduling period multiplier
- F: number of families
- B: maximum number of batches to be opened in two scheduling periods
- K: number of positions in each batch
- SP: the length of scheduling period
- TC: tardiness coefficient

7.1.1. Deterministic Parameters in Simulation

Number of families (F) to be considered during simulation runs are selected as below:

$$F = \{2, 3, 4\}$$

Batch allocation problem is solved for two scheduling periods ahead. The maximum number of batches that can be opened within two scheduling periods is bounded by a number (B). During batch allocation, a family can be assigned to several batches and all batches are not necessarily opened among these B batches. A preliminary study is conducted to determine the value of B. By letting B be a very large number, the number of opened batches are analyzed for each family. It is observed that, although rarely there are extreme cases, generally the number of opened batches is less than five times the number of families in two scheduling periods. Hence, the following B values are used for each family:

$$B = \{F \cdot 5\}$$

F=2 : $B = 10$

F=3 : $B = 15$

F=4 : $B = 20$

Number of positions in each batch (K) is taken as the maximum of the number of non-complete jobs among all families. K is determined every time when any of the problems (batch allocation or due date assignment) is solved. While determining the number of positions in each batch, in the first phase, there are only non-complete jobs taken into account whereas in the second phase, the new arrival is also considered in addition to non-complete jobs. Let $|J_t^{NC} \in f|$ represent the number of non-complete jobs from family f at time t where t is the time when the problem is solved.

$$For\ \ batch\ \ allocation\ \ phase : K = Max\left\{|J_{T_{begin}}^{NC} \in f|; f = 1, 2, \ldots, F\right\}$$

$$For\ \ due\ \ date\ \ assignment\ \ phase : K = \left[Max\left\{|J_t^{NC} \in f|; f = 1, 2, \ldots, F\right\}\ \ \right] + 1$$

Without loss of generality, the value of the mean job interarrival time (τ) is fixed to 20, in other words, on the average, a job arrival is expected in every 20 time units:

$$\tau = 20$$

The length of scheduling period (SP) is determined by taking multiples of mean interarrival time as $SP = \{\phi \cdot \tau\}$, where ϕ denotes the expected number of job arrivals per scheduling period. Three different lengths of scheduling periods are considered as short, medium and long where the scheduling period multiplier $\phi = \{20, 30, 40\}$, representing $20, 30$ and 40 job arrivals per scheduling period, respectively:

$$SP = \{\phi \cdot \tau\} = \{20\tau, 30\tau, 40\tau\} = \{400, 600, 800\}$$

A single server queue system can be said to be stable if arrival rate is less than service rate, or equivalently, mean service time is less than mean interarrival time. Analogously, a single machine production system can be said to be stable if mean processing time (μ) is less than mean interarrival time (τ), under the absence or ignorance of family setups. In other words, the system is said to be stable if the number of jobs waiting to be processed in the system remains finite. Hence, utilization rate ($\rho = \frac{\mu}{\tau}$) is expected to be less than or equal to 1. Otherwise, queue in front of the machine continuously increases and system may become unstable. However, in the presence of family setups, an effective machine utilization parameter (ρ_{eff}) can be defined which does not indicate the real utilization. The reason can be explained as follows. Setups are non-value-added activities in production and can be expressed as below where (υ) is the mean family setup time, (μ) is the mean job processing time and (τ) is the mean job interarrival time:

$$\rho_{eff} = \frac{\upsilon + \mu}{\tau} \tag{7.1}$$

Since jobs that belong to the same family can be processed in row by forming a batch, the production efficiency can be increased by increasing the number of jobs within batches. Hence, (υ) is an underestimation on the time spent on family setups. Therefore, in the presence of family setups, the system might be stable even if ρ_{eff} is greater than 1. For determining the most appropriate effective utilization parameters, the following set of values are tried where the details of the selection process and final values of utilization parameters are explained in Section 7.5:

$$\rho_{eff} = \{0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9\}$$

Under the setting, since the value of the mean job interarrival time (τ) is fixed to 20, the value of effective machine utilization parameter determines the mean processing time (μ) and the mean setup time (υ), based on Equation (7.1). Therefore, the mean setup time (υ) is defined as a multiple of the mean processing time (μ) as below where (π) is the setup time multiplier:

$$\upsilon = \{\pi \cdot \mu\} \quad \text{where} \quad \pi = \{0.5, 1.5\}$$

Accordingly:

$$\text{if } \pi = \{0.5\} \Rightarrow \rho_{eff} = \frac{0.5\mu + \mu}{\tau} \Rightarrow \mu = \frac{\rho_{eff} \cdot \tau}{1.5} \Rightarrow \upsilon = (0.5) \cdot \left(\frac{\rho_{eff} \cdot \tau}{1.5}\right)$$
$$\text{if } \pi = \{1.5\} \Rightarrow \rho_{eff} = \frac{1.5\mu + \mu}{\tau} \Rightarrow \mu = \frac{\rho_{eff} \cdot \tau}{2.5} \Rightarrow \upsilon = (1.5) \cdot \left(\frac{\rho_{eff} \cdot \tau}{2.5}\right)$$

Based on different effective machine utilization parameters, mean job processing time and mean family setup time values can be computed as in Table 7.1. Also, lower and upper bounds on job processing time and family setup time distributions are provided for the interval where they are drawn during simulation.

7.1.2. Stochastic Parameters in Simulation

Each family is equally likely to be picked up for a job with ($1/F$) probability. In this case, $\theta_f = \frac{1}{F}$ where $f \in \{1, 2, ...F\}$.

$$Family \;\; types \sim Discrete \;\; Uniform\,[1, F]$$

Interarrival time for the j^{th} job is the time between the arrival of two consecutive jobs, say $(j-1)^{st}$ and j^{th} job; in other words, it is the time elapsed until the next arrival. It is assumed that jobs arrive to production system according to *Poisson* process with arrival rate (λ) where $\lambda = \frac{1}{\tau}$ and τ is the mean interarrival time. Therefore, the distribution of job interarrival times are *Exponential* with rate (λ), or mean (τ).

$$Job \;\; interarrival \;\; times \sim \;\; Exponential \;\left(\lambda = \frac{1}{20}\right)$$

Table 7.1. Mean Job Processing Times and Mean Family Setup Times.

ρ_{eff}	π	μ	Lower bound (job (processing times)	Upper bound (job (processing times)	υ	Lower bound (family (setup times)	Upper bound (family (setup times)
0.8	0.5	11	5	17	6	4	8
0.9	0.5	12	6	18	6	4	8
1.0	0.5	13	6	20	7	5	9
1.1	0.5	15	7	23	8	6	10
1.2	0.5	16	8	24	8	6	10
1.3	0.5	17	8	26	9	7	11
1.4	0.5	19	9	29	10	8	12
1.5	0.5	20	10	30	10	8	12
1.6	0.5	21	10	32	11	8	14
1.7	0.5	23	11	35	12	9	15
1.8	0.5	24	12	36	12	9	15
1.9	0.5	25	12	38	13	10	16
0.8	1.5	6	3	9	9	7	11
0.9	1.5	7	3	11	11	8	14
1.0	1.5	8	4	12	12	9	15
1.1	1.5	9	4	14	14	11	17
1.2	1.5	10	5	15	15	12	18
1.3	1.5	10	5	15	15	12	18
1.4	1.5	11	5	17	17	13	21
1.5	1.5	12	6	18	18	14	22
1.6	1.5	13	6	20	20	16	24
1.7	1.5	14	7	21	21	16	26
1.8	1.5	14	7	21	21	16	26
1.9	1.5	15	7	23	23	18	28

Inverse transform technique is used for generating job interarrival times, as explained in [97]. In this method, cumulative distribution function (CDF) of uniform and exponential distributions are equated and solved for X where X is the desired exponentially distributed random variable and U is a random number, $U \sim Uniform(0,1)$:

$$\begin{aligned} u &= 1 - e^{\frac{-x}{\tau}} \\ e^{\frac{-x}{\tau}} &= 1 - u \\ \ln(e^{\frac{-x}{\tau}}) &= \ln(1-u) \\ \frac{-x}{\tau} &= \ln(1-u) \\ x &= -\tau \ln(1-u) \end{aligned} \tag{7.2}$$

Job processing times are random integers. During simulation, processing time of jobs is drawn from a discrete uniform distribution with mean processing time being equal to (μ) where (ω) is the processing time variability parameter and $\omega = \{0.5\}$:

$$Job\ \ processing\ \ times \sim\ Discrete\ \ Uniform\ [\mu \cdot (1-\omega), \mu \cdot (1+\omega)]$$

$$\Rightarrow Job\ \ processing\ \ times \sim\ Discrete\ \ Uniform\ [(0.5)\mu, (1.5)\mu]$$

Family setup times are also random integers and in simulation, they are also drawn from a discrete uniform distribution with mean setup time being equal to (υ) where (ψ) is the setup time variability parameter and $\psi = \{0.2\}$:

$$Family\ \ setup\ \ times \sim\ Discrete\ \ Uniform\ [\upsilon \cdot (1-\psi), \upsilon \cdot (1+\psi)]$$

$$\Rightarrow Family\ \ setup\ \ times \sim\ Discrete\ \ Uniform\ [(0.8)\upsilon, (1.2)\upsilon]$$

7.2. Parameters in Optimization Models

Family-related parameters to be decided in optimization models in Chapter 5 are as follows:

- $\mathcal{S}_f$: Expected setup time requirement for family f
- $\mathcal{P}_f$: Expected processing time for family f
- $\mathcal{A}_f$: Expected interarrival time for family f

While solving the optimization models, the following assumptions hold regarding family attributes such that:

- Interarrival times of all jobs from the same family are assumed to be identical. Since job interarrival times are independent and identically distributed exponential random variables with mean ($\tau = 20$) and each family is equally likely to be selected for a new arrival, then the minimum job interarrival time among all families also follows exponential distribution with mean ($\tau \cdot F$). That is, a new job is expected to arrive from a particular family in every ($\tau \cdot F$) time units, on the average. The proof is given in [98].
- Unit processing times of jobs that belong to the same family are assumed to be identical.

Based on the assumptions and expected values of distributions mentioned in Section 7.1.2, family-related parameters are taken as follows and utilized as if deterministic in the optimization models:

$$\mathcal{S}_f = \upsilon \quad \forall f \in \{1, 2, ...F\}$$

$$\mathcal{P}_f = \mu \quad \forall f \in \{1, 2, ...F\}$$

$$\mathcal{A}_f = \tau \cdot F \quad \forall f \in \{1, 2, ...F\}$$

where υ is the mean family setup time, μ is the mean job processing time, τ is the mean job interarrival time and F is the number of families.

The tardiness coefficient (TC) values used in Chapter 5 are taken as multiples of the optimum objective function value which is obtained by solving the batch allocation model so as to minimize tardiness of non-complete jobs. Then, multiplying it with (TC), two different upper bounds are imposed on the maximum allowable tardiness for non-complete jobs. The tardiness coefficient takes two different values to reflect tight and less tighter upper bounds on tardiness of non-complete jobs:

$$TC = \{1, 1.5\}$$

7.3. Dispatching Policies for Simulation

In terms of dispatching policies for processing sequence of jobs, two major policies are considered:

(i) Dispatching Policy 1: Jobs are processed according to FCFS sequencing principle, regardless of the proposed solution methodology. In this policy, a sequence independent setup incurs if two consecutive jobs do not belong to the same family. No batching takes place unless two consecutive jobs belong to the same family. The number of setups can be at most equal to the number of jobs processed if no two jobs that arrive in succession belong to the same family.

(ii) Dispatching Policy 2: Jobs are processed in batches, based on the outputs of batch allocation problem. This policy is divided into two versions. In the first version, the batch structure after solving batch allocation problem is directly taken and cannot be changed whereas in the second version, batch structure can be modified slightly if necessary.

7.3.1. Dispatching Policy 1: No Batching

According to this policy, jobs are processed based on their arrival times with FCFS sequencing rule. No batching takes place but consecutive jobs can share a common setup if they belong to the same family.

Related notation:

- J_t^{NC}: the set of non-complete jobs by time t
- $|J_t^{NC}|$: the number of non-complete jobs by time t
- j: non-complete job index, $j = 1, 2, \ldots, |J_t^{NC}|$
- d_j: assigned due date for job j
- *current_batch*: current batch being processed in simulation
- *current_job*: current job being processed in simulation
- *current_time*: current clock of simulation, denoted by t
- $\mathcal{F}(b)$: family assigned to batch on location b according to batch allocation model

- $\mathcal{F}(j)$: family type of job j
- $maxJobs$: number of job completions required to terminate simulation
- SP: scheduling period

Under this policy, there are four event types to be considered for discrete-event simulation with the following priority order:

(i) SOH: Start of Horizon
(ii) JA: Job Arrival
(iii) JC: Job Completion
(iv) SC: Setup Completion

Initialization:

- Let current_time = 0.
- Let the machine be idle at time 0, i.e. no setup is under operation or no job is being processed.
- Specify maxJobs, i.e. the number of job completions to terminate simulation.
- Create a FEL to collect all events with their event types and times.
- Create all job arrivals and add these to FEL as JA events by assigning arrival times as event times.
- Create the first SOH event to solve batch allocation model at time zero and add this event to FEL with event time being equal to 0.

The pseudocode for Dispatching Policy 1 is given in Figure 7.2.

```
while (No of jobs processed after warm-up ≤ maxJobs) do
  Find the imminent event with minimum occurrence time in FEL, say evim & Remove evim from FEL;
  Set current_time = event time of evim;
  if (Event type of evim = SOH) then
    Solve 1^st phase (batch allocation) via optimization or heuristic;
    Create a new SOH event with event time= current_time + SP and add this event to FEL;
  end if
  if (Event type of evim = JA) then
    Add the new arriving job (say i) to J_t^NC;
    Assign a due date for i (d_i) in 2^nd phase (due date assignment) via optimization or heuristic;
    if (Machine is idle) then
      if (F(current_batch) = F(i)) then
        Start processing job i & Remove i from J_t^NC;
        Create a new JC event with event time = realized completion time of i;
      else
        Start a setup operation for the family that job i belongs & Let F(current_batch) = F(i);
        Create a new SC event with event time = realized setup completion of current_batch;
      end if
    end if
  end if
  if (Event type of evim = JC) then
    if (|J_t^NC|=0) then
      Let the machine be idle;
    else
      Find the job with minimum arrival time from J_t^NC, say jb;
      if (F(jb) = F(current_job)) then
        Start processing job jb in simulation & Remove jb from J_t^NC;
        Create a new JC event with event time = realized completion time of jb;
      else
        Start a setup operation for the family that job jb belongs & Let F(current_batch) = F(jb);
        Create a new SC event with event time = realized setup completion of current_batch;
      end if
    end if
  end if
  if (Event type of evim = SC) then
    if (|J_t^NC| ≠ 0) then
      Find the job with minimum arrival time from J_t^NC, say jb;
      if (F(j) = F(current_job) ) then
        Start processing job jb & Remove jb from J_t^NC;
        Create a new JC event with event time = realized completion time of jb;
      else
        Start a setup operation for the family that job jb belongs & Let F(current_batch) = F(jb);
        Create a new SC event with event time = realized setup completion of current_batch;
      end if
    end if
  end if
end while
```

Figure 7.2. Pseudocode for Dispatching Policy 1.

7.3.2. Dispatching Policy 2: Batching

Under this policy, the batching structure obtained after solving batch allocation problem is taken into account as a guideline. Instead of having a setup for every new arriving job, realized setup starts in simulation incur based on the planned setup start times obtained in the first phase. By this way, until the start time of next setup, there is the chance of processing more than one job belonging to the family being processed currently. Using this approach, the number of setups is expected to decrease as compared to the previous policy, through sharing a common setup. There are two different versions of this policy:

(i) Dispatching Policy (2.1): Planned setup starts cannot be changed anyway and realized setup starts take place based on the planned outputs. If still, there is a job from the current family being processed, the job cannot be processed if the start time of the next planned setup is violated. Then, this job must wait until the next setup for the family it belongs. In other words, the setup starts cannot be shifted. Hence, planned family domains are not violated.

(ii) Dispatching Policy (2.2): This is a slight modified version of the previous policy where realized setup starts can be shifted when necessary. This is the case when there is a job waiting to be processed from the current family being processed. Through this policy, a job does not wait until the next setup for its family and more jobs will be processed by sharing a common setup. However, job completions from the following families may be further depending on the shift for the setup starts.

In order to distinguish these two versions in simulation, a threshold value is defined that represents total number of jobs waiting to be processed from the next family in turn. Dispatching Policy (2.1) corresponds to a threshold value of (0) whereas for Dispatching Policy (2.2), threshold is set to a large positive number (1000) to reflect the case where next setup start can be shifted as long as there are jobs waiting to be processed from current family.

There are two additional sets to be considered under this policy which are subsets of J_t^{NC}:

- J_t^{CF}: set of jobs waiting to be processed from the most current family being processed before time t
- J_t^{NF}: set of jobs waiting to be processed from next family for which next setup start is planned at time t

Related notation:

- p_j: processing time of job j
- EST: earliest start time
- $|J_t^{CF}|$: the number of jobs waiting to be processed from the most current family being processed before time t
- $|J_t^{NF}|$: the number of jobs waiting to be processed from next family for which next setup start is planned at time t

Under this policy, there are six event types to be considered for discrete-event simulation with the following priority order:

(i) SOH: Start of Horizon
(ii) UCB: Update Current Batch
(iii) JC: Job Completion
(iv) SNS: Start of Next Setup
(v) SC: Setup Completion
(vi) JA: Job Arrival

The pseudocode for Dispatching Policy 2 is given in Figure 7.3.

```
while (No of processed jobs after warm-up ≤ maxJobs) do
    Find the imminent event with minimum occurrence time in FEL, say evim & Remove evim from FEL;
    Set current_time = event time of evim;
    if (Event type of evim = SOH) then
        Solve 1^st phase (batch allocation) with optimization or heuristic & Get planned setup starts;
        For each setup start, create a new SES & UCB event with event times = planned setup starts & Add to FEL;
        Create a new SOH event with event time= current_time + SP and add this event to FEL;
    end if
    if (Event type of evim = UCB) then
        Increase current_batch by 1 in 1^st phase;
    end if
    if (Event type of evim = JC) then
        Let the machine be idle;
        if (|J_t^CF| ≠ 0) then
            Select the job with minimum arrival time from J_t^CF, say jb;
            if (Machine is idle and current_time + p_jb ≤ next setup start) then
                Start processing job jb & Remove jb from J_t^CF;
                Create a new JC event with event time = realized completion time of jb;
            end if
            if (Threshold ≠ 0 and current_time + p_jb > next setup start and |J_t^NF| ≤ Threshold) then
                Shift next setup start to the realized completion time of jb;
                Start processing job jb & Remove jb from J_t^CF;
                Create a new JC event with event time = realized completion time of jb;
            end if
        end if
    end if
    if (Event type of evim = SNS) then
        if ((Machine is idle and |J_t^CF| = 0) or (Machine is idle and |J_t^CF| ≠ 0 and |J_t^NF| ≥ Threshold)) then
            Start a setup operation for the family of the event & Let F(current_batch) = F(evim);
            Create a new SC event with event time = realized setup completion of current_batch;
        else if (Threshold ≠ 0 and machine is busy and |J_t^CF| < Threshold) then
            Shift next setup start to the realized completion time of current_job;
        else if (b = 0 and F(evim) = F(current_batch)) then
            Create a new SC event with event time = EST;
        end if
    end if
    if (Event type of evim = SC) then
        if (|J_t^CF| ≠ 0) then
            Select the job with minimum arrival time from J_t^CF, say jb;
            if (Machine is idle and current_time + p_jb ≤ next setup start) then
                Start processing job jb & Remove jb from J_t^CF;
                Create a new JC event with event time = realized completion time of jb;
            end if
            if (Threshold ≠ 0 and Machine is busy and current_time + p_jb > next setup start and |J_t^NF| ≤ Threshold)
            then
                Shift next setup start to the realized completion time of current_job;
                Start processing job jb & Remove jb from J_t^CF;
                Create a new JC event with event time = realized completion time of jb;
            end if
        end if
    end if
    if (Event type of evim = JA) then
        Assign a due date for i (d_i) with 2^nd phase (due date assignment) optimization model or heuristic;
        if (Machine is idle and (current_time + p_i) ≤ start of next planned setup) then
            Start processing job i & Create a new JC event with event time = realized completion time of i;
        end if
        if (Threshold ≠ 0 and Machine is idle and current_time + p_i > next setup start and |J_t^NF| ≤ Threshold) then
            Shift next setup start to the realized completion time of current_job;
            Start processing job i & Remove i from J_t^CF;
            Create a new JC event with event time = realized completion time of i;
        end if
    end if
end while
```

Figure 7.3. Pseudocode for Dispatching Policy 2.

7.4. Determination of Warm-up Period

System needs to reach steady state before statistics are collected so as to measure performance. In many systems, steady state is not achieved until some time has elapsed after the system is initiated at time zero. A typical remedy for initialization is to consider a start-up or warm-up period. Warm-up period is the time until the behavior of system settles into a regular pattern. The aim of warm-up is to get to steady state. Since we are interested in the long-term behavior of the system, transient state period is ignored and the information gathered during warm-up period is extracted.

During the determination of the length of the warm-up period, the proposed heuristic algorithms in Chapter 6 are used since heuristic algorithms require less computation time compared to optimization models. There are three different cases based on three different dispatching policies for simulation as it can be seen in Table 7.2. Among these three possible cases, FCFS discipline is expected to create the longest queue length since no batching takes place. Hence, as batch allocation and due date asignment problems are solved with the help of the developed heuristic algorithms, jobs are processed according to FCFS policy in simulation for determining the length of warm-up period.

Table 7.2. Cases under Heuristic Algorithms.

1^{st} Phase	2^{nd} Phase	Dispatching Policy
Heuristic	Heuristic	(1) No Batching (FCFS discipline)
Heuristic	Heuristic	(2.1) Batching (Planned setups cannot be shifted ≡ Strict)
Heuristic	Heuristic	(2.2) Batching (Planned setups can be shifted ≡ Flexible)

While deciding on the length of warm-up period, average cumulative queue length at the arrival time of jobs is considered. By experimentation, average cumulative queue length is plotted versus time and the time at which the graph appears to stabilize is inspected, as explained in [99].

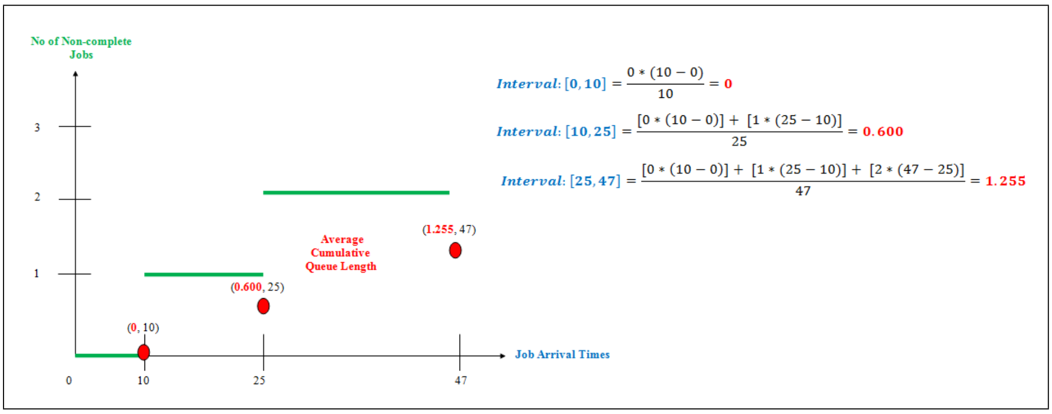


Figure 7.4. Average Cumulative Queue Length.

Consider Figure 7.4. Suppose that three jobs arrive at $10^{th}, 25^{th}$ and 47^{th} time units, respectively. Also, suppose that no jobs are processed until time 47. At each job arrival, the number of non-complete jobs waiting to be processed (including the job itself) is recorded. Since interarrival times between jobs are stochastic, average cumulative queue length is computed based on the number of non-complete jobs and how much time they spend in the queue. The average cumulative queue length at the time of each job arrivals is shown with red dots, together with their computations on the right hand side.

The termination criterion is 3000 job completions in simulation. Simulations are carried out and queue accumulation is observed, starting from effective machine utilization parameter ($\rho_{eff} = 0.8$) to the utilization parameter where the average cumulative queue length becomes out of control and the system becomes oversaturated. For the FCFS dispatching policy, system goes out of state when ($\rho_{eff} = 1.1$). This is an expected outcome in fact, since the number of setups can be as high as the number of jobs if no consecutive jobs belong to the same family.

As it can be observed from some selected examples in Figures 7.5, 7.6 and 7.7 for $F = 2, 3$ and 4, respectively, the system stabilizes around $10,000$ time units for all families. Since mean interarrival time (τ) is fixed to 20, on the average, this time corresponds to the completion time of 500^{th} job. Hence, before starting to collect statistics, in order to reach a steady state, 500 jobs are to be simulated.

No statistics are collected regarding the first 500 jobs processed during the warm-up period. The realized completion time of the 500^{th} job is recorded as the end of warm-up period. Then, as stopping criterion, instead of taking an end time for simulation, simulation is stopped after the completion of a predetermined number of jobs which is taken as 1000 for all simulations, except the jobs processed during warm-up period. In total, $500 + 1000 = 1500$ jobs are processed.

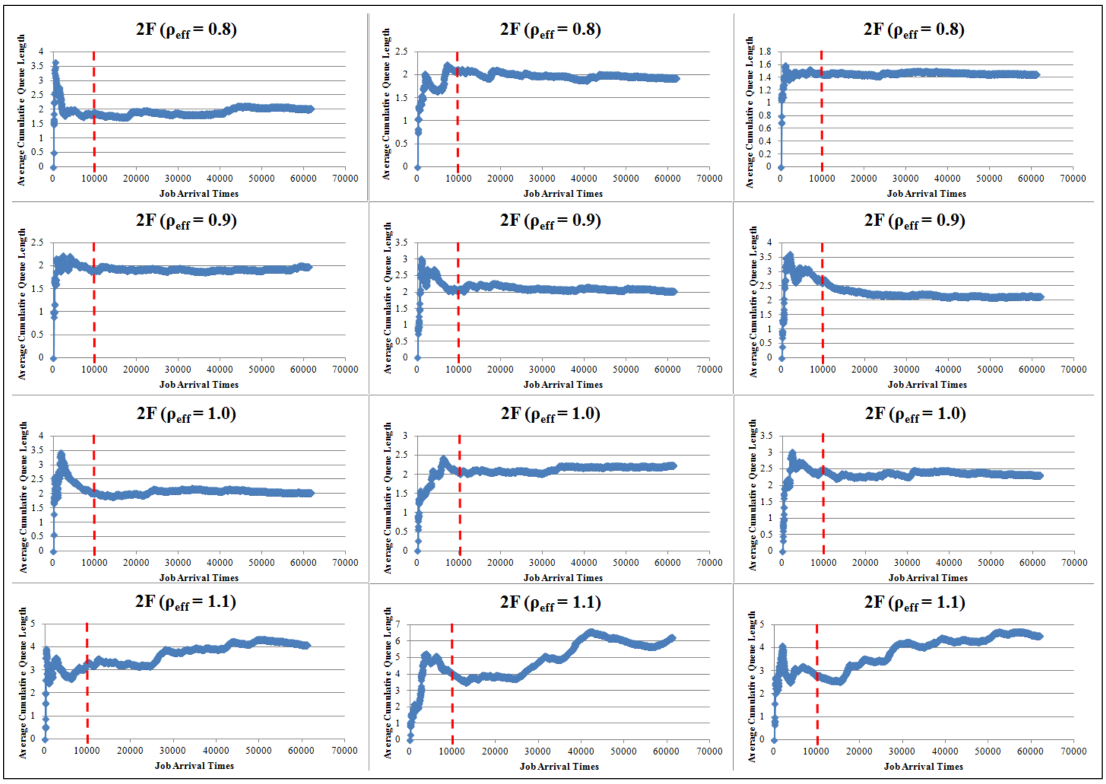


Figure 7.5. Warm-up Period Determination for F=2.

Moreover, since the scheduling periods are taken as $400, 600, 800$ time units (which correspond to $20, 30, 40$ job arrivals per scheduling period on the average, respectively), the number of times the batch allocation problem is solved can be computed as in Table 7.3, which seems quite sufficient.

7.5. Determination of Effective Machine Utilization Parameters

The production system is examined under two utilization levels: high and low. In low utilization cases, 20 per cent less of high utilization parameters are used and rounded to the nearest decimal. The effective utilization parameters for high utilization

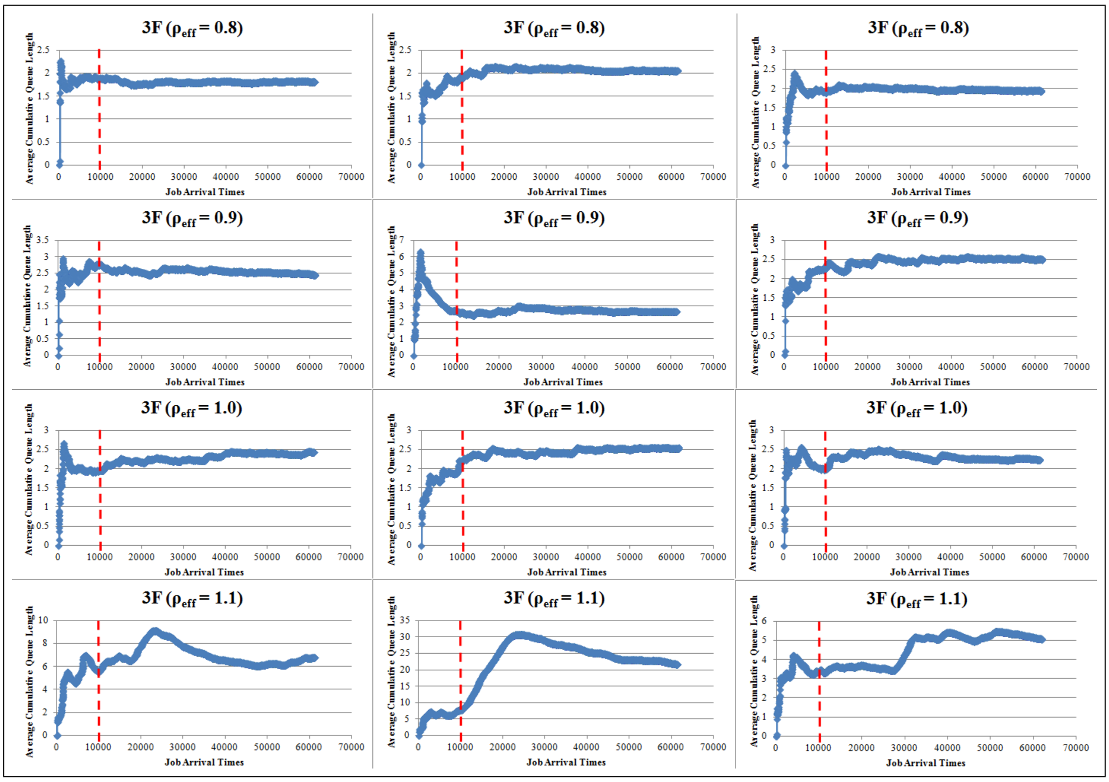


Figure 7.6. Warm-up Period Determination for F=3.

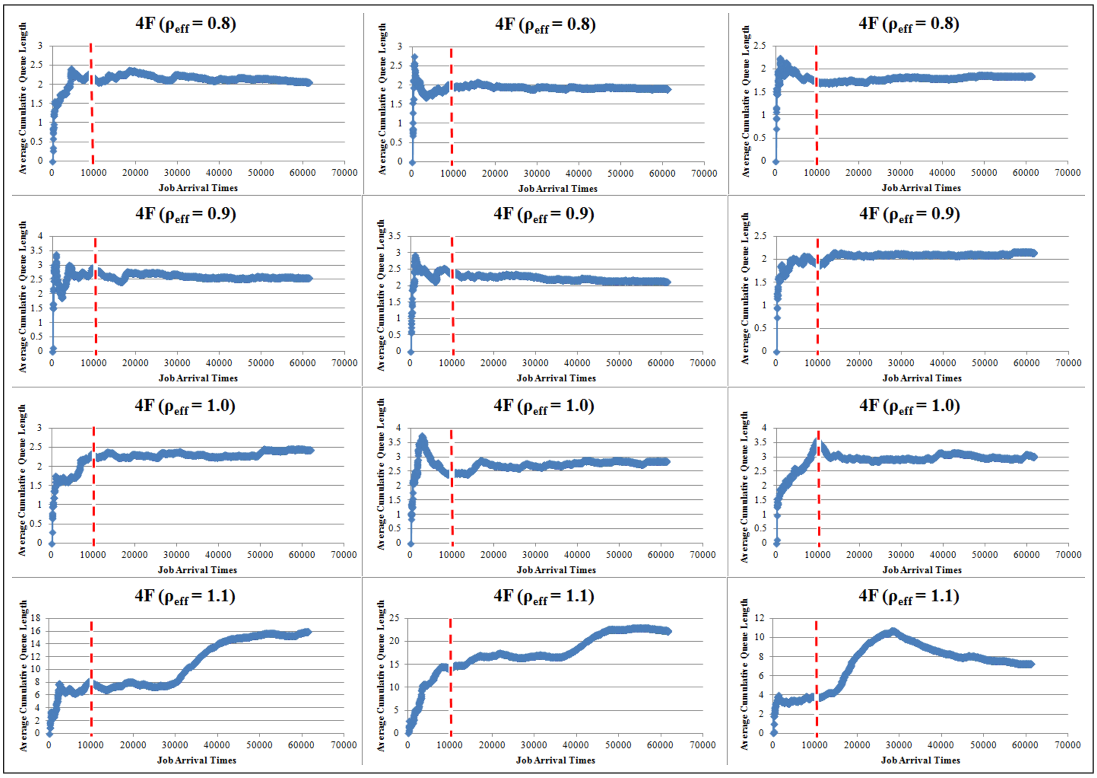


Figure 7.7. Warm-up Period Determination for F=4.

Table 7.3. Number of Times Batch Allocation Problem is Solved.

SP	Expected no of job arrivals per SP	No of times batch allocation problem is solved
400	20	$1500/20 = 75$
600	30	$1500/30 = 50$
800	40	$1500/40 \cong 37$

are determined such that there are non-complete jobs waiting to be processed in the system but the queue size is reasonable and reaches to steady state at a point. Hence, in high utilization cases, there is adequate number of jobs waiting to be processed and the system is not idle. Under low utilization cases, the system is still not idle but the queue length is relatively smaller in size when compared to high utilization cases.

Likewise determination of the length of warm-up period, effective utilization parameters are determined by running the constructed heuristics for both phases for 3000 jobs for three different dispatching policies. Starting from ($\rho_{eff} = 0.8$), average cumulative queue length is drawn as a chart at the time of job arrivals. Every time, (ρ_{eff}) is increased by 0.1 and the analysis is continued until the system starts not to converge to steady state and the queue size gets out of control, by means of average cumulative queue length.

Going through the dispatching policies one by one, the high effective utilization parameter for the Dispatching Policy 1 (FSFC) principle can be taken as $\rho_{eff} = 1.0$ for all families because the system explodes when $\rho_{eff} = 1.1$ for all cases. This can also be viewed in Figures 7.5, 7.6, and 7.7. Then, low utilization parameter can be taken as 0.8, which corresponds to 20 per cent less of 1.0. Under this policy, as it can be seen in Table 7.4, for all cases, the effective utilization parameters do not depend on the duration of setup time (i.e. setup time multiplier π) or the length of scheduling period.

Table 7.4. Effective Utilization Parameters for Dispatching Policy (1).

Family	SP	$\pi = 0.5$		$\pi = 1.5$	
		ρ_{eff}^{low}	ρ_{eff}^{high}	ρ_{eff}^{low}	ρ_{eff}^{high}
F=2	400	0.8	1.0	0.8	1.0
	600	0.8	1.0	0.8	1.0
	800	0.8	1.0	0.8	1.0
F=3	400	0.8	1.0	0.8	1.0
	600	0.8	1.0	0.8	1.0
	800	0.8	1.0	0.8	1.0
F=4	400	0.8	1.0	0.8	1.0
	600	0.8	1.0	0.8	1.0
	800	0.8	1.0	0.8	1.0

However, under Dispatching Policy (2.1), it is observed that utilization parameters differ based on both the setup time multiplier (π) and the length of scheduling period. The effective utilization parameters for Dispatching Policy (2.1) are given in Table 7.5.

Under Dispatching Policy (2.2), it is seen that utilization parameters differ based on only setup time multiplier (π), independent of the length of scheduling period. The effective utilization parameters for Dispatching Policy (2.2) are given in Table 7.6.

Table 7.5. Effective Utilization Parameters for Dispatching Policy (2.1).

Family	SP	$\pi = 0.5$		$\pi = 1.5$	
		ρ_{eff}^{low}	ρ_{eff}^{high}	ρ_{eff}^{low}	ρ_{eff}^{high}
F=2	400	1.0	1.3	1.5	1.9
	600	1.0	1.3	1.5	1.9
	800	1.0	1.3	1.5	1.9
F=3	400	0.9	1.1	1.4	1.7
	600	1.0	1.3	1.5	1.9
	800	1.0	1.3	1.5	1.9
F=4	400	0.8	1.0	1.2	1.5
	600	0.9	1.1	1.4	1.7
	800	0.9	1.1	1.5	1.9

Table 7.6. Effective Utilization Parameters for Dispatching Policy (2.2).

Family	SP	$\pi = 0.5$		$\pi = 1.5$	
		ρ_{eff}^{low}	ρ_{eff}^{high}	ρ_{eff}^{low}	ρ_{eff}^{high}
F=2	400	1.1	1.4	1.5	1.9
	600	1.1	1.4	1.5	1.9
	800	1.1	1.4	1.5	1.9
F=3	400	1.1	1.4	1.5	1.9
	600	1.1	1.4	1.5	1.9
	800	1.1	1.4	1.5	1.9
F=4	400	1.0	1.3	1.5	1.9
	600	1.0	1.3	1.5	1.9
	800	1.0	1.3	1.5	1.9

7.6. Cases

There are eight different cases to be considered in order to measure the performance of the proposed solution methodology. The heuristic algorithm that is constructed for batch allocation is not utilized any more. All cases are listed in Table 7.7. However, due to running time requirement of Case VIII, only 2-family results are reported.

Table 7.7. Design of Solutions.

Case	1^{st} Phase	2^{nd} Phase	Dispatching Policy in Simulation	Shifting allowed?
I	Optimization	Heuristic	No batching (FCFS)	No
II	Optimization	Heuristic	Batching (Strict)	No
III	Optimization	Heuristic	Batching (Flexible)	No
IV	Optimization	Heuristic	No batching (FCFS)	Yes
V	Optimization	Heuristic	Batching (Strict)	Yes
VI	Optimization	Heuristic	Batching (Flexible)	Yes
VII	Optimization	Optimization	No batching (FCFS)	-
VIII	Optimization	Optimization	Batching (Flexible)	-

The parameters and their values in simulation are summarized below:

- F: Number of families = $\{2, 3, 4\}$
- ρ_{eff}: Effective machine utilization parameter = $\left\{\rho_{eff}^{low}, \rho_{eff}^{high}\right\}$
- π: Setup time multiplier = $\{0.5, 1.5\}$
- SP: Length of scheduling period multiplier = $\{400, 600, 800\}$
- TC: Tardiness coefficient = $\{1, 1.5\}$

There are $3(F) \cdot 2(\rho_{eff}) \cdot 2(\pi) \cdot 3(\phi) \cdot 2(TC) = 72$ different scenarios to be tested.

7.7. Performance Measures

Performance measures are as follows:

- $AQLT$: Average quoted lead time
- $ATRDY$: Average tardiness
- AFT: Average flow time
- $BUSY(\%)$: Utilization percentage of the system

Quoted lead time of job j (Q_j) is the difference between quoted due date (d_j) and arrival time (a_j). Total lead time is obtained by adding up quoted lead times for all jobs processed after warm-up period (i.e after 500^{th} job). Then, total quoted lead time is divided by the number of jobs processed after warm-up period (1000 jobs) in order to find the average quoted lead time ($AQLT$):

$$AQLT = \frac{\sum_{j=501}^{j=1500} Q_j}{1000} = \frac{\sum_{j=501}^{j=1500} (d_j - a_j)}{1000} \tag{7.3}$$

Tardiness of job j ($TARD_j$) is the maximum of the difference between actual completion time (c_j) and assigned due date (d_j) or zero. After warm-up period, each job's tardiness is computed and summed to obtain total tardiness which is then divided by the number of jobs processed after warm-up period (1000 jobs) in order to find the average tardiness ($ATRDY$):

$$ATRDY = \frac{\sum_{j=501}^{j=1500} TARD_j}{1000} = \frac{\sum_{j=501}^{j=1500} Max(0,(c_j - d_j))}{1000} \tag{7.4}$$

Flow time of job j (FT_j) is the difference between realized completion time (c_j) and arrival time (a_j). Hence, total flow time is calculated for all jobs processed after warm-up period and divided by the number of jobs processed after warm-up period (1000 jobs) in order to find the average flow time (AFT):

$$AFT = \frac{\sum_{j=501}^{j=1500} FT_j}{1000} = \frac{\sum_{j=501}^{j=1500} (c_j - a_j)}{1000} \tag{7.5}$$

Slack denotes the time in simulation when the machine is idle. Neither a setup is being operated nor a job is being processed at these idle times. Slack time can be formulated as below:

$$SLACK = (ES) - (WP) - (TST) - (TPT) \tag{7.6}$$

where ES is the end of simulation (equivalently, realized completion time of 1500^{th} job), WP is the discarded warm-up period (equivalently, realized completion time of 500^{th} job), TST is the total setup time requirement after warm-up period and TPT is the total processing time requirement of jobs processed after warm-up period.

The utilization of the system, in other words the percentage of time the system is busy, can be computed as below:

$$BUSY(\%) = 100 - \left(\frac{SLACK}{ES}\right) \cdot 100 \tag{7.7}$$

Besides these measures, total number of planned setups in optimization and total number of realized setups in simulation are also recorded.

7.8. Time Limit and Relative MIP Gap Tolerance

The proposed MIP formulations for batch allocation and due date assignment phases are solved using Cplex based on a branch-and-cut algorithm. Based on the pilot runs, although for some instances optimality can be reached within a small amount of running time, sometimes the models may get stuck due to integrality constraints and size of the models. Therefore, the optimization models have to be intervened and terminated.

There are two options for termination. Either a time limit can be specified or an MIP gap tolerance can be defined as stopping criterion. According to [100], there are two kinds of MIP gaps: absolute and relative MIP gap. Absolute optimality tolerance guarantees that a solution lies within a certain absolute range of the optimal solution whereas relative optimality tolerance guarantees that a solution lies within a certain percentage of the optimal solution.

For this study, setting a time limit is preferred to setting a relative MIP gap tolerance. By this way, the optimization models may avoid the chance of missing the optimum or the best possible solution within the given time limit. Specified time limits to terminate solving the optimization models for each phase under each objective are given in Figure 7.8.

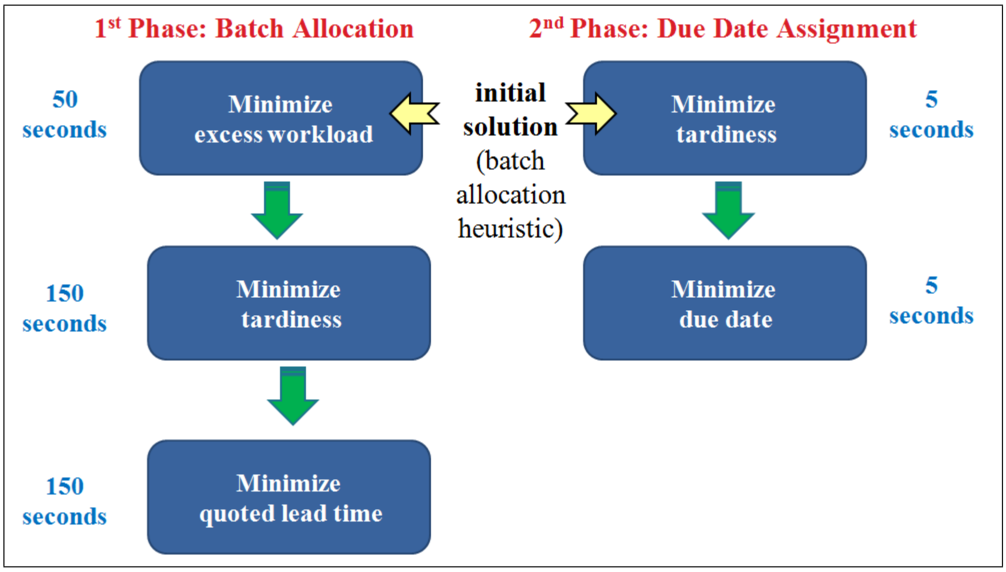


Figure 7.8. Time Limits in Cplex.

8. COMPUTATIONAL TESTS

To demonstrate the performance of the proposed solution methodology, a series of computational experiments are carried out on randomly generated test instances. There are eight proposed solution designs as it can be seen in Table 7.7. All cases are coded in $C\sharp$. The optimization models are embedded into the code using IBM ILOG CPLEX Optimization Studio (v12.6).

The performance of the proposed due date assignment policies with various control parameters are examined under distinct production environments through different values of system parameters. Control parameters are the ones set by the decision maker and as the name implies controllable. On the other hand, systems parameters, that define the production environment, are assumed to be dictated and uncontrollable.

8.1. Effect of Control Parameters

In this study, the length of the scheduling period is one of the controllable parameters. Three different lengths are analyzed as short, medium and long which corresponds to 20, 30 and 40 job arrivals per scheduling period, respectively. Total tardiness allowed for non-complete jobs is managed through a tardiness coefficient which is another controllable parameter. Two different tardiness coefficients are examined that reflect very tight and more flexible upper bounds on total tardiness for non-complete jobs.

The relationship between the control parameters and two key performance indicators (KPI): average quoted lead time (AQLT) and average tardiness (ATRDY), are investigated, in addition to some others. Expectation in terms of KPI is as follows. When average quoted lead time (for new jobs) decreases, average tardiness (for old non-complete jobs) may increase. In order to assign a closer due date for a new arrival, non-complete jobs may be shifted to the right hand side of the schedule and their tardiness may increase due to that shift. On the contrary, in order to minimize the tardiness of non-complete jobs, these jobs may be scheduled at the left hand side of the schedule whereas new arrivals may be assigned to the right hand side of the schedule which is equivalent to assigning further due dates for them.

(i) *The relationship between Scheduling Period (SP) and performance measures*:

(a) *SP and KPI*: SP is dependent on the scheduling period multiplier (ϕ) which represents the expected number of new arrivals per scheduling period. When SP increases, AQLT is expected to increase due to the increase in the number of new arrivals since more number of jobs from the same family can be scheduled consecutively. Then, it may cause larger batch sizes, which in turn, may increase AQLT.

In order to interpret the relationship between two key performance indicators (AQLT and ATRDY) under different SP values, Figure 8.1 can be observed. Figure 8.1 depicts that AQLT increases as SP increases for all families under all dispatching policies, except for Case VII. On the other side, it cannot be concluded that ATRDY reduces against the rise in AQLT when SP grows. ATRDY inclines with the increase in SP for Cases II, V and VIII.

(b) *SP and Number of batches opened in optimization and AQLT*: When SP increases, the number of opened batches in batch allocation optimization model is expected to decrease due to batching. Moreover, AQLT may incline with the decline in the number of batches opened since further due dates may be assigned due to larger batch sizes.

Figure 8.2 highlights that the number of batches opened in optimization model decreases with the increase in SP, which in turn increases AQLT for all families under all dispatching policies, except Case VII. For Case VII, AQLT stays the same, in spite of the decrease in the number of opened batches.

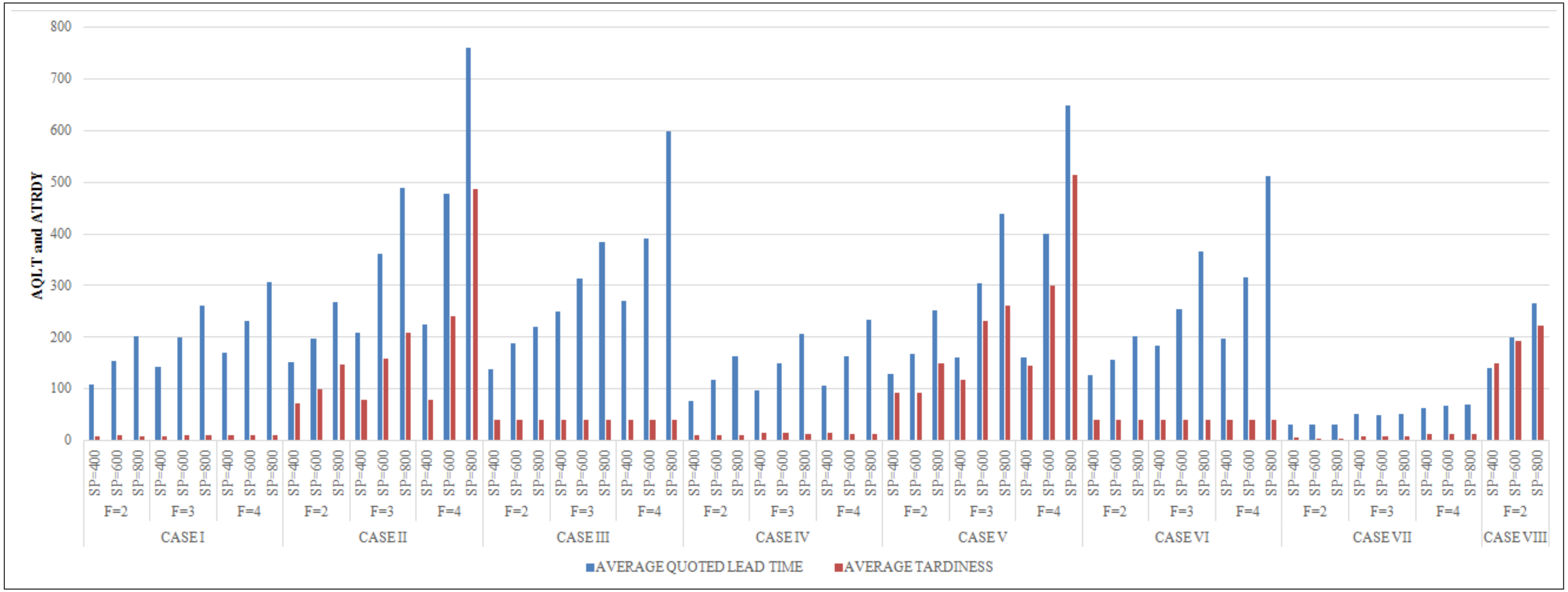


Figure 8.1. Relationship between SP and KPI.

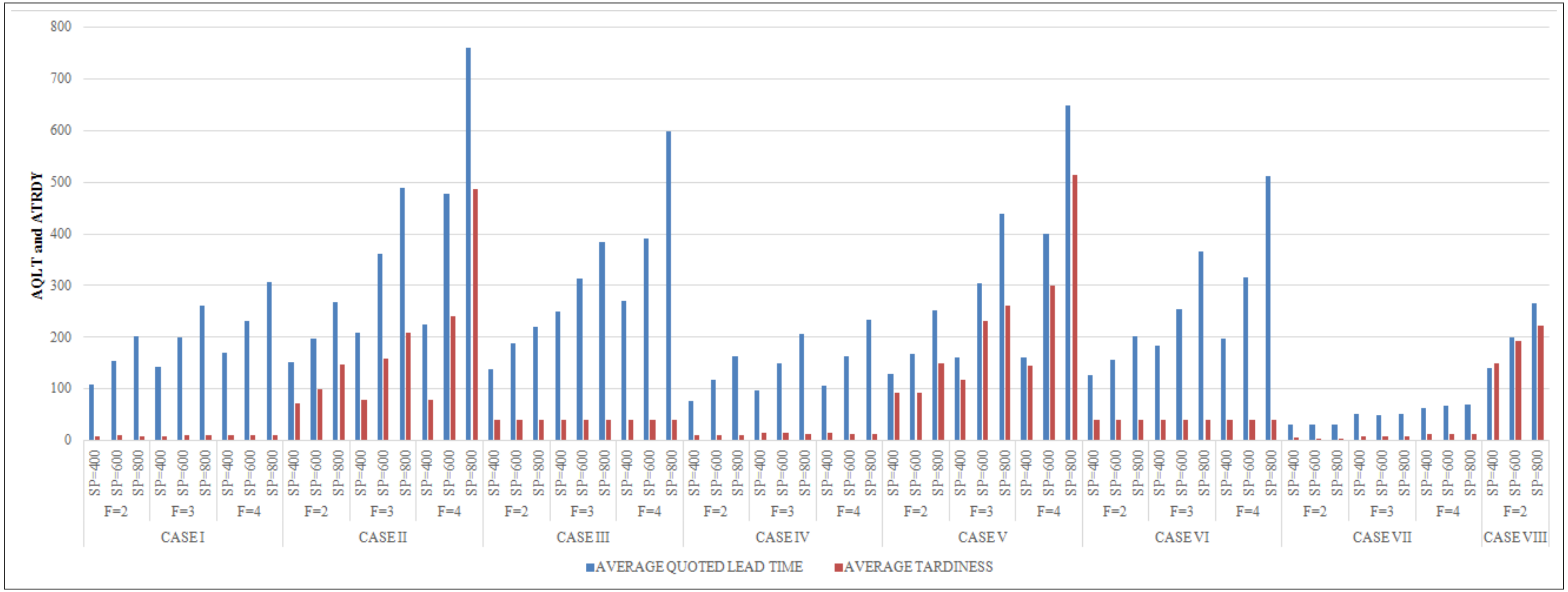


Figure 8.2. Relationship Between SP, Number of Opened Batches in Optimization and AQLT.

(c) *SP and Actual setups incurred in simulation*: When SP increases and planned batching policy is applied, more jobs from the same family can be scheduled successively which may lead to a reduction in the number of setups performed in simulation.

Figure 8.3 indicates that actual setups performed in simulation drop as SP rises for all families. These are the cases which reflect the proposed batching strategy to simulation through dispatching rules. On the other hand, it also reveals that SP has no effect on actual setups performed when jobs are processed according to FCFS principle. This is an expected outcome since the proposed batching policy is not reflected (neither strict nor flexible) and not taken as a guideline during simulation and no batching takes place under FCFS. Under FCFS, the number of setups performed are much higher than the number of setups performed under proposed batching structures.

(ii) *The relationship between TC and key performance measures*: When TC increases, total allowed tardiness for non-complete jobs increases. An increase in TC may give rise to an increase ATRDY. This is because when TC increases, non-complete jobs can be assigned relatively to the right hand side of the schedule, by allowing new arrivals to be assigned to the left hand side of the schedule in order to assign closer due dates.

When TC is strict (TC=1), total tardiness on non-complete jobs is restricted by the minimum amount of tardiness that can be obtained optimally. In this case, new jobs may be scheduled after non-complete jobs in the schedule since due dates already assigned for non-complete jobs cannot be violated not to increase tardiness.

When TC increases (TC=1.5), total tardiness on non-complete jobs becomes less strict. Then, shorter due dates can be quoted for new arrivals by scheduling them earlier and non-complete jobs further in the schedule within the allowed total tardiness. This may bring a decline in AQLT, owing to shorter assigned due dates for new arrivals.

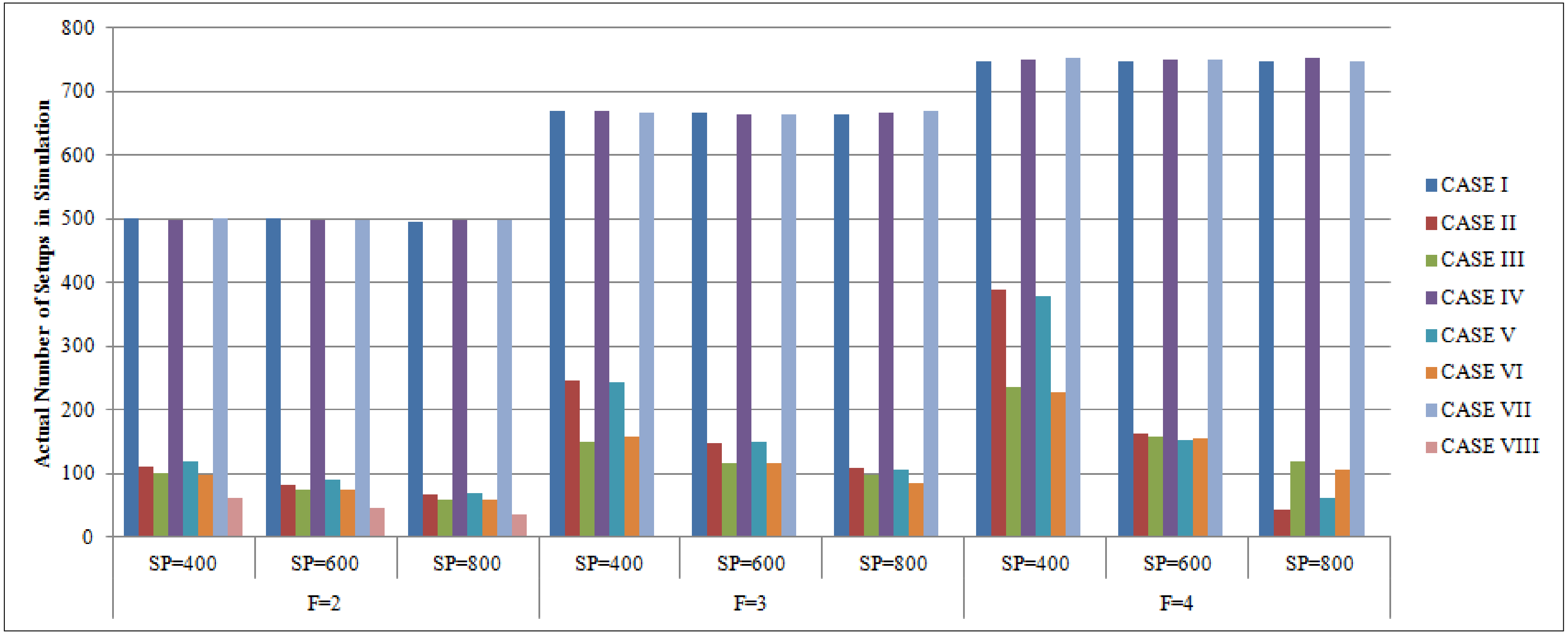


Figure 8.3. Relationship Between SP and Actual Setups in Simulation.

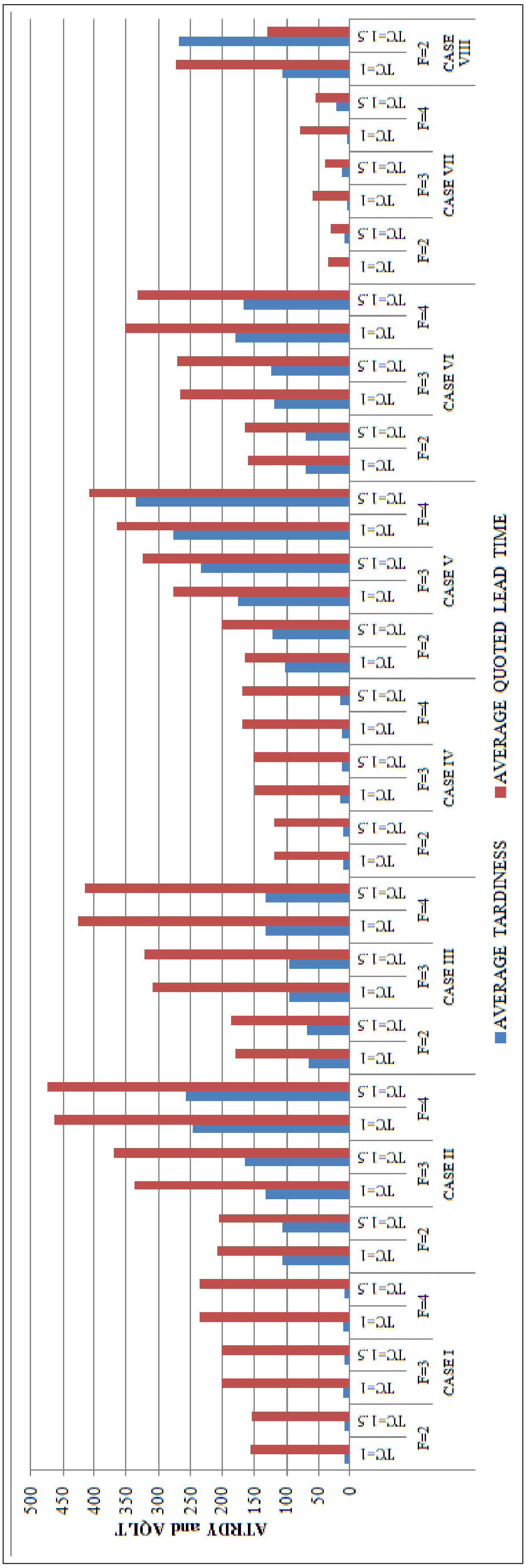


Figure 8.4. Relationship between TC and KPI.

According to Figure 8.4, only Case VII and VIII match with the expectation. This is not surprising because in the first six cases where due dates are assigned through the due date assignment heuristic. Based on the heuristic, non-complete jobs are assigned to the first available batch in terms of their family type. When a new job arrives, the new arriving job is scheduled following non-complete jobs. Hence, for the first six cases, non-complete jobs are not rescheduled in the second phase. Once they are assigned to a position in a batch, no change takes place until the beginning of the next scheduling period.

On the other hand, the results support the power of the proposed optimization models because due dates are assigned via the developed due date assignment optimization model for Case VII and VIII. In both cases, when allowed tardiness on non-complete jobs increases, at the time of a new arrival, these non-complete jobs can be rescheduled further in the schedule which enables assigning a closer due date for the new arrival, at the expense of increasing tardiness of non-complete jobs. This situation exactly reflects the trade-off between the competitive power (which can be increased by assigning a closer due date for a new arriving job) and delivery reliability (which can be minimized by compliance to the assigned due dates of non-complete jobs).

The findings generally overlap with the expectations which support the verification of the constructed solutions.

8.2. KPI Comparison between Cases within each Dispatching Policy

In terms of dispatching policies for simulation: Case I, IV and VII implement FCFS rule, Case II and V apply proposed strict batching whereas Case III, VI and VIII practice proposed flexible batching policy, as tabulated in Table 7.7. Based on different effective utilization parameters of each dispatching policy, cases under the same dispatching policy are comparable with each other. Two KPI's (AQLT and ATRDY) are examined for each case under each dispatching policy.

According to Figure 8.5, lowest AQLT values are obtained by Case VII where due dates are assigned through the due date assignment optimization model. Additionally, lower AQLT values with Case IV compared to Case I.

According to Figure 8.6, lower AQLT values are obtained by Case V where due dates are assigned through the due date assignment heuristic and unlimited shifting is allowed within the heuristic. On the contrary, lower ATRDY values are obtained by Case II where due dates are assigned through the due date assignment heuristic and shifting is not allowed within the heuristic. This result apparently shows that the trade-off between AQLT and ATRDY. Lower ATRDY values are obtained by not shifting the completion times of non-complete jobs, at the expense of assigning further due dates for new arriving jobs. Yet, closer due dates can be quoted for new arrivals by shifting their completion times.

According to Figure 8.7, as opposed to expectation, lowest AQLT values are not obtained by Case VIII where due dates are assigned through the due date assignment optimization model. Then, a further investigation is conducted by analyzing the outputs based on tardiness coefficients. As displayed in Figure 8.8, nearly the same AQLT and ATRDY values are obtained by Case III and VI when TC=1 and TC=1.5. This is because once non-complete jobs are assigned to a position by the due date assignment heuristic, rescheduling of them is not possible. But, lower AQLT values are observed when shifting is allowed (as in Case VI) and lower ATRDY values are observed when shifting is not allowed (as in Case III).

In terms of Case VIII, for F=2 and TC=1.5, lower AQLT and higher ATRDY values are obtained, compared to Case III and VI, as demonstrated in Figure 8.8. This can be explained as follows. When a strict upper bound is applied for total tardiness of non-complete jobs (when TC=1), new arrivals may be scheduled after non-complete jobs the model. However, when the upper bound on total tardiness of non-complete jobs becomes more flexible (when TC=1.5), closer due dates may be assigned for new arrivals by assigning them before non-complete jobs.

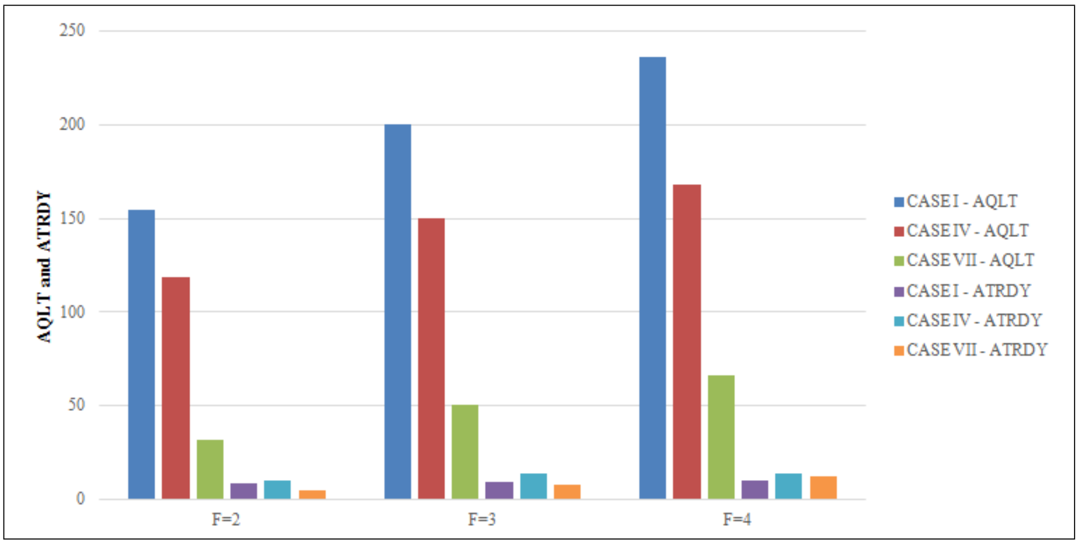


Figure 8.5. Comparison of KPI for Case I, IV and VII under FCFS Policy.

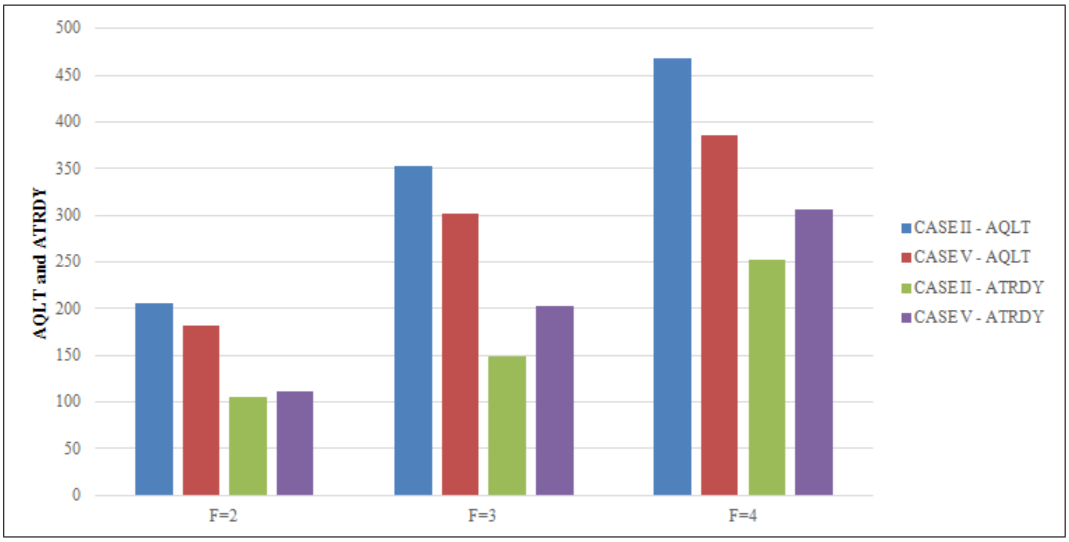


Figure 8.6. Comparison of KPI for Case II and V under Strict Batching Policy.

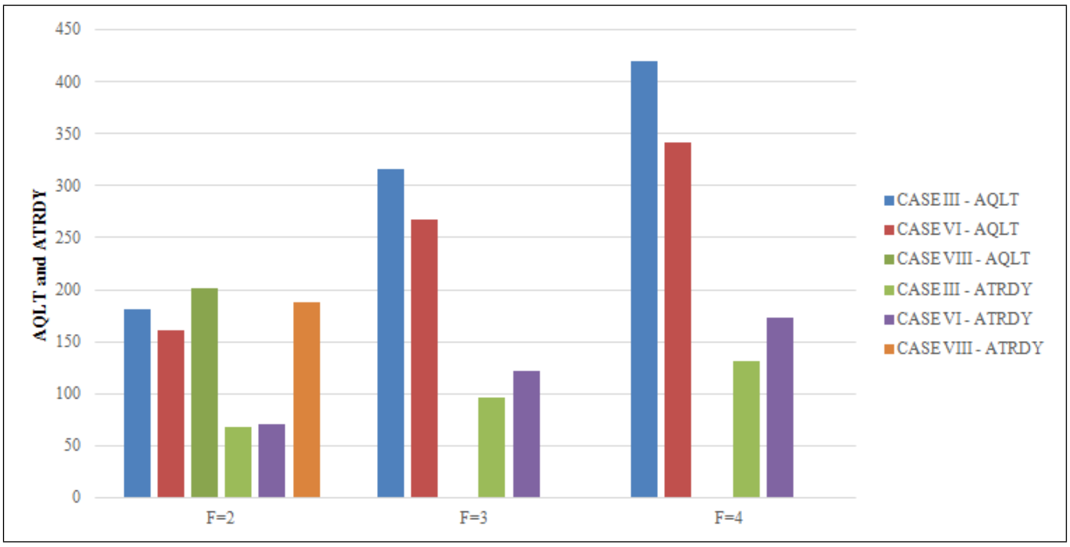


Figure 8.7. Comparison of KPI for Case III, VI and VIII under Flexible Batching Policy.

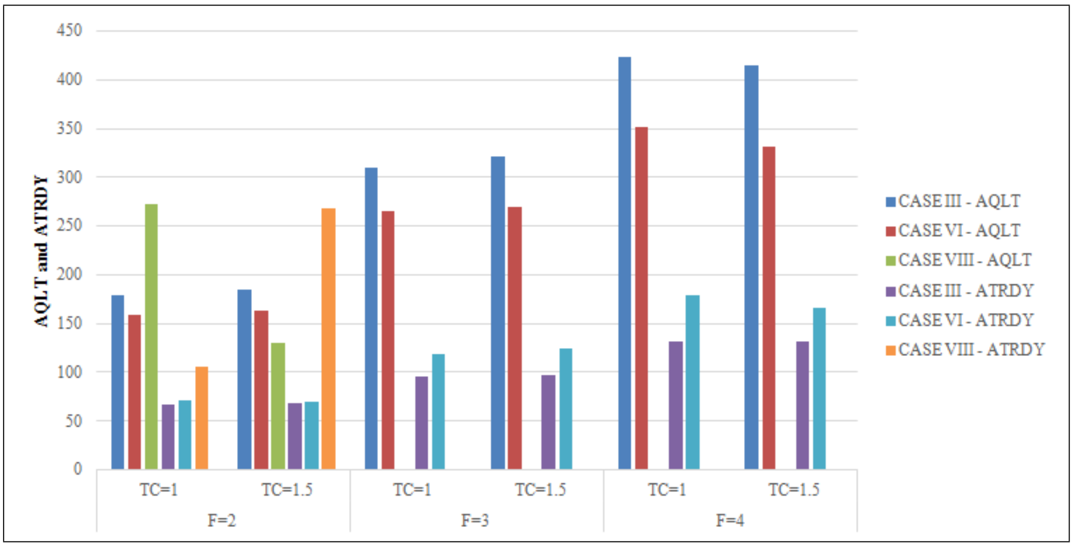


Figure 8.8. Comparison of KPI for Case III, VI and VIII under Flexible Batching Policy through TC.

8.3. Actual System Utilization Rates

Two different utilization levels are considered, as low and high. High utilization refers to the case where there is sufficiently enough number of jobs in the queue, waiting to be processed, whereas there is relatively less number of jobs in the queue in low utilization case. On the average, the difference between high and low utilization is expected to be around 20 per cent, based on the effective utilization parameters. Figure 8.9 demonstrates actual busy percentages (in other words, utilization rates) of the machine for all cases.

According to Figure 8.9, actual utilization rate does not drop below 70 per cent for any case. Higher (lower) utilization levels are obtained based on high (low) effective utilization parameters.

8.4. Relative MIP Gap

For this study, relative MIP gaps are recorded for all cases and for all objective function levels. Based on Figure 8.10, PHASE I/OBJ 0, PHASE I/OBJ 1 and PHASE I/OBJ 2 refers to Z_0, Z_1 and Z_2 in Section 5.1. Also, PHASE II/OBJ 1 and PHASE II/OBJ 2 refers to Z_1 and Z_2 in Section 5.2.

According to Figure 8.10, if relative MIP gap is not zero for a 2-family case, relative MIP gap increases as F grows. Lower MIP gaps are observed for Case I, IV and VII, compared to other cases. The highest relative MIP gap is observed for 4-family cases while minimizing PHASE I/OBJ 2. Lower relative MIP gaps are observed in the second phase, compared to the first phase which may be associated with the amount of binary restrictions and size of the optimization models.

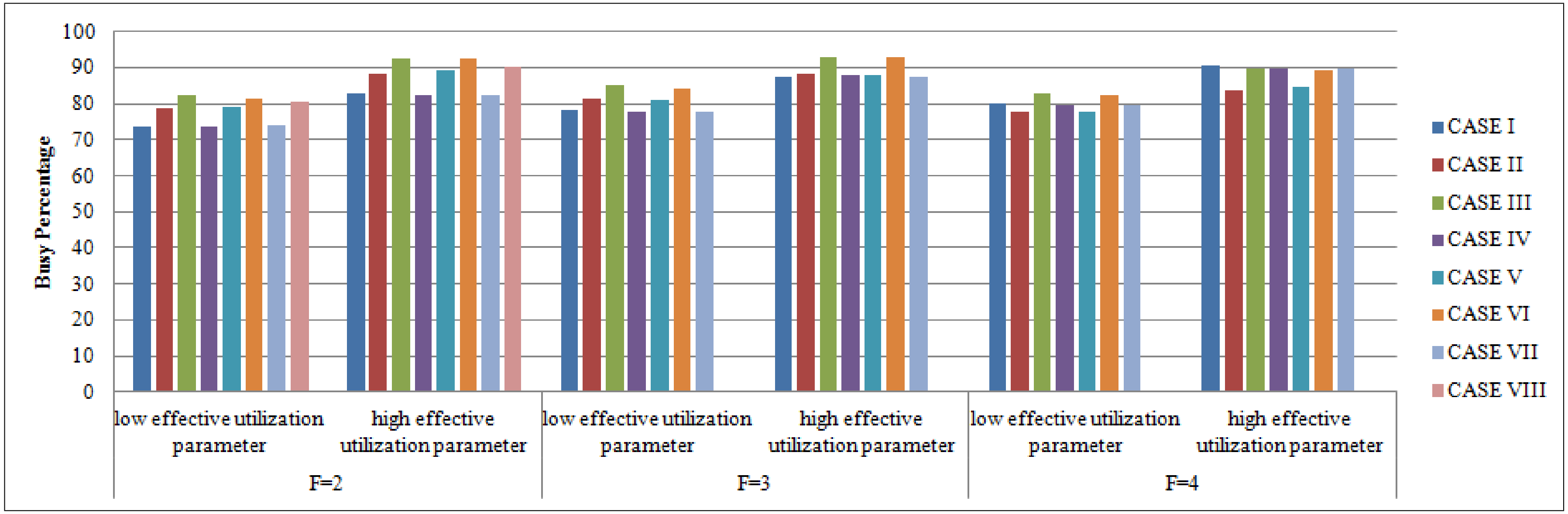


Figure 8.9. Actual Busy Percentages.

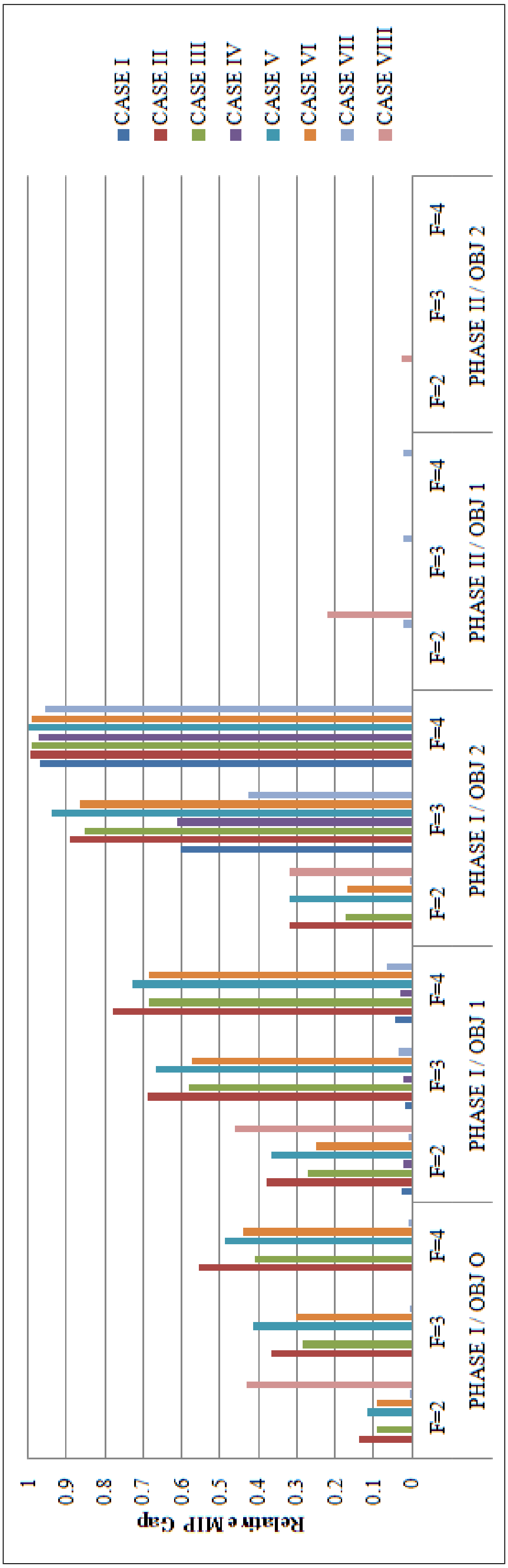


Figure 8.10. Relative MIP Gaps.

9. PERFORMANCE COMPARISON

Unfortunately, there is no study that can be used to make a direct performance comparison that deals with the same problem. Therefore, the idea is to compare the performance of the proposed solution with FCFS dispatching rule by combining it with the frequently used due date setting rules.

As mentioned in the literature review, in most of the studies in the area of due date assignment, a flow allowance (δ_j) is added to arrival time (a_j) and due date of job j can be represented as $d_j = a_j + \delta_j$.

There are two selected papers ([48] and [80]) that consider CON, SLK and TWK due date assignment methods in dynamic and stochastic single machine environments. As mentioned in Chapter 2, these are parametric due date assignment methods and due date tightness depends on these parameters such that:

- CON ($d_j = a_j + \gamma$): all jobs are given constant flow allowances
- SLK ($d_j = a_j + p_j + \beta$): all jobs are given equal flow allowances that reflect equal slack (waiting time)
- TWK ($d_j = a_j + \alpha \cdot p_j$): jobs are given flow allowances proportional to processing time requirements

However in both studies, family setups are not taken into consideration. But to be able to make a performance comparison, a comparison procedure is constructed by applying some modifications, based on the point of view of these papers.

In both papers, an allowance factor is defined to normalize the due date assignment policy and represent the long-run aggregate tightness of due dates such that:

$$Allowance\ \ factor = a = \frac{\bar{f}}{\bar{p}}$$

where $\bar{f}$ is the mean flow allowance and $\bar{p}$ is the mean processing time. By increasing the allowance factor (a) gradually, performance is measured in terms of the average tardiness.

For each of the considered due date setting rules in both papers, the relationship between tightness parameters and the allowance factor is expressed as follows:

- CON: $\bar{f} = \gamma \rightarrow \gamma = a \cdot \bar{p} \quad (\Rightarrow a = \frac{\bar{f}}{\bar{p}} = \frac{\gamma}{\bar{p}} = \frac{a \cdot \bar{p}}{\bar{p}} = a)$
- SLK: $\bar{f} = \bar{p} + \beta \rightarrow \beta = (a-1) \cdot \bar{p} \quad (\Rightarrow a = \frac{\bar{f}}{\bar{p}} = \frac{\bar{p}+\beta}{\bar{p}} = \frac{\bar{p}+(a-1)\cdot\bar{p}}{\bar{p}} = a)$
- TWK: $\bar{f} = \alpha \cdot \bar{p} \rightarrow \alpha = a \quad (\Rightarrow a = \frac{\bar{f}}{\bar{p}} = \frac{\alpha \cdot \bar{p}}{\bar{p}} = a)$

On the other hand, the problem that we consider consists of family setups and an objective so as to minimize quoted lead time of new arriving jobs while minimizing tardiness of old non-complete job. For all cases, average quoted lead time (AQLT), average tardiness (ATRDY) and average flow time (AFT) information are already known based on simulation runs. And in fact, the flow allowance in the due date setting rules above is equivalent to quoted lead time in our study. Because, by definition both refer to the difference between the assigned due date of a job and its arrival time. Moreover, mean processing time $\bar{p}$ is denoted by μ in our study (which depends on ρ). Two different ways are proposed for comparison:

(i) Let $\bar{f} = AQLT$ (average quoted lead time, obtained through simulations) and observe average tardiness (ATRDY). Then, due date of job j can be assigned as follows:
  - CON: $\gamma = AQLT \rightarrow d_j = a_j + \gamma = a_j + AQLT$
  - SLK: $\beta = AQLT - \mu \rightarrow d_j = a_j + p_j + \beta = a_j + p_j + (AQLT - \mu)$
  - TWK: $\alpha = \frac{AQLT}{\mu} \rightarrow d_j = a_j + \alpha \cdot p_j = a_j + \frac{AQLT}{\mu} \cdot p_j$

(ii) Let $\bar{f} = AFT$ (average flow time, obtained through simulations) and observe average tardiness (ATRDY). Then, due date of job j can be assigned as follows:
  - CON: $\gamma = AFT \rightarrow d_j = a_j + \gamma = a_j + AFT$
  - SLK: $\beta = AFT - \mu \rightarrow d_j = a_j + p_j + \beta = a_j + p_j + (AFT - \mu)$
  - TWK: $\alpha = \frac{AFT}{\mu} \rightarrow d_j = a_j + \alpha \cdot p_j = a_j + \frac{AFT}{\mu} \cdot p_j$

In the first case, by assigning the same quoted lead time and in the second case, by assigning the same actual flow allowance; if any of the three due date assignment rules can obtain a lower average tardiness value, then they are said to outperform the proposed solution. Otherwise, the proposed solution is said to be superior to the commonly used due date assignment rules, in terms of performance.

The average of outputs are rounded to the nearest integers. Bar plots are used to make a comparison over the mean values. Additionally, boxplots are drawn to be able to observe variability of the outputs.

There are seven seven due date assignment methods under consideration:

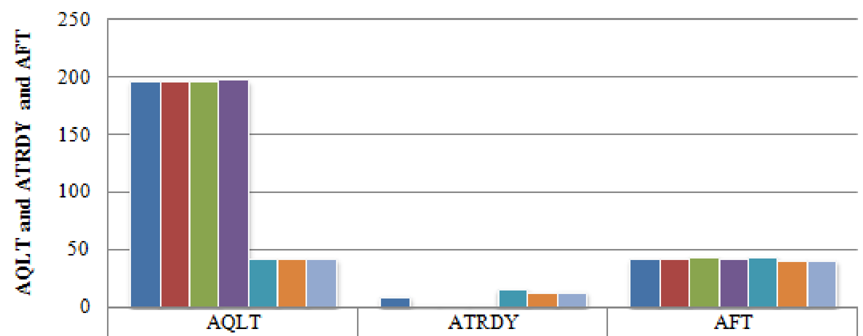
(i) Proposed policy (FCFS for Case I, IV, VII; Batching (Strict) for Case II, V; Batching (Flexible) for Case III, VI,VIII)
(ii) CON (AQLT based)
(iii) SLK (AQLT based)
(iv) TWK (AQLT based)
(v) CON (AFT based)
(vi) SLK (AFT based)
(vii) TWK (AFT based)

Figure 9.1 exhibits the behavior of Case I under seven due date assignment methods. According to Figure 9.1, lower ATRDY values can be obtained with AQLT based due date setting rules, under the same AQLT of the proposed policy. However, based on AFT based due date setting rules, ATRDY increases with the decrease in AQLT and ATRDY values are found to be higher than ATRDY of the proposed policy. Based on Figure 9.4, the same consequence can be drawn for Case IV (as Case I) which also follows FCFS rule in simulation.

Case VII also implements FCFS dispatching policy in simulation, however higher ATRDY values are obtained with AQLT based due date setting rules, as it is displayed in Figure 9.7. Moreover, based on AFT based due date setting rules, higher ATRDY values are observed than ATRDY values of AQLT based due date setting rules. The difference of Case VII from Case I and IV is that, due date assignment takes place based on due date assignment optimization model and non-complete jobs can be rescheduled. On the other hand, due dates are assigned via due date assignment heuristic in Case I and IV where once non-complete jobs are assigned to a position in a batch, rescheduling is not possible until solving the next batch allocation problem.

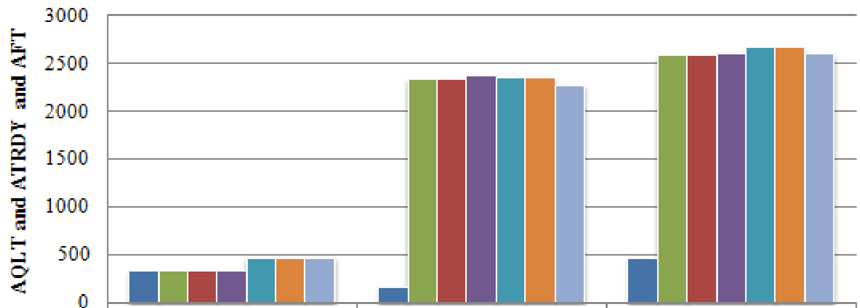
According to Figure 9.2 and Figure 9.5 where jobs are processed according to strict batching policy, very high ATRDY values are obtained with both AQLT based and AFT based due date setting rules. ATRDY values are about 14 times and 11 times greater than the ATRDY value of the proposed policy, for Case II and V, respectively.

According to Figure 9.3, Figure 9.6 and Figure 9.8 where jobs are processed according to flexible batching policy,very high ATRDY values are obtained with both AQLT based and AFT based due date setting rules. ATRDY values are about 35 times, 30 times and 11 times greater than the ATRDY value of the proposed policy, for Case III, VI and VIII, respectively.



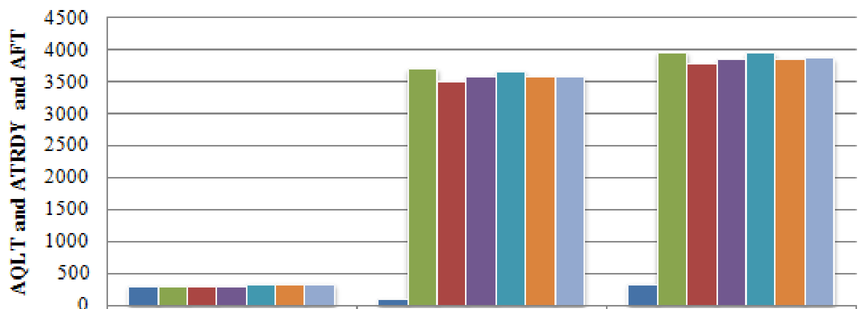
	AQLT	ATRDY	AFT
PROPOSED POLICY	197	9	42
CON (AQLT based)	197	1	42
SLK (AQLT based)	197	1	44
TWK (AQLT based)	197	2	42
CON (AFT based)	42	16	43
SLK (AFT based)	42	13	41
TWK (AFT based)	42	13	40

Figure 9.1. Case I - Comparison of Mean Values under CON, SLK, TWK.



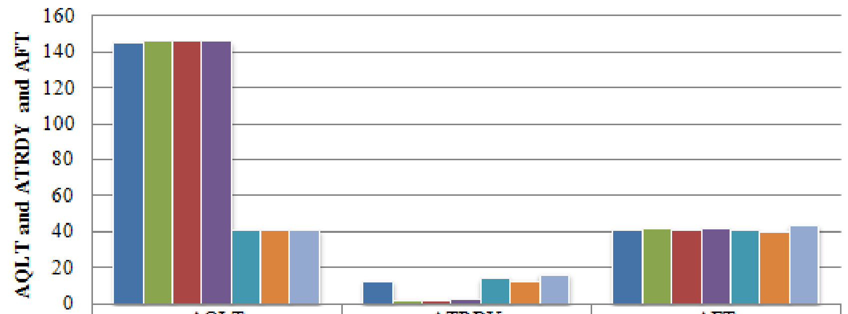
	AQLT	ATRDY	AFT
PROPOSED POLICY	339	167	472
CON (AQLT based)	339	2344	2586
SLK (AQLT based)	339	2344	2586
TWK (AQLT based)	339	2378	2614
CON (AFT based)	472	2354	2671
SLK (AFT based)	472	2351	2674
TWK (AFT based)	472	2279	2605

Figure 9.2. Case II - Comparison of Mean Values under CON, SLK, TWK.



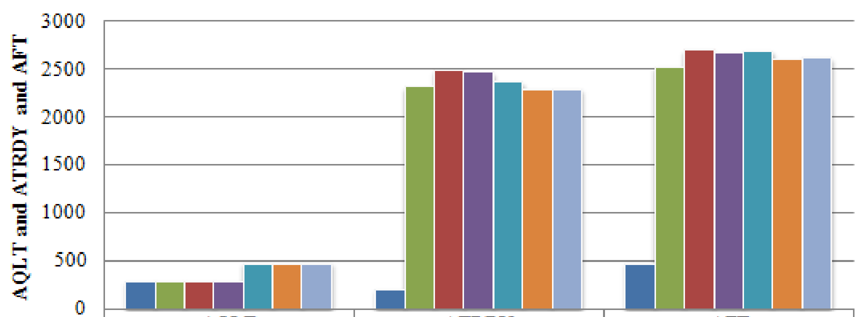
	AQLT	ATRDY	AFT
PROPOSED POLICY	306	98	340
CON (AQLT based)	306	3716	3973
SLK (AQLT based)	306	3514	3779
TWK (AQLT based)	306	3594	3850
CON (AFT based)	340	3668	3954
SLK (AFT based)	340	3587	3865
TWK (AFT based)	340	3589	3875

Figure 9.3. Case III - Comparison of Mean Values under CON, SLK, TWK.



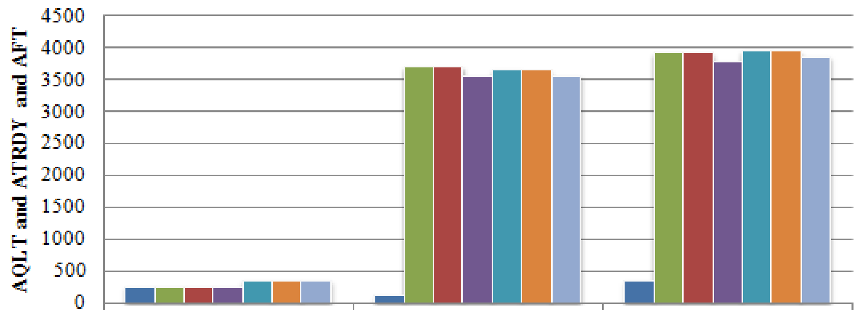
	AQLT	ATRDY	AFT
PROPOSED POLICY	146	13	41
CON (AQLT based)	146	2	42
SLK (AQLT based)	146	2	41
TWK (AQLT based)	146	3	42
CON (AFT based)	41	14	41
SLK (AFT based)	41	13	40
TWK (AFT based)	41	17	43

Figure 9.4. Case IV - Comparison of Mean Values under CON, SLK, TWK.



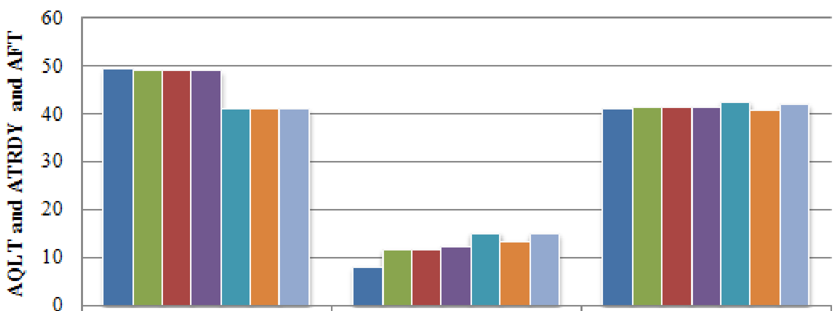
	AQLT	ATRDY	AFT
PROPOSED POLICY	288	205	477
CON (AQLT based)	288	2317	2524
SLK (AQLT based)	288	2496	2704
TWK (AQLT based)	288	2468	2677
CON (AFT based)	477	2368	2695
SLK (AFT based)	477	2290	2605
TWK (AFT based)	477	2296	2622

Figure 9.5. Case V - Comparison of Mean Values under CON, SLK, TWK.



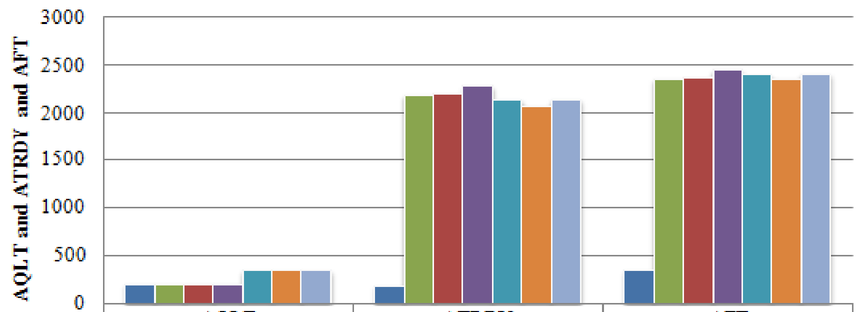
	AQLT	ATRDY	AFT
PROPOSED POLICY	257	121	354
CON (AQLT based)	257	3712	3934
SLK (AQLT based)	257	3709	3929
TWK (AQLT based)	257	3569	3791
CON (AFT based)	354	3666	3964
SLK (AFT based)	354	3650	3950
TWK (AFT based)	354	3572	3868

Figure 9.6. Case VI - Comparison of Mean Values under CON, SLK, TWK.



	AQLT	ATRDY	AFT
PROPOSED POLICY	49	8	41
CON (AQLT based)	49	12	41
SLK (AQLT based)	49	12	41
TWK (AQLT based)	49	12	41
CON (AFT based)	41	15	43
SLK (AFT based)	41	13	41
TWK (AFT based)	41	15	42

Figure 9.7. Case VII - Comparison of Mean Values under CON, SLK, TWK.



	AQLT	ATRDY	AFT
PROPOSED POLICY	201	188	345
CON (AQLT based)	201	2194	2364
SLK (AQLT based)	201	2201	2371
TWK (AQLT based)	201	2286	2457
CON (AFT based)	345	2142	2399
SLK (AFT based)	345	2081	2362
TWK (AFT based)	345	2147	2413

Figure 9.8. Case VIII - Comparison of Mean Values under CON, SLK, TWK.

In addition to comparisons through mean values, box plots are drawn to analyze the outputs deeper, using R as data analysis software. A sample box plot is given in Figure 9.9. The minimum and maximum observations, first and third quartiles and median are exhibited in box plots. 25 per cent of the observations lie below the first quartile whereas 25 per cent lie above the third quartile. The difference between the third and first quartiles, which is called the interquartile range, covers the central 50 per cent of the data. Median is the number separating the higher half from the lower half, when data is ranked.

The position of median shows whether the distribution is symmetric or skewed, either to the right or left. If the median is closer to the third quartile, then the distribution is skewed to the left. If the median is closer to the first quartile, then the distribution is skewed to the right. Also, if the median is fairly close to the mean, the distribution is symmetric. For right-skewed distributions, mean is greater than median (i.e. mean is on the right of median). For left-skewed distributions, mean is less than median (i.e. mean is on the left of median).

The proposed policies are compared with six different due date assignment methods, in terms of average quoted lead time, average tardiness and average flow time. Although we have statistical outputs regarding these performance measures through simulations, the underlying distributions are not known. Hence, non-parametric box plots are preferred to parametric confidence intervals since they require no assumptions about the distribution of these performance measures. Moreover, a box plot is an indicator of central tendency, spread and symmetry. In that sense, a box plot provides insights about important characteristics of a distribution. Box plots allow to display how the data is distributed without making any assumptions about underlying distribution but most parametric methods generally assume a common parametric distribution, such as normal distribution.

When box plots are lined up side by side, they can be compared at a glance through parallel box plots. Parallel box plots are useful for visually comparing more than one data set. Moreover, instead of comparing only the proposed policy with each traditional due date assignment one by one for significance, they enable to compare all seven methods within each other visually and simultaneously.

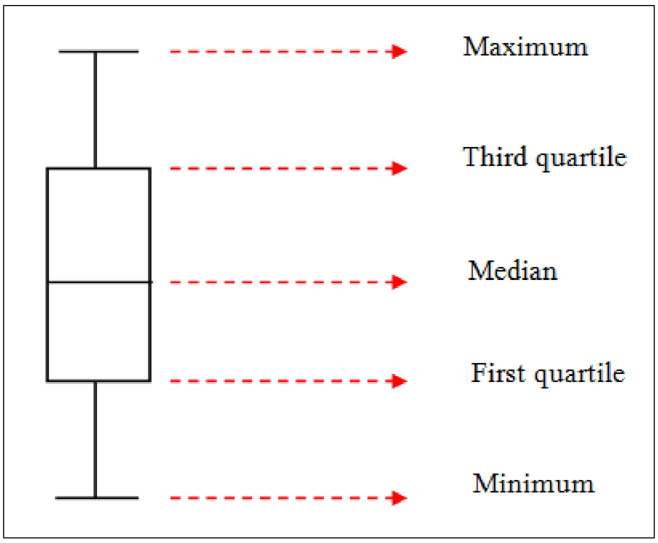


Figure 9.9. Five-Number Summary of a Box Plot.

In three parallel boxplots, AQLT, ATRDY and AFT take place on the vertical axis, respectively. Due date assignment methods take place on the horizontal axis and these are represented with numbers as follows:

(i) Proposed policy
(ii) CON (AQLT based)
(iii) SLK (AQLT based)
(iv) TWK (AQLT based)
(v) CON (AFT based)
(vi) SLK (AFT based)
(vii) TWK (AFT based)

Consider Case I in Figure 9.10. Median of AQLT is fairly close to mean AQLT, representing a symmetric distribution for AQLT under the proposed policy. As expected, median of AQLT is approximately equal to the mean AQLT for the first three and mean AFT for the last three due date assignment methods. Besides, the variability of AQLT of the proposed policy is higher than the other methods whereas the outputs are clustered around the median for the others. This is very normal since AQLT values for them are set, based on given mean AQLT for the first three of the due date assignment methods and given mean AFT for the last three methods. Yet, all box plots regarding AQLT can be used to examine the distribution of AQLT for the proposed policy under all cases. Variability of ATRDY for the proposed policy is more than AQLT based due date setting rules whereas it is less than AFT based due date setting rules. The proposed policy is superior to AFT based due date setting rules, in terms of ATRDY. However, AQLT based due date setting rules outperform the proposed policy, in terms of ATRDY. In terms of AFT, the medians are very close to each oher for each method and variability are nearly the same. This is due to FCFS dispatching policy applied by all seven methods under Case I. Consider Figure 9.13 and 9.16. As it is displayed in these graphs, similar consequences can be drawn for Case IV and VII which also implement FCFS policy in simulation, such as Case I.

Consider Case II in Figure 9.11. Median of AQLT is less than mean AQLT and median is also closer to the first quartile, representing a right-skewed distribution for AQLT under the proposed policy. The distributions of ATRDY and AFT are skewed to the right for all due date assignment methods. Variability of ATRDY is very less than both AQLT based and AFT based due date setting rules. Under the proposed policy, very low ATRDY values are obtained which cluster around the median. The proposed policy is significantly superior to other six due date assignment methods, in terms of ATRDY. For all policies except the proposed policy, the median and variability are nearly the same with each other but worse than the proposed due date assignment policy, in terms of AFT. When the proposed policy is applied in simulation, significantly lower AFT values are obtained. Consider Figure 9.14. As it is displayed in this graph, similar consequences can be drawn for Case V which applies the same policy with Case II. Consequently, the proposed strict batching policy dramatically outperforms the other six methods, in terms of ATRDY and AFT.

Consider Case III in Figure 9.12. Median of AQLT is less than mean AQLT and median is also closer to the first quartile, representing a right-skewed distribution for AQLT under the proposed policy. The distributions of ATRDY and AFT can be said to be slightly skewed to the left, except for the proposed policy. Variability of ATRDY is extremely less than both AQLT based and AFT based due date setting rules. Under the proposed policy, very low ATRDY values are obtained which gather around the median. The proposed policy is considerably better than other six due date assignment methods, in terms of ATRDY. For all policies except the proposed policy, the median and variability are nearly the same with each other but worse than the proposed due date assignment policy, in terms of AFT. When the proposed policy is reflected to simulation, significantly lower AFT values are obtained. Consider Figure 9.15 and 9.17. As it is displayed in these graphs, similar consequences can be drawn for Case VI and VIII, which also implement the same policy with Case III. Consequently, the proposed flexible batching policy considerably outperforms the other six methods, in terms of ATRDY and AFT.

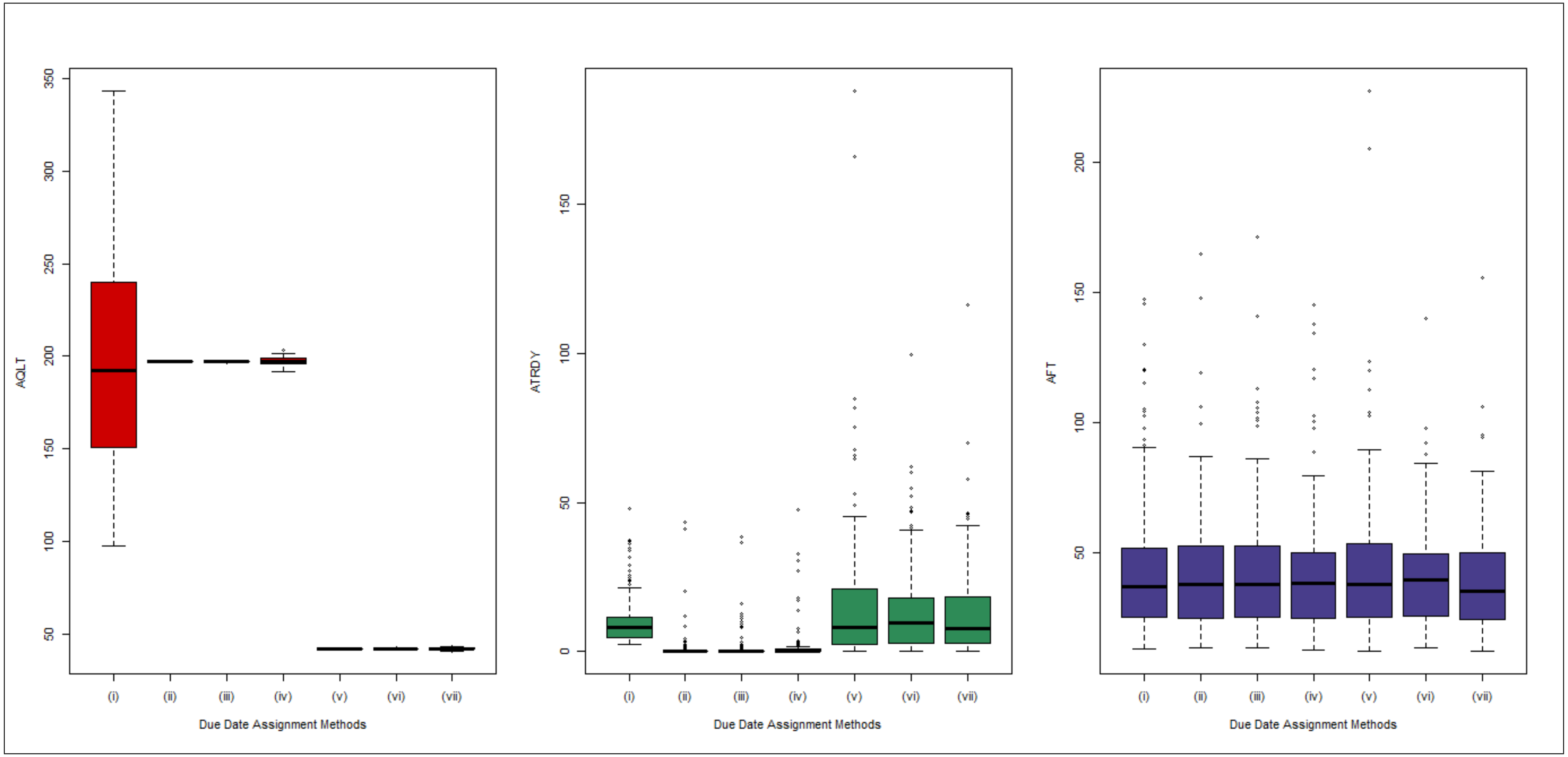


Figure 9.10. Case I - Box Plot.

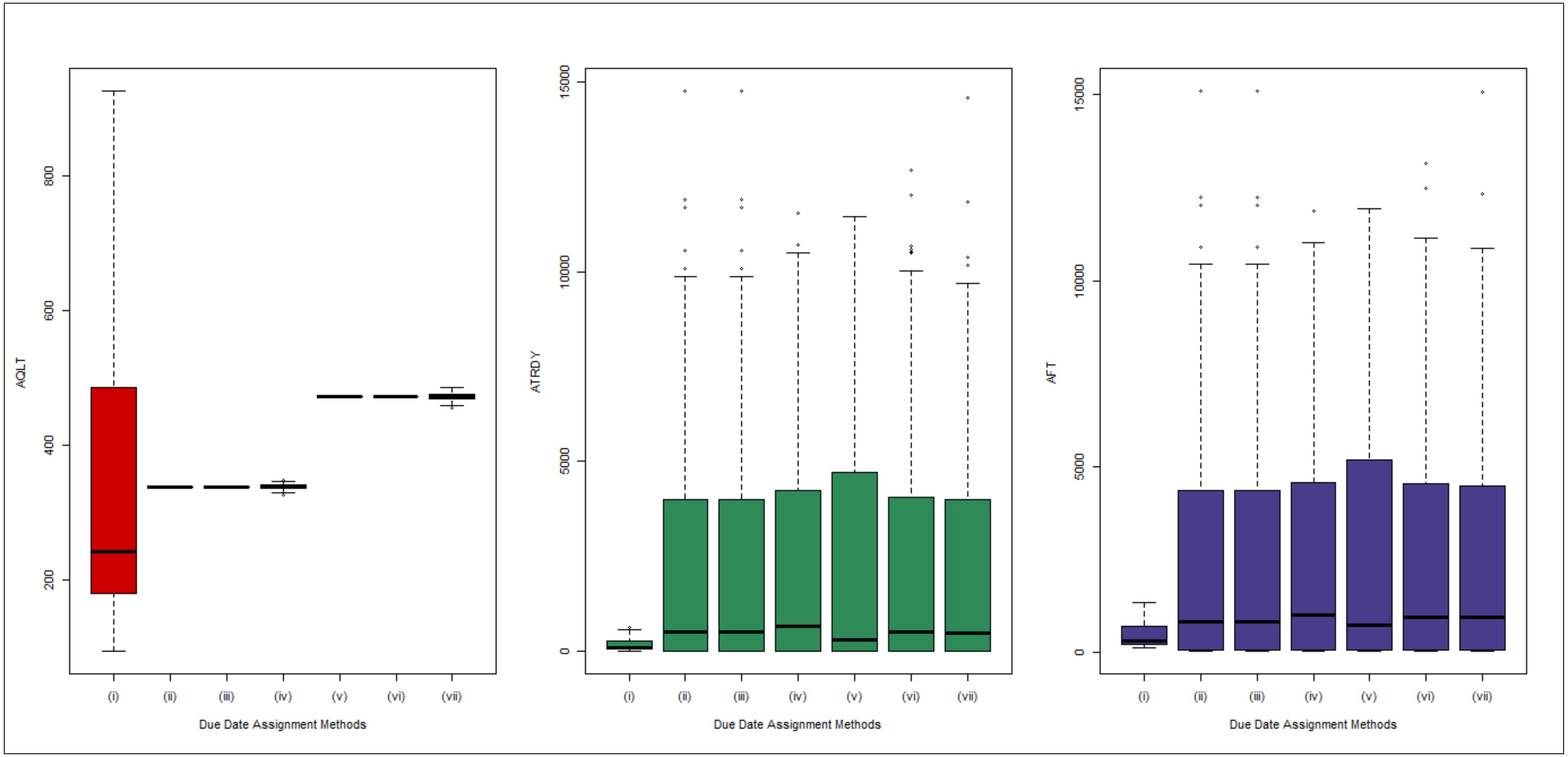


Figure 9.11. Case II - Box Plot.

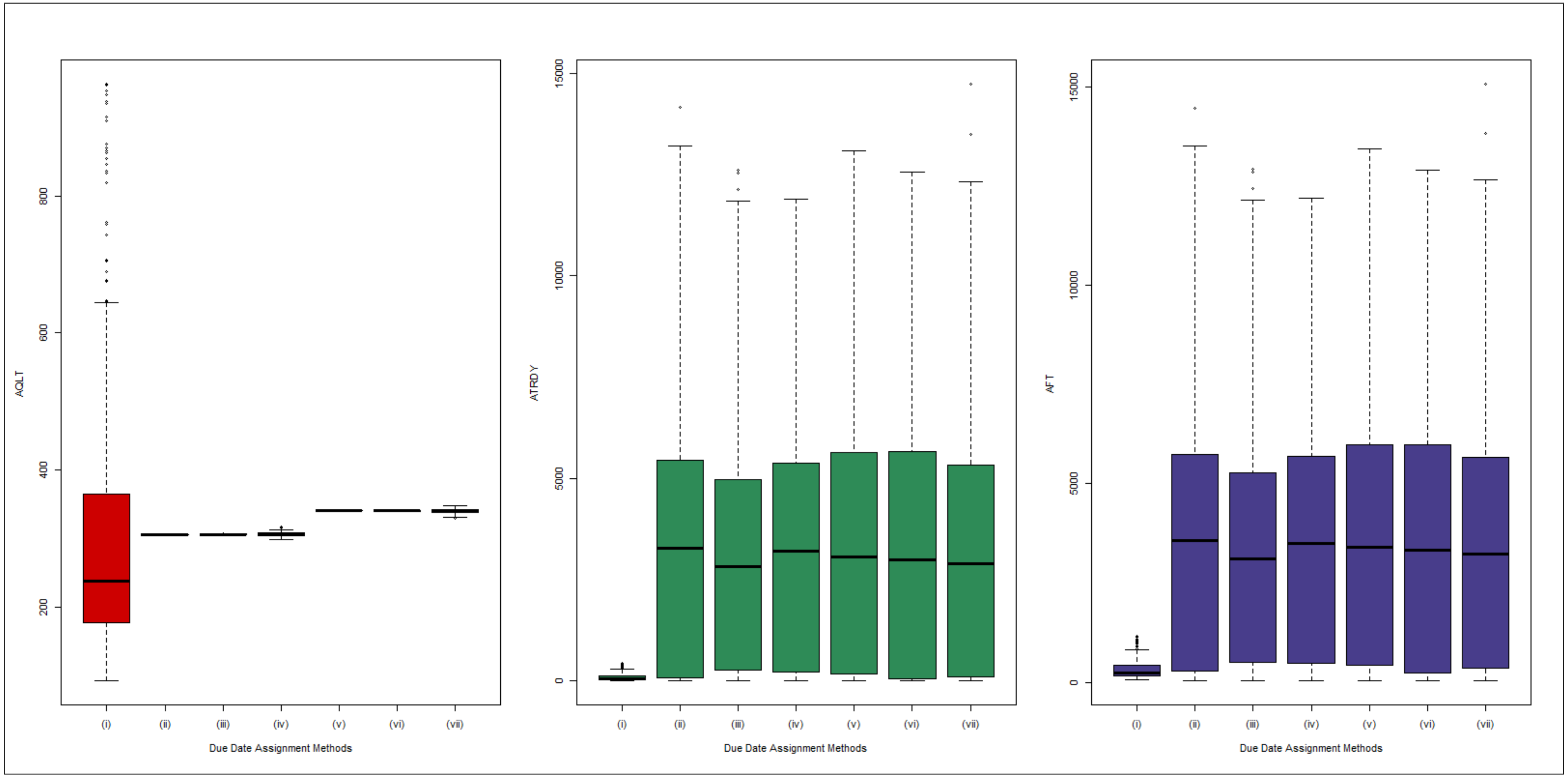


Figure 9.12. Case III - Box Plot.

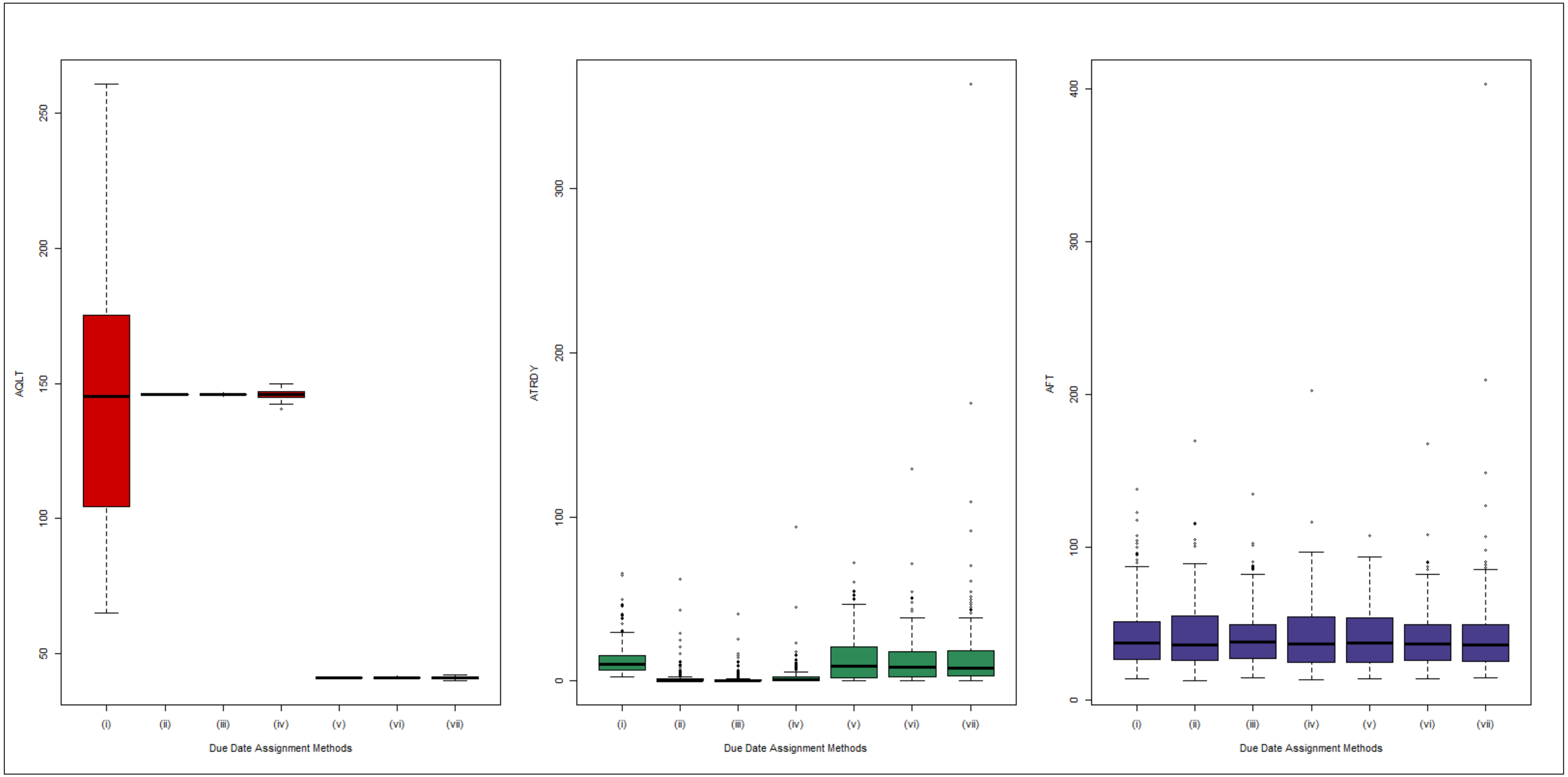


Figure 9.13. Case IV - Box Plot.

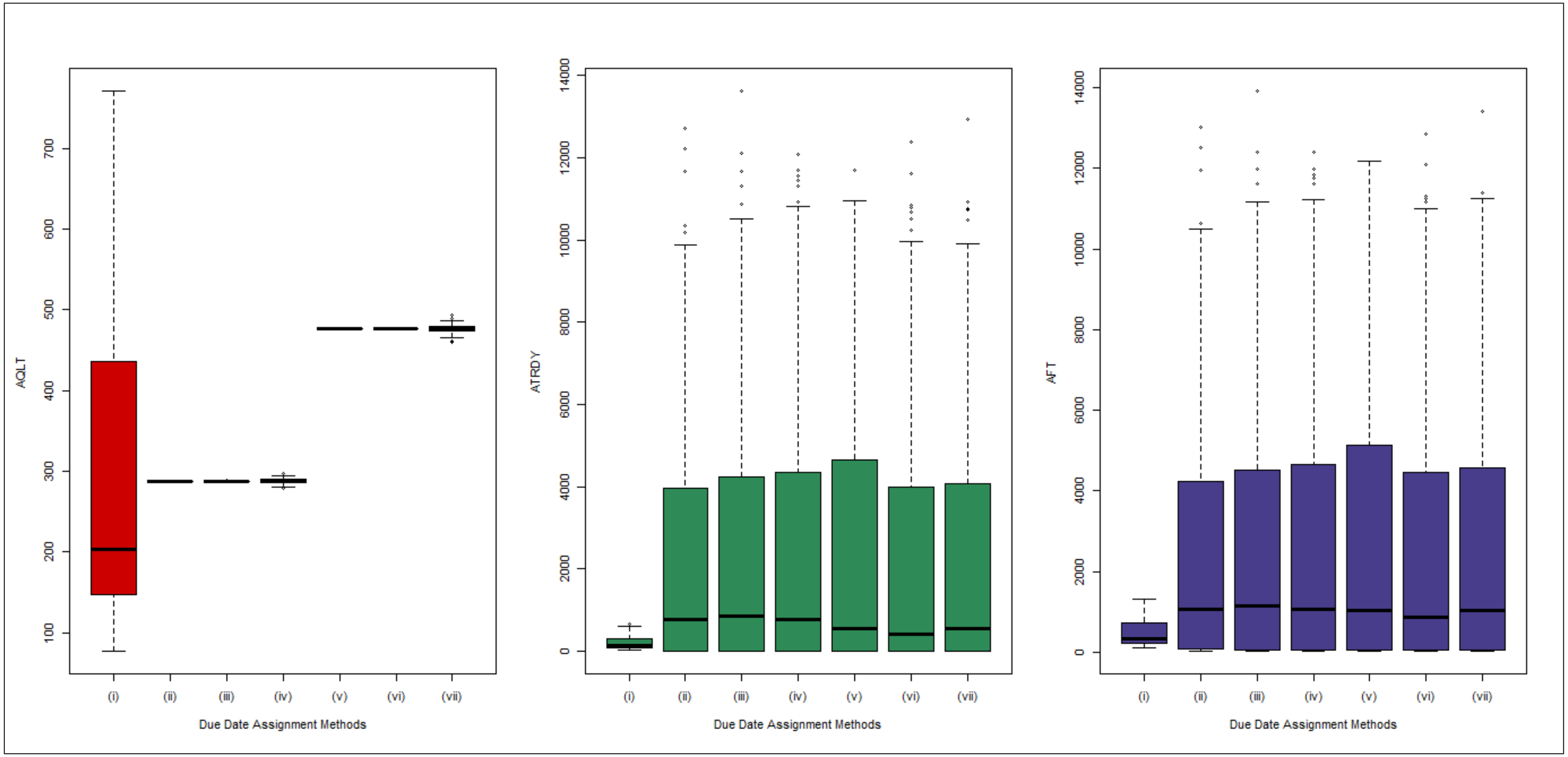


Figure 9.14. Case V - Box Plot.

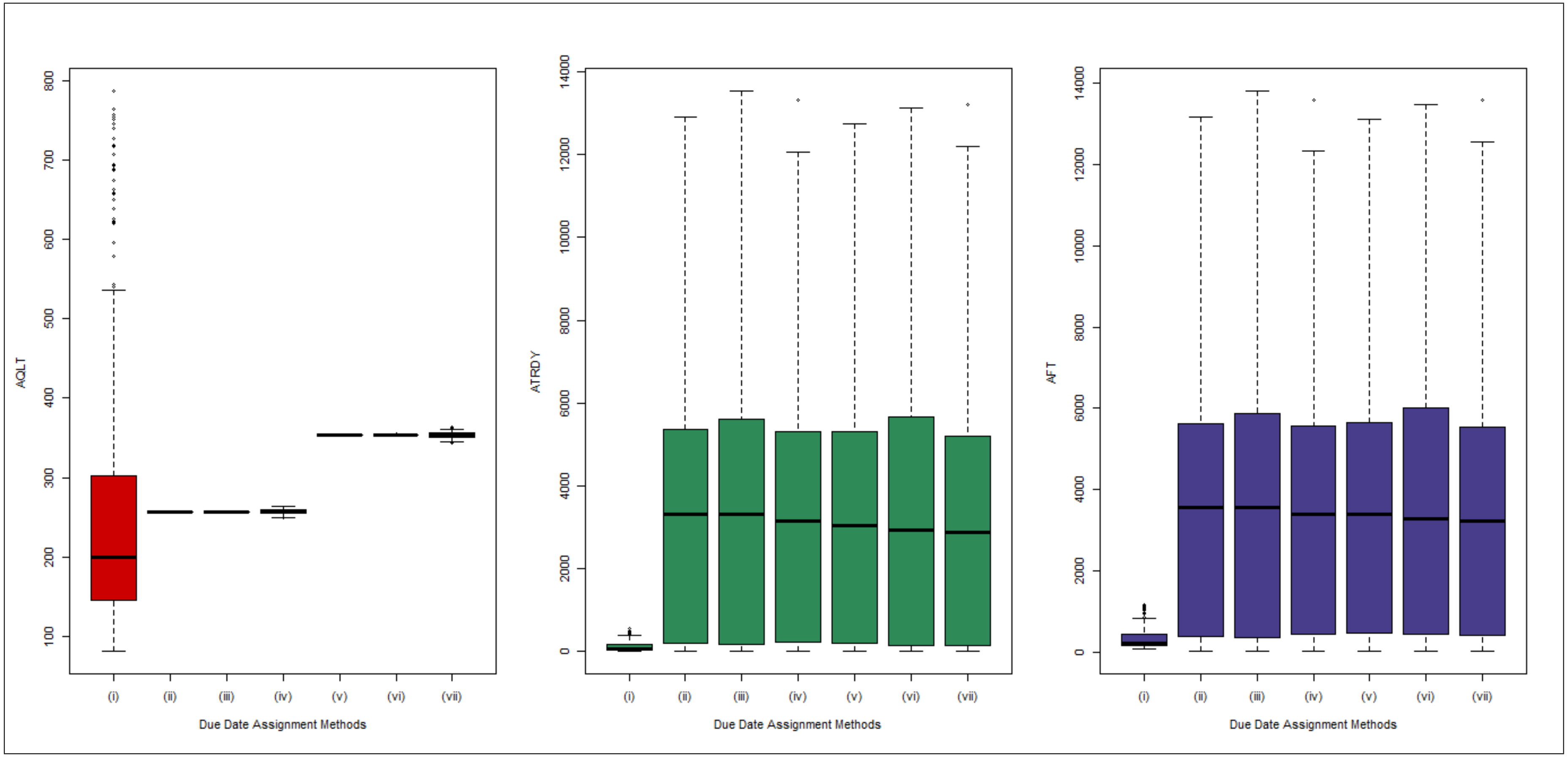


Figure 9.15. Case VI - Box Plot.

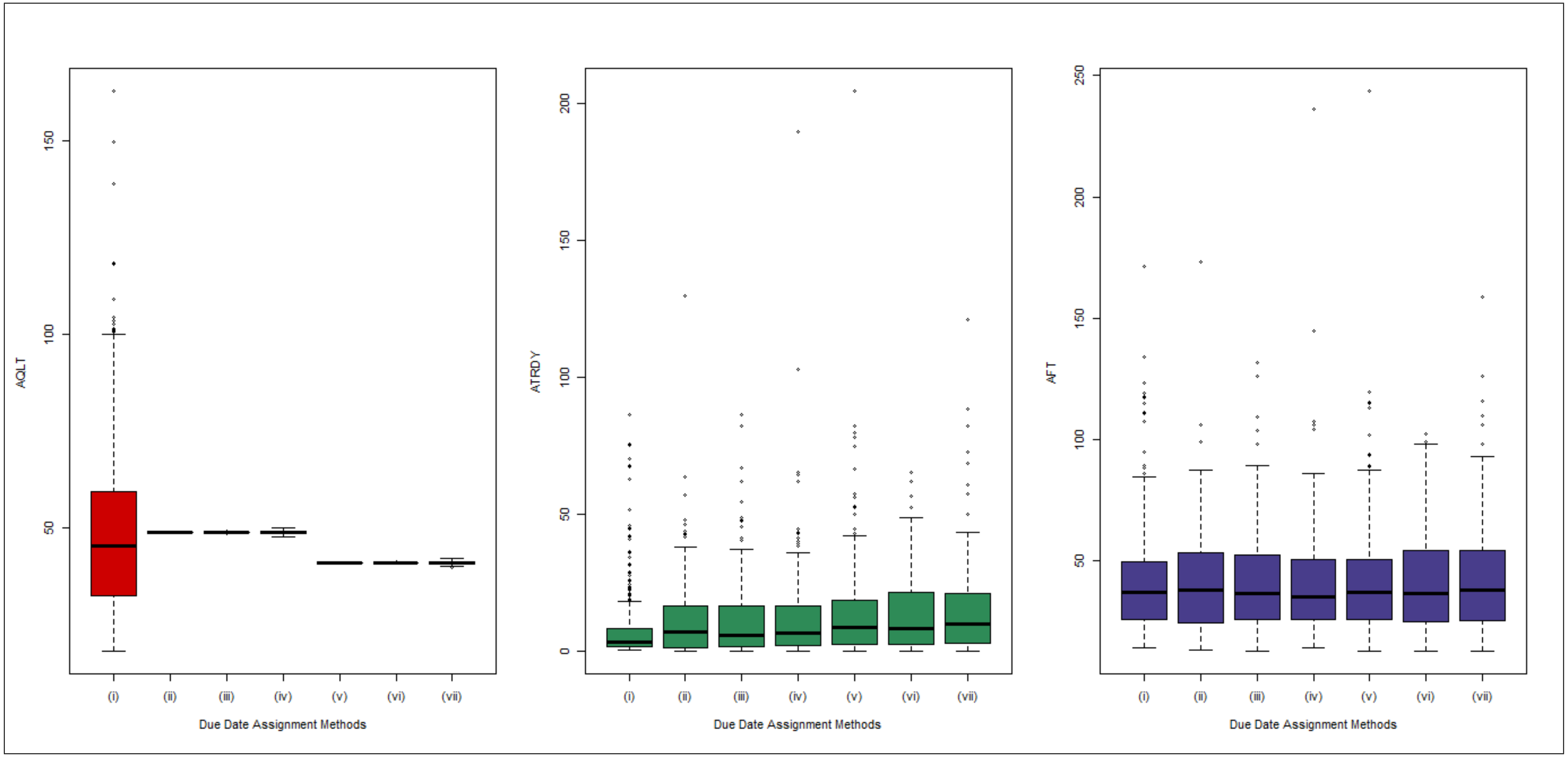


Figure 9.16. Case VII - Box Plot.

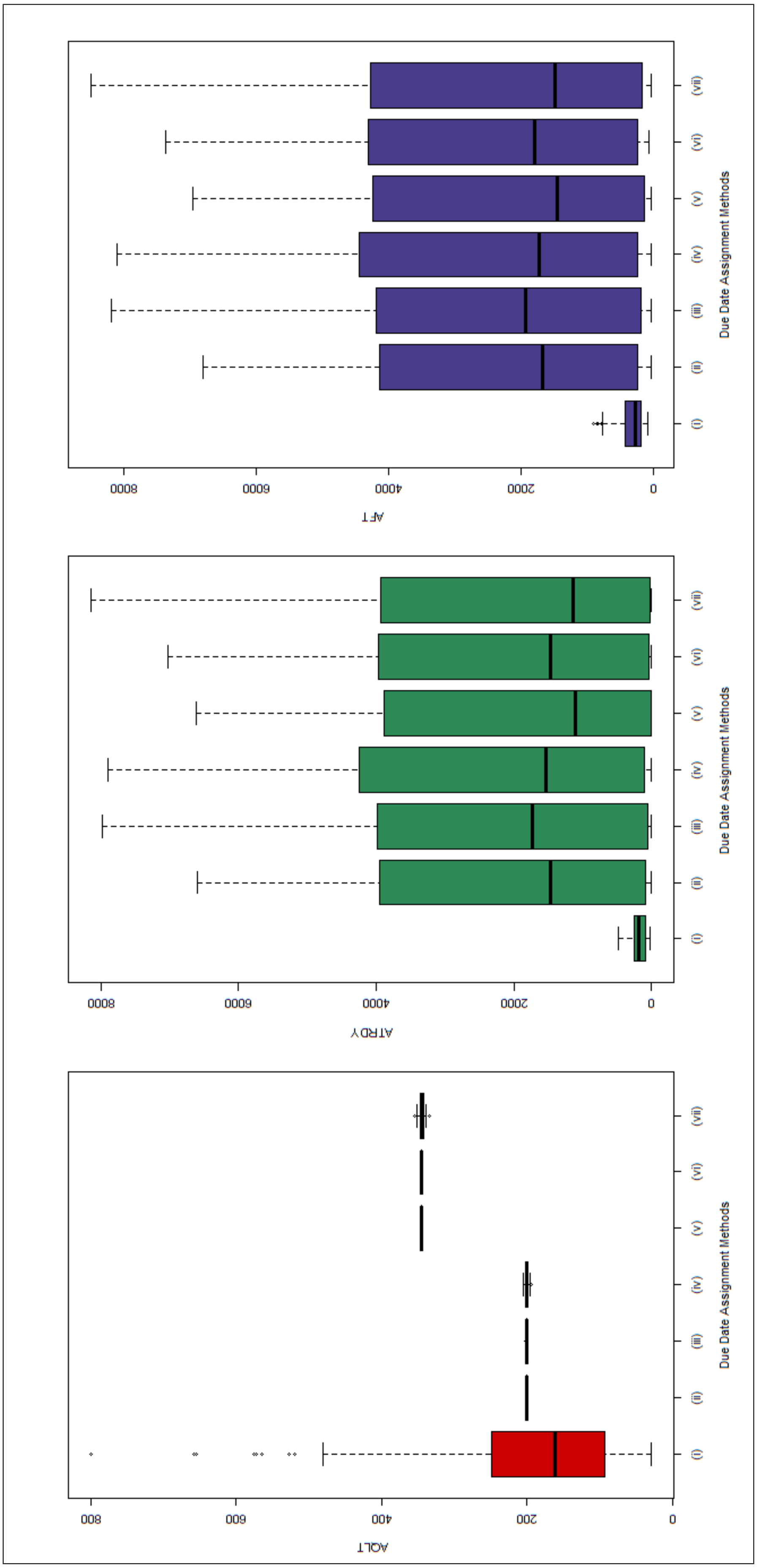


Figure 9.17. Case VIII - Box Plot.

10. CONCLUSION

Due date assignment and lead time quotation are two popular concepts of the last decades in connection with the well known inventory management and production systems such as JIT or lean manufacturing. Missing promised due dates and delivering orders afterwards cause different kinds of tardiness penalties for a company. Asssigned due dates must be as tight as possible to gain and sustain competitive power in competitive markets, yet compliance to these dates are also critical in terms of delivery reliability.

As of special interest, batch processing companies that take place in highly competitive markets are considered within the scope of the study. Such companies generally have family structures and a significant family setup time is necessary before processing a job from a different family. Thus, this is an important problem encountered in real life by most of batch processing companies.

In dynamic environments where all jobs are not readily available at the production system and under stochastic job characteristics where job related data is not known certainly, due date assignment process becomes very challenging. Moreover, assigning due dates in an online fashion at the arrival time of a job is a much more complicated process, due to integration of a new arrival into an existing schedule involving already scheduled non-complete jobs.

Instead of adding a flow allowance to job arrival times as in the majority of literature studies, a new approach is developed to assign due dates in an online fashion under family setups in dynamic and stochastic single machine environments. Single machine model is a building block for more complicated models under uncertainty.

Under the aforementioned circumstances, processing jobs according to FCFS dispatching rule and assigning a new arriving job to the first available position is not reasonable in the presense of family setups since this may lead to excessive number of setups. Instead of processing jobs based on FCFS rule, a two-phase solution methodology is proposed to be able to assign due dates for new arrivals immediately and to gain operational efficiency, simultaneously.

According to the proposed two-phase solution procedure, in the first phase, the capacity is allocated to families in a periodic, offline and forward-looking fashion, considering estimated workload and arrival pattern for families, before actual job arrivals. During the capacity allocation, family splitting is allowed through a batching strategy to enable sharing a single setup and processing a group of jobs from the same family in a row. While solving the batch allocation problem, in addition to expected potential new arrivals, non-complete jobs from previous periods whose due dates are already assigned, are also taken into account and rescheduled. Following the first phase, in the second phase, based on the constructed batch-family structure, due date of a new arriving job is assigned in an online way.

There are trade-offs to be considered during batch allocation phase. Large batches may increase machine utilization but lead to further due dates. Small batches may give rise to shorter due dates, yet cause more number of setups. Another trade-off to be considered during due date assignment phase is that a shorter due date can be assigned for a new arriving job at the expense of shifting the completion time of non-complete jobs whereas tardiness of non-complete jobs can be prevented or kept at minimum at the cost of assigning a further due date for a new arrival.

Within the framework of the solution approach, a mathematical model and a heuristic algorithm are developed for each phase. Two conflicting objectives are considered to be minimized: average quoted lead time of new arrivals and tardiness of non-complete jobs.

A simulation environment is constructed to mimic a virtual single machine shop production system so as to measure the effectiveness of the proposed solution approach. Different production environments are designed in terms of number of families, setup time structure and busyness levels. The effect of two control parameters, the length of scheduling period and tardiness coefficient that is related to total allowed tardiness on non-complete jobs, are observed. The effect of control parameters on two key performance indicators (average quoted lead time and average tardiness) and the relationship between them are examined and reported. Number of planned and realized setups, machine utilization rates and relative MIP gaps are also reported.

The inference to be drawn from the computational tests can be summarized as follows. When the length of scheduling period inclines, average quoted lead time grows due to larger batch sizes and further due dates. But the increment in the length of scheduling period does not always lead to a decrement in average tardiness. Moreover, with the increase in the length of scheduling period, number of opened batches by the batch allocation optimization model drops, as a consequence of batching. When the number of opened batches during batch allocation diminishes, average quoted lead time increases. This result reveals the claim that more number of opened batches leads to lower average quoted lead time.

When allowed total tardiness on non-complete jobs rises, it enables rescheduling of non-complete jobs at the time of a new arrival in the proposed due date assignment optimization model. By this way, lower average quoted lead times are obtained through assigning closer due dates for new arrivals, at the expense of inflating the tardiness of committed non-complete jobs through shifting their expected completion times further in the schedule. On the contrary, when allowed total tardiness on non-complete jobs drops, it ensures minimizing tardiness of non-complete jobs by the help of the proposed optimization models, at the expense of causing higher average quoted lead times. Thus, the basic trade-off between competitive power and delivery reliability is captured by the proposed optimization models.

For performance comparison, the proposed solution approach is compared by the traditional due date setting rules on a FCFS basis, with a slight modification to be able to consider family setups. The comparison is two-way. Firstly, average tardiness is observed by taking the flow allowance equal to the average quoted lead time obtained through the proposed solution methodology which is attained through simulation runs.

Secondly, average tardiness is measured by taking the flow allowance equal to the actual flow time derived from simulation. The proposed strict and flexible batching policies dramatically outperform CON, SLK and TWK due date assignment methods under both flow allowances, in terms of average tardiness.

Finally, in the presence of family setups in a production system, processing the jobs by reflecting the proposed batching structures via the proposed optimization models is significantly better than processing jobs according to FCFS dispatching rule, in terms of key performance indicators.

As a future research direction, the production environment can be extended to multi-machine setting and machine breakdown can be considered.

REFERENCES

1. Bagchi, U., Y.-L. Chang and R. S.Sullivan, "Minimizing Absolute and Squared Deviations of Completion Times with Different Earliness and Tardiness Penalties and a Common Due Date", *Naval Research Logistics*, Vol. 34, pp. 739 – 751, 1987.

2. Sen, T. and S. K. Gupta, "State-of-art Survey of Static Scheduling Research Involving Due Dates", *Omega*, Vol. 12, pp. 63 – 76, 1984.

3. Germs, R., *Controlling the Order Pool in Make-To-Order Production Systems*, University of Groningen, Groningen, the Netherlands, 2012.

4. Panwalkar, S., M. Smith and A. Seidmann, "Common Due Date Assignment to Minimize Total Penalty for the One Machine Scheduling Problem", *Operations Research*, Vol. 30, pp. 391 – 399, 1982.

5. Seidmann, A., S. Panwalkar and M. Smith, "Optimal Assignment of Due-dates for a Single Processor Scheduling Problem", *Omega*, Vol. 19, pp. 399 – 405, 1981.

6. Chand, S. and D. Chhajed, "A Single Machine Model for Determination of Optimal Due Dates and Sequence", *Operations Research*, Vol. 40, pp. 596–602, 1992.

7. Quaddus, M., "A Generalized Model of Optimal Due-Date Assignment by Linear Programming", *The Journal of the Operational Research Society*, Vol. 38, pp. 353–359, 1987.

8. Baker, K. R. and G. D. Scudder, "On the Assignment of Optimal Due Dates", *The Journal of the Operational Research Society*, Vol. 40, pp. 93 – 95, 1989.

9. Koulamas, C., "A Faster Algorithm for a Due Date Assignment Problem with Tardy Jobs", *Operations Research Letters*, Vol. 38, pp. 127–128, 2010.

10. Shabtay, D. and G. Steiner, "Two Due Date Assignment Problems in Scheduling

a Single Machine", *Operations Research Letters*, Vol. 34, pp. 683 – 691, 2006.

11. Koulamas, C., "A Unified Solution Approach for the Due Date Assignment Problem with Tardy Jobs", *International Journal of Production Economics*, Vol. 132, pp. 292 – 295, 2011.

12. Shabtay, D., "Due Date Assignments and Scheduling a Single Machine with a General Earliness/tardiness Cost Function", *Computers & Operations Research*, Vol. 35, pp. 1539–1545, 2008.

13. Koulamas, C. and G. Kyparisis, "Single-machine Scheduling with Waiting-time-dependent Due Dates", *European Journal of Operational Research*, Vol. 191, pp. 577–581, 2008.

14. Tuong, N. and A. Soukhal, "Due Dates Assignment and JIT Scheduling with Equal-size Jobs", *European Journal of Operational Research*, Vol. 205, pp. 280–289, 2010.

15. Mosheiov, G. and U. Yovel, "Minimizing Weighted Earliness Tardiness and Due Date Cost with Unit Processing Time Jobs", *European Journal of Operational Research*, Vol. 172, pp. 528 – 544, 2006.

16. Zhang, R. and C. Wu, "A Double-layered Optimisation Approach for the Integrated Due Date Assignment and Scheduling Problem", *International Journal of Production Research*, Vol. 50, pp. 5 – 22, 2012.

17. Gerstl, E. and G. Mosheiov, "Minmax Due-date Assignment with a Time Window for Acceptable Lead-times", *Annuals of Operations Research*, Vol. 211, pp. 167 – 177, 2013.

18. Qi, X., G. Yu and J. Bard, "Single Machine Scheduling with Assignable Due Dates", *Discrete Applied Mathematics*, Vol. 122, pp. 211–233, 2002.

19. Shabtay, D., G. Steiner and L. Yedidsion, "Bicriteria Problems to Minimize Max-

imum Tardiness and Due Date Assignment Cost in Various Scheduling Environments", *Discrete Applied Mathematics*, Vol. 158, pp. 1090–1103, 2010.

20. Shabtay, D. and G. Steiner, "Optimal Due Date Assignment in Multi-machine Scheduling Environments", *Journal of Scheduling*, Vol. 11, pp. 217–228, 2008.

21. Proceedings of the American Control Conference, *Stochastic Single Machine Scheduling with Earliness and Tardiness Penalties and Proportional Weights*, May 2002.

22. Wu, X. and X. Zhou, "Stochastic Scheduling to Minimize Expected Maximum Lateness", *European Journal of Operational Research*, Vol. 190, pp. 834–850, 1997.

23. Portougal, V. and D. Trietsch, "Setting Due Dates in a Stochastic Single Machine Environment", *Computers & Operations Research*, Vol. 33, pp. 1681 – 1694, 2006.

24. Soroush, H. M., "Sequencing and Due-date Determination in the Stochastic Single Machine Problem with Earliness and Tardiness Costs", *European Journal of Operational Research*, Vol. 113, pp. 450 – 468, 1999.

25. Kaplan, A. C. and A. T. Unal, "A Probabilistic Cost-based Due Date Assignment Model for Job Shops", *International Journal of Production Research*, Vol. 31, pp. 2817 – 2834, 1993.

26. Wang, J. and Q. Guo, "A Due-date Assignment Problem with Learning Effect and Deteriorating Jobs", *Applied Mathematical Modelling*, Vol. 34, pp. 59–66, 2010.

27. Qian, J. and G. Steiner, "Fast Algorithms for Scheduling with Learning Effects and Time-dependent Processing Times on a Single Machine", *European Journal of Operational Research*, Vol. 225, pp. 547 – 551, 2013.

28. Kuo, W. and D. Yang, "A Note on Due-date Assignment and Single-machine

Scheduling with Deteriorating Jobs", *Journal of the Operational Research Society*, Vol. 59, pp. 857–859, 2008.

29. Cheng, T., L. Kang and C. Ng, "Due-Date Assignment and Single Machine Scheduling with Deteriorating Jobs", *The Journal of the Operational Research Society*, Vol. 55, pp. 198 – 203, 2004.

30. Li, S., C. T. Ng and J. Yuan, "Scheduling Deteriorating Jobs with CON/SLK Due Date Assignment on a Single Machine", *International Journal of Production Economics*, Vol. 131, pp. 747 – 751, 2011.

31. Yin, Y., M. Liu, T. Cheng, C.-C. Wu and S.-R. Cheng, "Four Single-machine Scheduling Problems Involving Due Date Determination Decisions", *Information Sciences*, Vol. 251, pp. 164 – 181, 2013.

32. Xia, Y., B. Chen and J. Yue, "Job Sequencing and Due Date Assignment in a Single Machine Shop with Uncertain Processing Times", *European Journal of Operational Research*, Vol. 184, pp. 63–75, 2008.

33. Gordon, V. and V. Strusevich, "Single Machine Scheduling and Due Date Assignment with Positionally Dependent Processing Times", *European Journal of Operational Research*, Vol. 198, pp. 57–62, 2009.

34. Hsu, C., S. Yang and D. Yang, "Two Due Date Assignment Problems with Position-dependent Processing Time on a Single-machine", *Computers & Industrial Engineering*, Vol. 60, pp. 796–800, 2011.

35. Shabtay, D. and G. Steiner, "The Single-machine Earliness-tardiness Scheduling Problem with Due Date Assignment and Resource-dependent Processing Times", *Annals of Operations Research*, Vol. 159, pp. 25–40, 2008.

36. Seo, D. K., C. M. Klein and W. Jang, "Single Machine Stochastic Scheduling to Minimize the Expected Number of Tardy Jobs using Mathematical Programming

Models", *Computers & Industrial Engineering*, Vol. 48, pp. 153 – 161, 2005.

37. Cai, X. and X. Zhou, "Single-Machine Scheduling with Exponential Processing Times and General Stochastic Cost Functions", *Journal of Global Optimization*, Vol. 31, pp. 317 – 332, 2005.

38. Gordon, V., V. Strusevich and A. Dolgui, "Scheduling with Due Date Assignment under Special Conditions on Job Processing", *Journal of Scheduling*, Vol. 15, pp. 447 – 456, 2012.

39. Baker, K. and D. Trietsch, "Safe Scheduling: Setting Due Dates in Single-machine Problems", *European Journal of Operational Research*, Vol. 196, pp. 69–77, 2009.

40. Baker, K. and D. Trietsch, "Safe Scheduling", *Tutorials in Operations Research, INFORMS*, pp. 79 – 101, 2007.

41. Elyasi, A. and N. Salmasi, "Due Date Assignment in Single Machine with Stochastic Processing Times", *International Journal of Production Research*, Vol. 51, pp. 2352 – 2362, 2013.

42. Li, C. and T. Cheng, "Due-date Determination with Resequencing", *IIE Transactions*, Vol. 31, pp. 183 – 188, 1999.

43. Duron, C., M. Louly and J. Proth, "The One Machine Scheduling Problem: Insertion of a Job Under the Real-time Constraint", *European Journal of Operational Research*, Vol. 199, pp. 695–701, 2009.

44. Bansal, N., H. Chan and K. Pruhs, "Competitive Algorithms for Due Date Scheduling", *Algorithmica*, Vol. 59, pp. 569–582, 2011.

45. Dewan, P. and S. Joshi, "Dynamic Single-machine Scheduling under Distributed Decision-making", *International Journal of Production Research*, Vol. 38, pp. 3759 – 3777, 2000.

46. Vig, M. and K. Dooley, "Dynamic Rules for Due Date Assignment", *International Journal of Production Research*, Vol. 29, pp. 1361 – 1377, 1991.

47. Cheng, T., Z.-L. Chen and N. V. Shakhlevich, "Common Due Date Assignment and Scheduling with Ready Times", *Computers & Operations Research*, Vol. 29, pp. 1957 – 1967, 2002.

48. Baker, K. and J. Bertrand, "A Dynamic Priority Rule for Scheduling Against Due-Dates", *Journal of Operations Management*, Vol. 3, pp. 37–42, 1982.

49. Hopp, W. and M. Sturgis, "Quoting Manufacturing Due Dates Subject to a Service Level Constraint", *IIE Transactions*, Vol. 32, pp. 771 – 784, 2000.

50. Sridharan, V. and X. Li, "Improving Delivery Reliability by a New Due-date Setting Rule", *European Journal of Operational Research*, Vol. 186, pp. 1201–1211, 2008.

51. Hopp, W. J. and M. R. Sturgis, "A Simple, Robust Lead Time-quoting Policy", *Manufacturing & Service Operations Management*, Vol. 3, pp. 321 – 336, 2001.

52. Savas, S., P. Griffin and P. Keskinocak, "Dynamic Lead-time Quotation for an M/M/1 Base-stock Inventory Queue", *Operations Research*, Vol. 58, pp. 383–395, 2010.

53. Wein, L., "Due-Date Setting and Priority Sequencing in a Multiclass M/G/1 Queue", *Management Science*, Vol. 37, pp. 693–700, 2010.

54. Duenyas, I., "Single Facility Due Date Setting with Multiple Customer Classes", *Management Science*, Vol. 41, pp. 608 – 619, 1995.

55. Kaminsky, P. and Z. Lee, "Effective On-line Algorithms for Reliable Due Date Quotation and Large-scale Scheduling", *Journal of Scheduling*, Vol. 11, pp. 187–204, 2008.

56. Shabtay, D., "Scheduling and Due Date Assignment to Minimize Earliness, Tardiness, Holding, Due Date Assignment and Batch Delivery Costs", *International Journal of Production Economics*, Vol. 123, pp. 235–242, 2010.

57. Chen, Z., "Scheduling with Batch Setup Times and Earliness-tardiness Penalties", *European Journal of Operational Research*, Vol. 96, pp. 518 – 537, 1997.

58. Cheng, T. and V. Gordon, "Batch Delivery Scheduling on a Single Machine", *Journal of the Operational Research Society*, Vol. 45, pp. 1211–1216, 1994.

59. Yin, Y., T. C. E. Cheng, D. Xu and C.-C. Wu, "Common Due Date Assignment and Scheduling with a Rate-modifying Activity to Minimize the Due Date, Earliness, Tardiness, Holding, and Batch Delivery Cost", *Computers & Industrial Engineering*, Vol. 63, pp. 223 – 234, 2012.

60. Chrétienne, P., O. Hazir and S. Kedad-Sidhoum, "Integrated Batch Sizing and Scheduling on a Single Machine", *Journal of Scheduling*, Vol. 14, pp. 541–555, 2011.

61. Unal, A. T. and A. S. Kiran, "Batch Sequencing", *IIE Transactions*, Vol. 24, pp. 73–83, 1992.

62. Cheng, T. and M. Kovalyov, "Batch Scheduling and Common Due-date Assignment on a Single Machine", *Discrete Applied Mathematics*, Vol. 70, pp. 231–245, 1996.

63. Kovalyov, M. Y., "Batch Scheduling and Common Due Date Assignment Problem: an NP-hard Case", *Discrete Applied Mathematics*, Vol. 80, pp. 251 – 254, 1997.

64. Park, T., T. Lee and C. O. Kim, "Due-date Scheduling on Parallel Machines with Job Splitting and Sequence-dependent Major/minor Setup Times", *The International Journal of Advanced Manufacturing Technology*, Vol. 59, pp. 325 – 333, 2012.

65. Pan, J. C.-H., J.-S. Chen and H.-L. Cheng, "A Heuristic Approach for Single-machine Scheduling with Due Dates and Class Setups", *Computers & Operations Research*, Vol. 28, pp. 1111–1130, 2001.

66. Azizoglu, M. and S. Webster, "Scheduling Job Families about an Unrestricted Common Due Date on a Single Machine", *International Journal of Production Research*, Vol. 35, pp. 1321 – 1330, 1997.

67. Kim, J.-G. and D.-H. Lee, "Algorithms for Common Due-date Assignment and Sequencing on a Single Machine with Sequence-dependent Setup Times", *Journal of the Operational Research Society*, Vol. 60, pp. 1264 – 1272, 2009.

68. Ghosh, J. B., "Batch Scheduling to Minimize Total Completion Time", *Operations Research Letters*, Vol. 16, pp. 271 – 275, 1994.

69. APIEMS 2008, Proceedings of the 9th Asia Pasific Industrial Engineering & Management Systems Conference, *Batch Scheduling for a Single Machine Processing Parts of a Single Item with Increasing Processing Time to Minimize Total Actual Flow Time*, December 2008.

70. Schutten, J. M. J., S. L. van de Velde and W. H. M. Zijm, "Single-machine Scheduling with Release Dates, Due Dates and Family Setup Times", *Management Science*, Vol. 42, pp. 1165–1174, 1996.

71. Unal, A. T., R. Uzsoy and A. Kiran, "Rescheduling on a Single Machine with Part-type Dependent Setup Times and Deadlines", *Annals of Operations Research*, Vol. 70, pp. 93–113, 1997.

72. Ang, A., A. Sivakumar and C. Qi, "Criteria Selection and Analysis for Single Machine Dynamic On-line Scheduling with Multiple Objectives and Sequence-dependent Setups", *Computers & Industrial Engineering*, Vol. 56, pp. 1223–1231, 2009.

73. Vieira, G., J. Herrmann and E. Lin, "Analytical Models to Predict the Performance of a Single-machine System under Periodic and Event-driven Rescheduling Strategies", *International Journal of Production Research*, Vol. 38, pp. 1899–1915, 2000.

74. Germs, R. and N. D. van Foreest, "Order Acceptance and Scheduling Policies for a Make-to-order Environment with Family-dependent Lead and Batch Setup Times", *International Journal of Production Research*, Vol. 51, pp. 940 – 951, 2013.

75. Chen, D., S. Li and G. Tang, "Single Machine Scheduling with Common Due Date Assignment in a Group Technology Environment", *Mathematical and Computer Modelling*, Vol. 25, pp. 81 – 90, 1997.

76. Shabtay, D., Y. Itskovich, L. Yedidsion and D. Oron, "Optimal Due Date Assignment and Resource Allocation in a Group Technology Scheduling Environment", *Computers & Operations Research*, Vol. 37, pp. 2218 – 2228, 2010.

77. Li, S., C. T. Ng and J. Yuan, "Group Scheduling and Due Date Assignment on a Single Machine", *International Journal of Production Economics*, Vol. 130, pp. 230 – 235, 2011.

78. Baker, K. and J. Bertrand, "A Comparison of Due-Date Selection Rules", *AIIE Transactions*, Vol. 13, pp. 123–131, 1981.

79. Gordon, V., J. Proth and C. Chu, "A Survey of the State-of-the-art of Common Due Date Assignment and Scheduling Research", *European Journal of Operational Research*, Vol. 139, pp. 1–25, 2002.

80. Baker, K. R. and J. W. M. Bertrand, "An Investigation of Due-Date Assignment Rules with Constrained Tightness", *Journal of Operations Management*, Vol. 1, pp. 109 – 120, 1981.

81. Gordon, V., J. Proth and C.Chu, "Due Date Assignment and Scheduling: SLK, TWK and Other Due Date Assignment Models", *Production Planning & Control: The Management of Operations*, Vol. 13, pp. 117–132, 2002.

82. Cheng, T., "Optimal Due-Date Determination and Sequencing of n Jobs on a Single Machine", *The Journal of the Operational Research Society*, Vol. 35, pp. 433–437, 1984.

83. Gordon, V., J.-M. Proth and V. Strusevich, "Scheduling with Due Date Assignment", J. Y.-T. Leung (Editor), *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, Chapman & Hall/CRC Computer and Information Science Series, chap. 21, pp. 21–1 – 21–22, CRC Press LLC, Boca Raton, Florida, 2004.

84. Adamopoulos, G. I. and C. P. Pappis, "Single Machine Scheduling with Flow Allowances", *The Journal of the Operational Research Society*, Vol. 47, pp. 1280 – 1285, 1996.

85. Moses, S., H. Grant, L. Gruenwald and S. Pulat, "Real-time Due-date Promising by Build-to-order Environments", *International Journal of Production Research*, Vol. 42, pp. 4353 – 4375, 2004.

86. Pinedo, M., *Scheduling: Theorem, Algorithms, and Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 2002.

87. Yang, W., "Survey of Scheduling Research Involving Setup Times", *International Journal of Systems Science*, Vol. 30, pp. 143–155, 1999.

88. Allahverdi, A., J. Gupta and T. Aldowaisan, "A Review of Scheduling Research Involving Setup Considerations", *Omega, The International Journal of Management Science*, Vol. 27, pp. 219–239, 1999.

89. Allahverdi, A., C. Ng, T. Cheng and M. Kovalyov, "A Survey of Scheduling

Problems with Setup Times or Costs", *European Journal of Operational Research*, Vol. 187, pp. 985–1032, 2008.

90. Potts, C. and M. Kovalyov, "Scheduling with Batching: A Review", *European Journal of Operational Research*, Vol. 120, pp. 228 – 249, 2000.

91. Cheng, T. C. E. and M. Gupta, "Survey of Scheduling Research Involving Due Date Determination Decisions", *European Journal of Operational Research*, Vol. 38, pp. 156 – 166, 1989.

92. Monma, C. L. and C. N. Potts, "On the Complexity of Scheduling with Batch Setup Times", *Operations Research*, Vol. 37, pp. 798 – 804, 1989.

93. Kaminsky, P. and D. Hochbaum, "Due Date Quotation Models and Algorithms", J. Y.-T. Leung (Editor), *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, Chapman & Hall/CRC Computer and Information Science Series, chap. 20, pp. 20–1 – 20–22, CRC Press LLC, Boca Raton, Florida, 2004.

94. Keskinocak, P. and S. Tayur, "Due–date management policies", D. Simchi-Levi, S. Wu and Z. Shen (Editors), *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, International Series in Operations Research and Management Science, pp. 485 – 553, Kluwer Academic Publishers, Norwell, 2003.

95. Potts, C. N. and L. N. V. Wassenhove, "Integrating Scheduling with Batching and Lot-Sizing: A Review of Algorithms and Complexity", *The Journal of the Operational Research Society*, Vol. 43, pp. 395 – 406, 1992.

96. Baker, K., *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York, 1974.

97. Banks, J., J. S. C. II, B. L. Nelson and D. M. Nicol, *Discrete-Event System Simulation*, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2005.

98. Ross, S. M., *Introduction to Probability Models*, Academic Press, San Diego, 2007.

99. Shell, R. L. and E. L. Hall, *Handbook of Industrial Automation*, Marcel Dekker, New York, 2000.

100. *ILOG CPLEX 10.0 User's Manual*, 2006.