공학 석사 학위논문

Simulation-based Decision Support Framework for Hybrid Flow Shop (HFS) Scheduling Problem

하이브리드 플로우 샵 (HFS) 스케줄링 문제의 시뮬레이션 기반 의사 결정 지원 프레임 워크에 관한 연구

2017 년 8 월

서울대학교 대학원

산업공학과 전공

SAHA SUDIPTA

Abstract

Simulation-based Decision Support Framework for Hybrid Flow Shop (HFS) Scheduling Problem

SAHA SUDIPTA
Department of Industrial Engineering
The Graduate School
Seoul National University

Manufacturing environments has become very complicated nowadays. They consist of hundreds of job varieties, diverse types of machines with complex architectural layouts. Hybrid flow shop (HFS) is one of them. Although there is no exact definition of HFS but flow shops with multiple parallel machines at each stage are referred as HFS in general. However, the characteristics of a hybrid flow shop might differ according to the particular production environment. HFS production scheduling is one of the most complex combinatorial problems encountered in many real world industries. Given HFS's complexity and importance, most of the literatures on HFS scheduling seem to focus on mono-criteria objectives which is sometimes quite unrealistic. Real world HFS scheduling problem involves several performance measures as objective functions, which eventually can often conflict and compete for decision making.

Industries have been using simulation extensively to model and analyze the impact of such variabilities on production system behavior and to explore several ways of coping under any changes or uncertainties. Simulation flexibility may help to find better or optimal solutions to a number of complex problems of HFS. The HFS scheduling problem requires all activities to be considered. Even though simulation is a good tool, there is one more aspect to be considered on using simulation. Almost each and every level of employees needs to be skilled enough with simulation software to deal with HFS scheduling problems. But not all of them are fully capable to utilize the simulation system. Inadequate capability of personnel to utilize simulation effectively can only be overcome if we can design custom interfaces and integrate flexible simulation framework with supportive programs.

In this study, a flexible ‘Simulation modeling framework’ is proposed to mimic HFS systems. This research analyzes the impact of different combinations of commonly used job sequencing and dispatching policies for multiple performance measures. A heuristic is also proposed to reduce the number of comparisons thus to reduce the number of simulation runs. By implementing the proposed heuristics, better combinations of dispatching policies are found each of the performance measure considered. In the end, an analysis is shown regarding the impact of varying batch size on certain HFS’s performance measures.

Keywords: Hybrid flow shop, Simulation, ispatching rules, Heuristic, Varying batch size

Student Number: 2015 - 23296

Contents

List of Tables

List of Figures

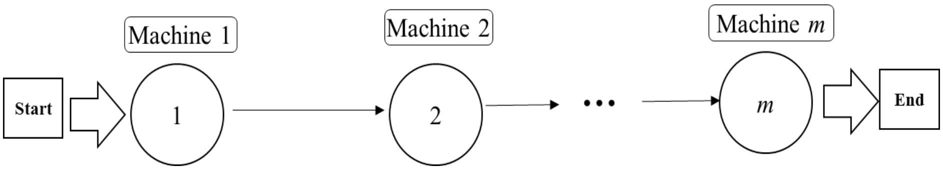
Chapter 1. Introduction

1.1 Background

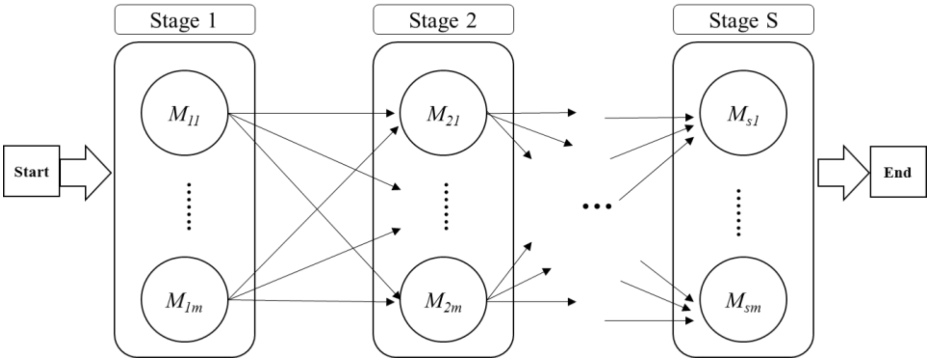
In a design when jobs flow from an initial machine, through several intermediate machines and ultimately to a final machine before completion, traditionally referred to as a flow shop. A pure flow shop model consists of m different machines; thus each job consists m operations. The operations of job j can be numbered (1, j), (2, j), ……, (m, j), so that they correspond to the machine required [Baker & Trietsch, 2013]. Whereas in a hybrid flow shop (HFS), a set of n jobs are to be processed in a series of S stages optimizing a given objective function [Ruiz & Vázquez-Rodríguez, 2010]. There are a number of variants, but some of the characteristics are in common such as:

1) The number of processing stages S is at least 2
2) Each stage s has $M_s \geq 1$ machines in parallel and in at least one of the stages $M_s > 1$
3) All jobs are processed following the same production flow: stage 1, stage 2, …., stage S. A job might skip any number of stages provided it is processed in at least one of them
4) Each job j requires a processing time p_{js} in stage s. Processing of job j in stage s is referred as operation o_{js}

[Figure 1.1] and [Figure 1.2] shows the basic architectural difference between flow shop and HFS.



[Figure 1.1] Flow shop configuration



[Figure 1.2] Hybrid flow shop configuration

From the computational point of view, scheduling is one of the hard optimization problems found in real industrial contexts. Among the various types of scheduling problems, flexible or hybrid flow shop scheduling problem (HFSP) is one of the most challenging [Montoya-Torres, Solano-Charris, & Muñoz-Villamizar, 2016]. This is considered to be NP-hard optimization problem, even for the case of a system with only two processing stages in which one stage contains two machines and the other stage contains a single machine [J. N. Gupta, 1988].

In complex production shops like hybrid flow shop or hybrid job shop, production scheduling usually takes place by simple FIFO or by any random scheduling strategies decided by the shop floor manager. But it is not found to be the efficient strategy all the time. A more detailed analysis is obvious in this respect to find out more efficient scheduling strategy to enhance the shop's performances.

Extensive research has been conducted on scheduling, especially in job shop and flow shop settings. On contrary, very little research has been done on hybrid flow shop systems, even though they are found in many industries, including beer processing, glass container product ion, petroleum refining, cable production, fertilizer production etc.

1.2 Motivation

This research is motivated from a real world job scheduling problem in an *Optical Lens Processing Industry,*[1] which belongs to HFS environment.

The flow shop consists of multiple lens types, multiple stages with parallel machines at each stage, variations in lenses' processing cycle time and different processing sequences. In such a HFS environment production managers need to deal with some key issues such as:

1) How to schedule those job orders in order to meet the deadline?
2) What would be the appropriate job sequencing strategy to achieve better performances from the HFS?
3) What kind of solution approach should be followed to deal with the scheduling problem along with uncertainties in the HFS?

Manufacturing management level needs to be well equipped with flexible tools due to rapidly changing nature of today's production environments. Newly invented hardware, software items, tailored for specific applications are being developed every day. But success in reducing expenses, increasing the efficiency, improving the performance is not easy until the application based integration of such components is achieved. The flexible or hybrid flow shop system with a concrete system model along with use of information technologies necessitates such requirements [Yücel, Şen, & Kılıç, 2004].

In this study, a flexible computer simulation study is performed for a pilot Optical lens processing layout to investigate the usefulness of simulation into hybrid flow shop. The complete integration of current processing system is not implemented fully into the simulation model as it is proposed as a future work. The simulation is done by using *Arena v14.70* integrated with *SIMAN*. The

---

[1] To protect proprietary information, the company's definitions are not provided here.

model customization part is accomplished by *Visual Basic for Applications.*

In addition, multiple performance measures are taken into account of the existing HFS under different job priority dispatching rules and a comparison heuristic is discussed to find out the best combinations of those policies for each performance measure. At last, the impact of varying batch size on HFS's performance measures is shown.

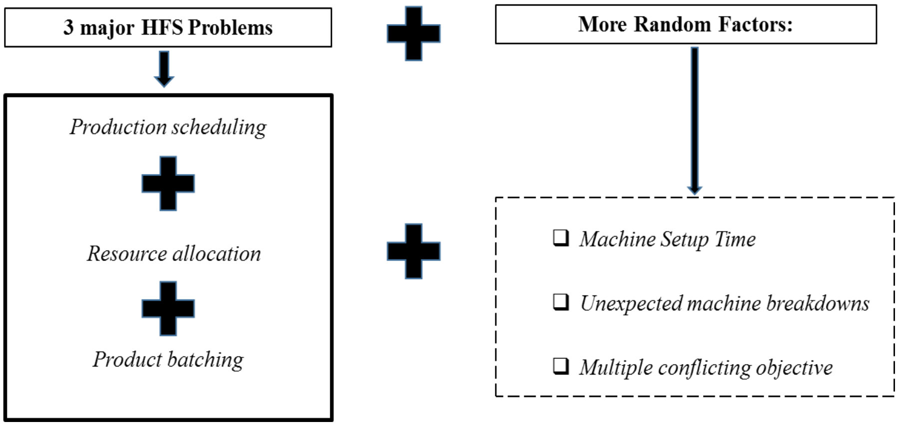
1.3 Outline of the thesis

This thesis is organized as follows: Chapter 2 concentrates on problem definitions. Chapter 3 discusses about the literatures based on HFSP and solution approaches. Chapter 4 elaborates the research goals and objectives. Chapter 5 talks about the development of a generic flexible simulation model framework for real time decision making. Chapter 5 examines the feasibility of the proposed methodology by experimenting for a case study involving an *Optical lens processing system.* Chapter 6 analyzes the simulation result based on the objectives. Chapter 7 ends with conclusion, limitation and future work of the thesis.

Chapter 2. Literature Review

2.1 Problems in HFS

According to [Al-Turki, Saleh, Deyab, & Almoghathawi, 2012],as shown in [Figure 2.1], resource allocation, product batching and production scheduling are three different problems in manufacturing systems of different structures like HFS manufacturing system. Usually such problems are handled independently for a certain objective function which is related to the efficiency and effectiveness of the particular production system. Integrated handling of such problems is a great challenge faced by many real world manufacturing systems. Dynamic or random arrival of jobs, machine setup time requirement, unexpected machine breakdown time consideration, multiple objective functions often increase the complications of the HFS. In practice, production scheduling, resource allocation and batching decisions are integrated to each other. However, in literature they have been dealt as separate optimization problems. Solving those integrated problems at a time is difficult especially in complex models like HFS.



[Figure 2.1] Major hybrid flow shop problems

2.2 HFS scheduling problem (HFSP)

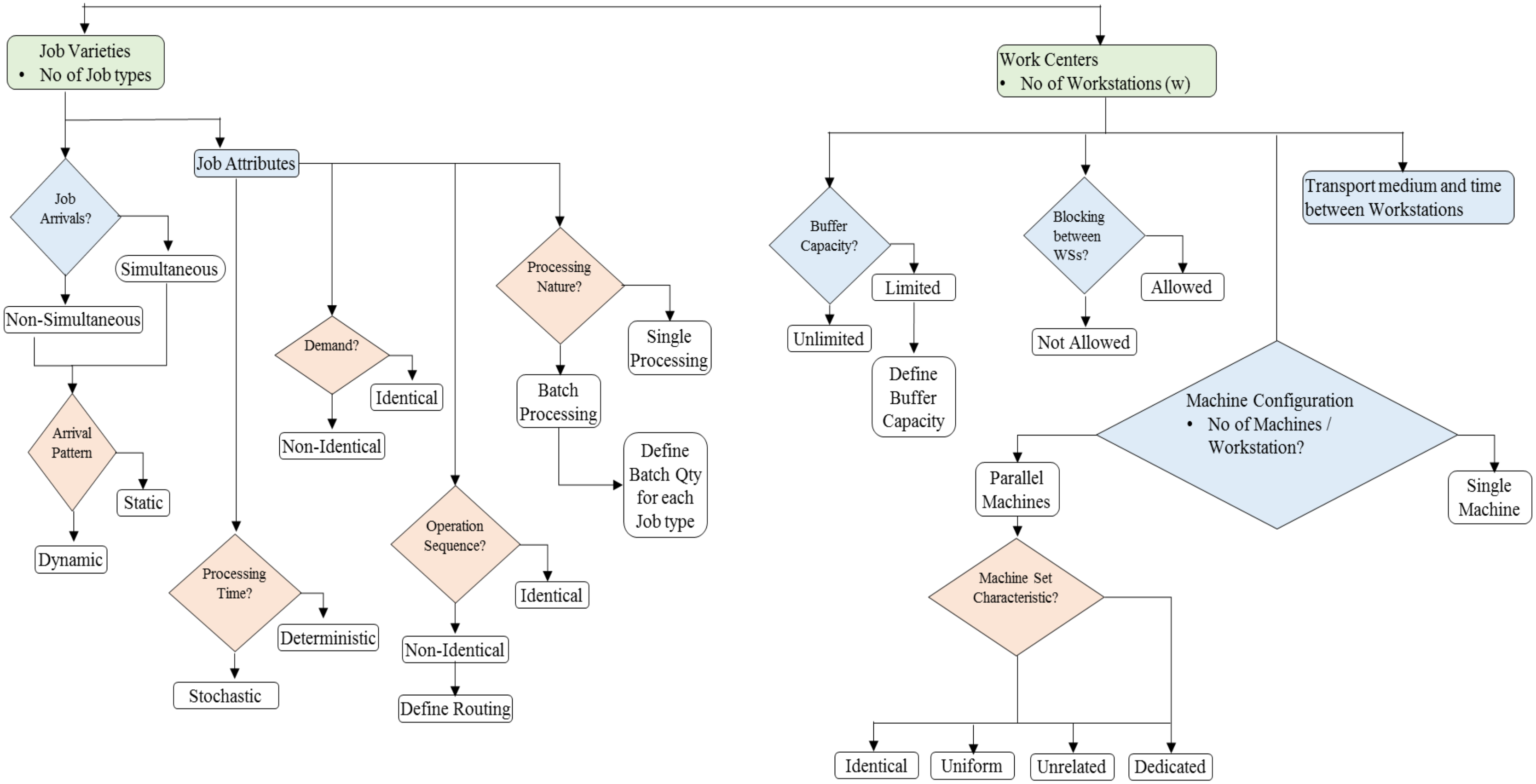
As described in [Section 2.1], production scheduling is one of the problems that can arise in a HFS. Scheduling problem is among the most difficult problems of resolution [Morais & Moccellin, 2010]. Scheduling problem consists of determining the order or sequence in which the machines will process the jobs so as to optimize some measure of performance [L. A. Johnson & Montgomery, 1974]. According to [Pinedo, 2002], scheduling goal is to optimize one or more objectives. [de Fatima Morais, Boiko, dos Santos Coelho, da Rocha, & Paraíso, 2014] said that production scheduling is always carried out in order to reach a criterion or set a performance criteria that characterize the nature of the scheduling problem.

Hybrid flow shop scheduling (HFS) was first proposed by [Salvador, 1973]. Generally HFS is proven to be NP-Hard problem [J. N. Gupta, 1988].

NP-hardness of the HFS scheduling problem means that large-sized problem instances cannot be solved in an exact (optimal) manner within a reasonable amount of time [Montoya-Torres et al., 2016].

2.2.1 HFS classifications

Through the past five decades of literatures HFS has been classified into many ways. Based on [de Fatima Morais et al., 2014], [Ruiz & Vázquez-Rodríguez, 2010], [Ribas, Leisten, & Framiñan, 2010], [Vignier, Billaut, & Proust, 1999], [Linn & Zhang, 1999], [Pinedo, 2008], [Baker & Trietsch, 2013], [Figure 2.2] represents major classifications possible for a HFS.



[Figure 2.2] Classifications of HFS

According to [Burtseva, Parra, & Yaurima, 2010], the possible parallel machine set environments at each stage of a HFS can be of different types such as:

- Identical (ID): When jobs can be processed by any of the available parallel machines.
- Uniform (UN): When available parallel machines have different speeds; a job can be processed by any machine of the set but its processing time would be proportional to the machine speed.
- Unrelated (UR): When processing time of all jobs are arbitrary and does not really depend on the machine characteristics.
- Dedicated (DED): When parallel machines in the set are dedicated to perform specific subsets of jobs.

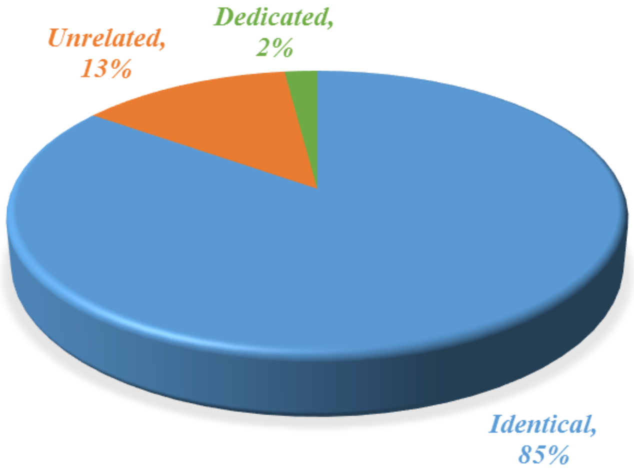
[Ruiz & Vázquez-Rodríguez, 2010] presented a percentage of the reviewed papers according to the number of stages and types of machine set environment of parallel machines as shown in [Table 2.1].

[Table 2.1] Survey of papers considering different machine environment types

Number of stages	**Type of machine environment**			**Total**
	ID	***UN***	***UR***	
2	25.12%	1.86%	4.65%	31.63%
3	4.19%	1.40%	0.00%	5.59%
s	54.41%	1.40%	6.97%	62.78%
Total	83.72%	4.66%	11.62%	100.00%

From their review work, it’s clearly seen that a large percentage of the studied papers considering identical machines at each stage (83.72%), whereas only few of the literatures (6.97%) tackled s-stage problems with unrelated parallel machine set type at each stage. It is well-known that s-stage problem with unrelated machine is most likely to be found in practice.

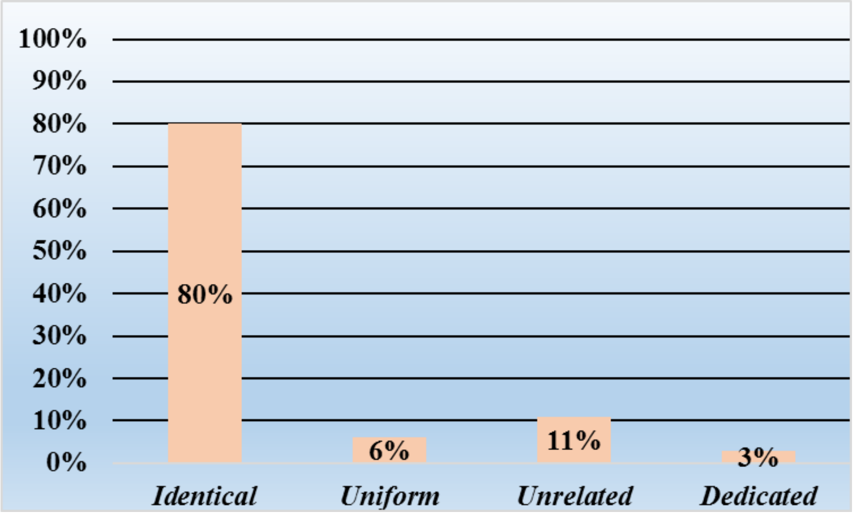
[de Fatima Morais et al., 2014] also surveyed HFS papers recently and presented a similar survey outcome as shown in [Figure 2.3].



[Figure 2.3] Distribution of parallel machine set type used in literature

Most of the papers (85%) adopted the identical machine environment. On the other hand, 13% papers considered the unrelated machine environment. From the [Figure 2.3], it's also seen that only 2% papers adopted the dedicate machine set phenomena.

Although since 2014, many researches have been presented based on different machine set characteristics. By considering the work presented in [Linn & Zhang, 1999], [Vignier et al., 1999], [Ribas et al., 2010], [Wang, 2011], [de Fatima Morais et al., 2014] and many other papers in recent times, an overall distribution of papers consisting different machine set environments is shown in [Figure 2.4]; where it's found that 80% of the papers to be based on identical machine. In contrary, only 6% researches dealt with uniform machines. The unrelated machine set environment is revealed to be around 11% and only a tiny portion (3%) of the papers considered dedicated machine. To the best of this study's knowledge almost none considered multiple machine type environment at a time in a single research.



[Figure 2.4] Distribution of parallel machine set type used in literature so far

2.2.2 Performance criteria

Based on [Ruiz-Diaz & French, 1982], [Bedworth & Bailey, 1987], [Maccarthy & Liu, 1993), [Pinedo, 2008], [Baker & Trietsch, 2013], [Jun & Park, 2015], [Moon, Lee, & Bae, 2008], [Morton & Pentico, 1993] and many other sources, [Table 2.2] presents a summary of different kinds of performance criteria considered so far for hybrid flow scheduling problem.

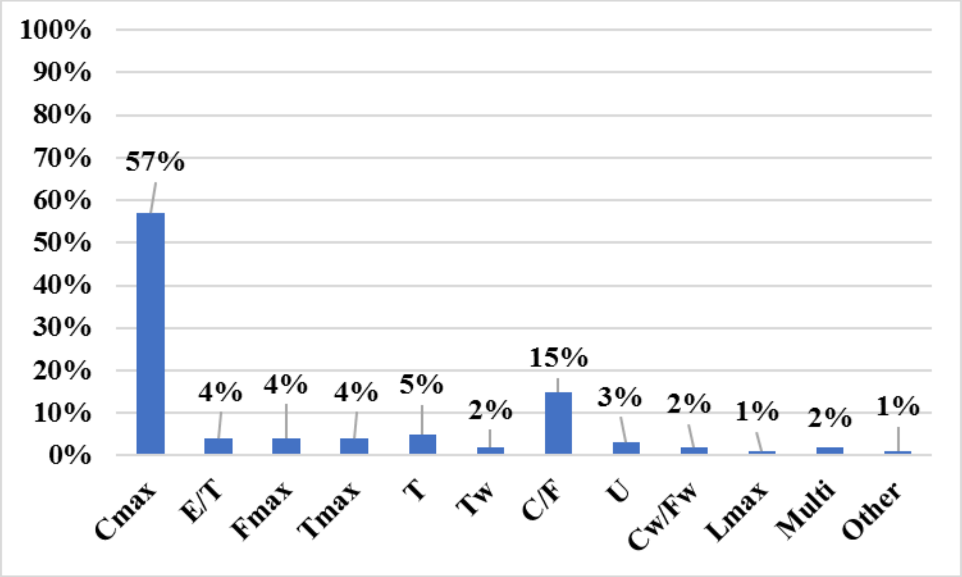
According to [Baker, 1974], all these performance criteria eventually relate to the three major types of decision making issues such as:

I. Efficient use of available resources
II. Rapid response to the demand
III. Adaptation to the prescribed deadline of a job

[Table 2.2] Performance Criteria adopted in literature for HFSP

No	Notation	Description	No	Notation	Description
1	C_{max}	Makespan	14	ΣLj/n	Mean Lateness
2	$\Sigma C_j/n$	Mean Completion Time	15	T_j	Tardiness of Job
3	ΣC_j	Total Completion Time	16	ΣT_j	Total Tardiness
4	$\Sigma w_j C_j$	Weighted Completion Time	17	$\Sigma w_j T_j$	Weighted Total Lateness
5	ΣE_j	Total Earliness	18	T_{max}	Tardiness Maximum
6	$\Sigma E_j/n$	Mean Earliness	19	$\Sigma T_j/n$	Mean Tardiness
7	$\Sigma w_j E_j$	Weighted Total Earliness	20	ΣU_j	Number of Late Jobs
8	F_j	Flow Time of Job	21	$\Sigma U_j/n$	Mean Number of Late Jobs
9	$\Sigma F_j/n$	Mean Flow Time	22	ΣW_j	Total Time to Wait
10	$\Sigma w_j F_j$	Weighted Total Flow Time	23	W_{max}	Wait Time Maximum
11	ΣL_j	Total Lateness	24	$\Sigma W_j/n$	Mean Time to Wait
12	L_j	Lateness of Job	25	$\Sigma w_j W_j$	Weighted Total Time to Wait
13	L_{max}	Lateness Maximum			

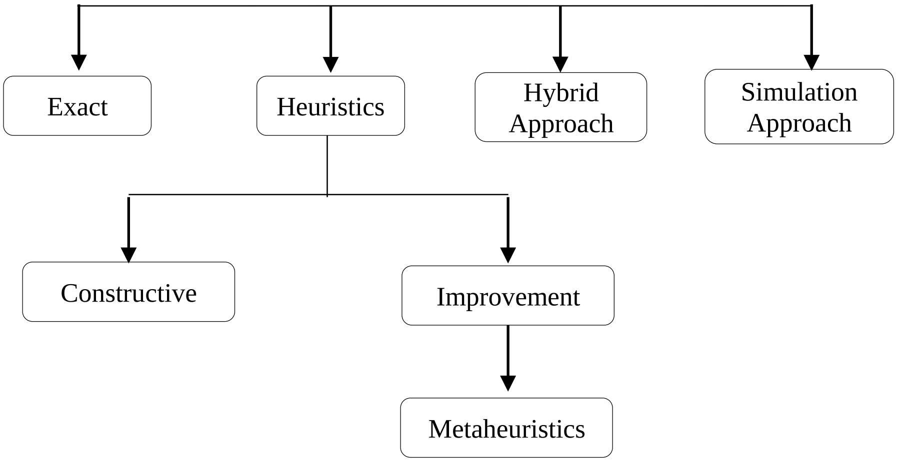
In [Figure 2.5], reviewed literatures are classified according to the different performance measures. Clearly it can be noted that most of the literatures heavily depended on C_{max} criterion with around 57% share. *C/F* criteria add up to 15% among all the reviewed papers. It's quite surprising to see that only 4% of the papers considered *E/T* criteria same time, which is actually very common in real practice. Another vital observation is that only 2% papers dealt with multi-objectives at a time. Multi-objective performance measure approach is a vast field of study as recently shown in [T'kindt & Billaut, 2006]. [Minella, Ruiz, & Ciavotta, 2008] reviewed that the number of existing multi-performance measure approaches is very huge for regular flow shop problems. That's why we think multi-performance measure approach for HFSP is a very necessary and realistic field of research.



[Figure 2.5] Statistics of performance measures used in literature

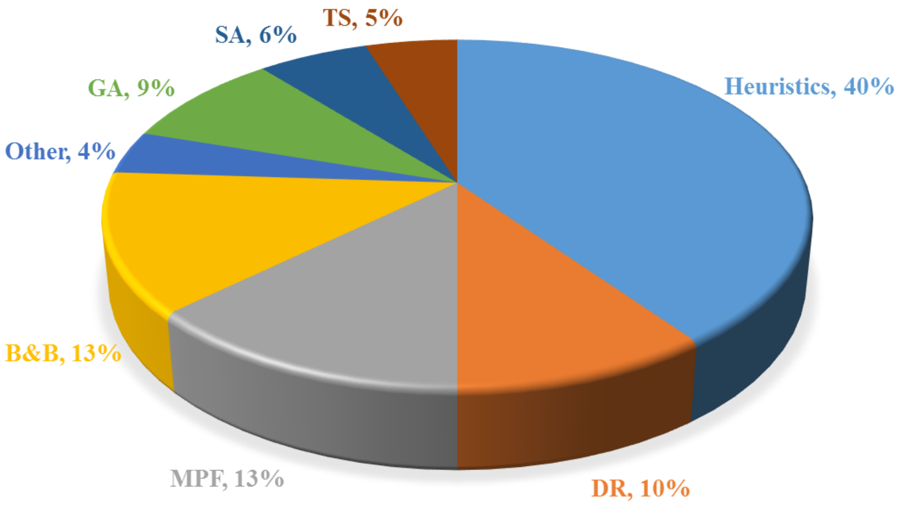
2.3 Solution methods for HFSP

Since the great work of [S. M. Johnson, 1954] published, many solution approaches have been proposed to solve the flow shop problems in so many several types of scheduling characteristics. Based on [Guinet, Solomon, Kedia, & Dussauchoy, 1996], [J. Gupta, Hariri, & Potts, 1997], [Wang, Pan, & Tasgetiren, 2010], [Yenisey & Yagmahan, 2014], [Sioud, Gagné, & Gravel, 2014], reviewed papers are classified into different solutions procedures as shown in [Figure 2.6].



[Figure 2.6] Classification of reviewed papers by solution methods

Optimum or exact methods are those that generate an optimal schedule with respect to a performance criterion by using mathematical models. On the other hand, approximate methods are those that seek to achieve a feasible solution closer to the optimum in a reasonable amount of time. The use of optimum methods is useful when the problem size is small. [Yenisey & Yagmahan, 2014] said that the optimum methods become inefficient for large problems, since they have many jobs, machines and goals. [Moccellin & Santos, 2000] classified heuristic into constructive and improvement heuristics. Hybrid methods are those procedures that combine two or more metaheuristics and uses search strategies [Sioud et al., 2014]. [Boschetti, Maniezzo, Roffilli, & Röhler, 2009] emphasizes the use of hybrid methods that are developed from the interpolation of metaheuristics and mathematical programming. By reviewing those literatures, papers are classified according to the solution approaches used as shown in [Figure 2.7].



[Figure 2.7] Distribution of applied methods in literature[2]

From the [Figure 2.7], it's clearly seen that most of the authors applied heuristics in their researches. 13% papers used MRP, whereas 10% applied simple DR policies in their study. Simulation based approaches is very difficult to find in the literature. Any previous methodologies combined with simulation is a potential field of research.

---

[2] B&B = branch and bound, MPF = mathematical programming and formulation, DR = dispatching rules, TS = tabu search, SA = simulated annealing, GA = genetic algorithms.

2.3.1 Dispatching rules

Dispatching rules are the simplest type of heuristics known as scheduling policies or even list scheduling algorithms. Number of papers dedicated to these policies and comparisons among them. [Brah, 1996] compared ten dispatching policies for m-stage problem with maximum tardiness criterion. [Kadipasaoglu, Xiang, & Khumawala, 1997] showed the comparisons of dispatching policies in both static and dynamic HFS. [Sriskandarajah & Sethi, 1989] proposed a set of sequencing rule based heuristics for two-stage problem. [Al-Turki, Arifusalam, El-Seliaman, & Khan, 2011] presented a problem of resource allocation and scheduling in a flexible job shop with the objective of selecting the best dispatching rule with regard to desired performance measure.

Dispatching rules are particularly suitable to deal with complex, dynamic, and unpredictable environments and hence their popularity have been increasing in practice. In [Paul, 1979], a two-stage glass container HFS was studied and ad-hoc dispatching rules were proposed. In [Adler et al., 1993], [Tsubone, Ohba, Takamuki, & Miyake, 1993] scheduling systems was designed by dispatching rules. [Verma & Dessouky, 1999] studied scheduling with dispatching policies for m-stage problem with uniform parallel machines and identical jobs.

2.3.2 Simulation

Simulation guru [Shannon & Johannes, 1976] defined simulation as the process of designing a model of a real or imaginary system and conducting experiments with the model either for understanding the behavior of the system or evaluating various strategies for the operation of the system. [Kelton, Sadowski, & Sturnock, 2004] identifies simulation as an iterative method that includes several stages. Some literatures used the simulation techniques to analyze the behavior of complex production environments like HFSs. [Brah & Wheeler, 1998] carried simulation studies to analyze further the performance of dispatching rules with makespan and maximum tardiness criteria. Simulation is also used in [Grangeon, Tanguy, & Tchernev, 1999] to check the effectiveness of dispatching rules. [Kadipasaoglu et al., 1997] evaluated different scheduling rules under both static and dynamic criteria using simulation. [Brah, 1996] constructed a HFS simulation model for analyzing the performance of different priority rules with the objectives of mean tardiness and maximum tardiness. [Brah & Wheeler, 1998] investigated various priority dispatching rules for mean flow time and makespan perspective. [Uetake, Tsubone, & Ohba, 1995] analyzed the effect of production run length and sequencing rules on makespan and maximum work in process. [Chtourou, Masmoudi, & Maalej, 2005] developed a simulation based expert system for finding optimal number of machines in regard to due date related performance measures. [Allahverdi & Tatari, 1996] utilized simulation for solving two-machine flow shop scheduling problem where machines are subject to random breakdowns.

Chapter 3. Problem Description

The considered problem in this study is to build a *generic simulation model* for HFSs of *any architectures* as shown in [Figure 2.2]. The simulation model should be *flexible* so that any kind of HFS can be studied over a period of time under certain criteria. The model is to be built for the purpose of studying the *performances* of HFS under *different scheduling (dispatching)* rules. With the help of simulation experiments, HFS related decision makers should be able to take *real time decisions* for scheduling problem. To the best of our knowledge, till now there has been no study attempted to make a generic flexible simulation model considering almost all types of characteristics of HFS. In the production line, the HFSP becomes trickier due to the existence of different uncertainties. Availability of machine setup time and random machine breakdown adds more complexity for decision making of HFSP. Also when the processing or cycle times of the available jobs are not deterministic (i.e. stochastic), it's necessary to find the appropriate probability distribution function for time parameters. In practice, production managers of HFSs have to take decision on a *day to day* operation basis for job scheduling rather than for a long period of time. Due to such possible complexities and stochasticities of HFS, simulation based solution method can be adopted for *real time decision making support system*.

3.1 Notations of parameters and variables

To describe the problem more formally, we first introduce the following notations:

- j = Job index; $j = 1, 2, \ldots, J$
- TL_j = Transfer batch size of job j[3]
- s = Stage index; $s = 1, 2, \ldots, S$
- w = Workstation index; $w = 1, 2, \ldots, W$
- m = Machine index; $m = 1, 2, \ldots, M$
- l = Lap index within individual machine, $l = 1, 2, \ldots\ldots, L$
- W_s = Total number of workstations at stage s
- M_{sw} = Total number of machines at workstation w of stage s
- O_{jsw} = Operation of job j at workstation w in stage s; if the job j need to processed at workstation w of stage s, then $O_{jsw} = 1$; otherwise $O_{jsw} = 0$
- $O_j = \sum_{s=1}^{S} \sum_{w=1}^{W} O_{jsw}$ = Total number of operations of job j
- $ReO_{jsw} = \sum_{s=s}^{S} \sum_{w=w}^{W} O_{jsw}$ = Total remaining number of operations of job j when its ready to be processed at workstation w of stage s
- t = Production time horizon index; $t = 1, 2, \ldots\ldots\ldots, T$ *(minute, hours, days, months etc.)*
- d_{jt} = Production target of job j at t unit time
- $d_{jT} = \sum_{t=1}^{T} d_{jt}$ = Total production target of job j at T unit time
- $d_{JT} = \sum_{j=1}^{J} \sum_{t=1}^{T} d_{jt}$ = Total production target at T unit time

---

[3] If jobs are processed by sublots then each sublot (with a batch size) acts as an individual job and it's also denoted by j

- P_{jml} (O_{jsw}) = Processing time of job j at lap l of machine m for operation O_{jsw}
- CT_{jm} (O_{jsw}) = $\sum_{l=1}^{L} P_{jml}\left(O_{jsw}\right)$ = Cycle time of job j at machine m for operation O_{jsw}
- CT_{jm} (TL_j) = $TL_j \times CT_{jm}$ (O_{jsw}) = Sublot cycle time of job j at machine m for operation O_{jsw}
- CT_j = $\sum_{s=1}^{S} \sum_{w=1}^{W} CT_{jm}\left(O_{jsw}\right)$ = Total cycle time of job j
- CT_j (TL_j) = $TL_j \times CT_j$ = Total batch cycle time of job j
- $ReCT_{jsw}$ = $\sum_{s=s}^{S} \sum_{w=w}^{W} C_{jm}\left(O_{jsw}\right)$ = Total remaining cycle time of job j when its ready to be processed at workstation w of stage s
- r_j = Release time of job j at the beginning of processing
- $ArrT_{jsw}$ = Arrival time of job j at workstation w of stage s
- r_{jsw} = Starting time of processing of job j at workstation w in stage s
- C_{jsw} = Completion time of job j at workstation w in stage s
- C_j = Completion time of job j
- $F_j = C_j - r_j$ = Flow time of job j at workstation w in stage s
- $F_{jsw} = C_{jsw} - ArrT_{jsw}$ = Flow time of job j at workstation w of stage s
- DD_j = Due date of job j
- MOD_j =$Max\left\{\frac{DD_j}{O_j}, C_{jm}\left(O_{jsw}\right)\right\}$ = Modified operation due date of job j
- S_{msw}^{SI} = Sequence independent setup time for machine m at workstation w in stage s, if job j is a predecessor of job j+1 and both of different type; otherwise S_{msw}^{SI} = 0

3.2 Objectives

As mentioned earlier this study is performed to make a flexible simulation for HFSP considering almost all the criteria in Section 2.2.1, so there are several objectives to be accomplished as well. The following objectives are set in this study:

- The main objective is to make *flexible simulation modelling framework* for making real time *decision support system* of HFSP.
- To create a system that enables making comparisons among different commonly used job scheduling (dispatching) policies.
- To compare and find the best combination of dispatching policies based on the following performance measures:
    - ☐ Mean flow time, $F_{avg} = \frac{1}{J}\sum_{j=1}^{J} F_j$
    - ☐ Makespan, $C_{max} = \max_{1 \leq j \leq J} C_j$
    - ☐ Mean tardiness, $T_{avg} = \frac{1}{J}\sum_{j=1}^{J} max\left(0,\ C_j - DD_j\right)$
- To analyze the impact of varying batch size on the following performance measures
    - ☐ Mean machine utilization rate $U_{avg} = \frac{1}{M}\sum_{m=1}^{M} U_m$
    - ☐ Total machine setup time $S = \sum_{s=1}^{S}\sum_{w=1}^{W}\sum_{m=1}^{M} S_{msw}^{SI}$

3.3 Constraints

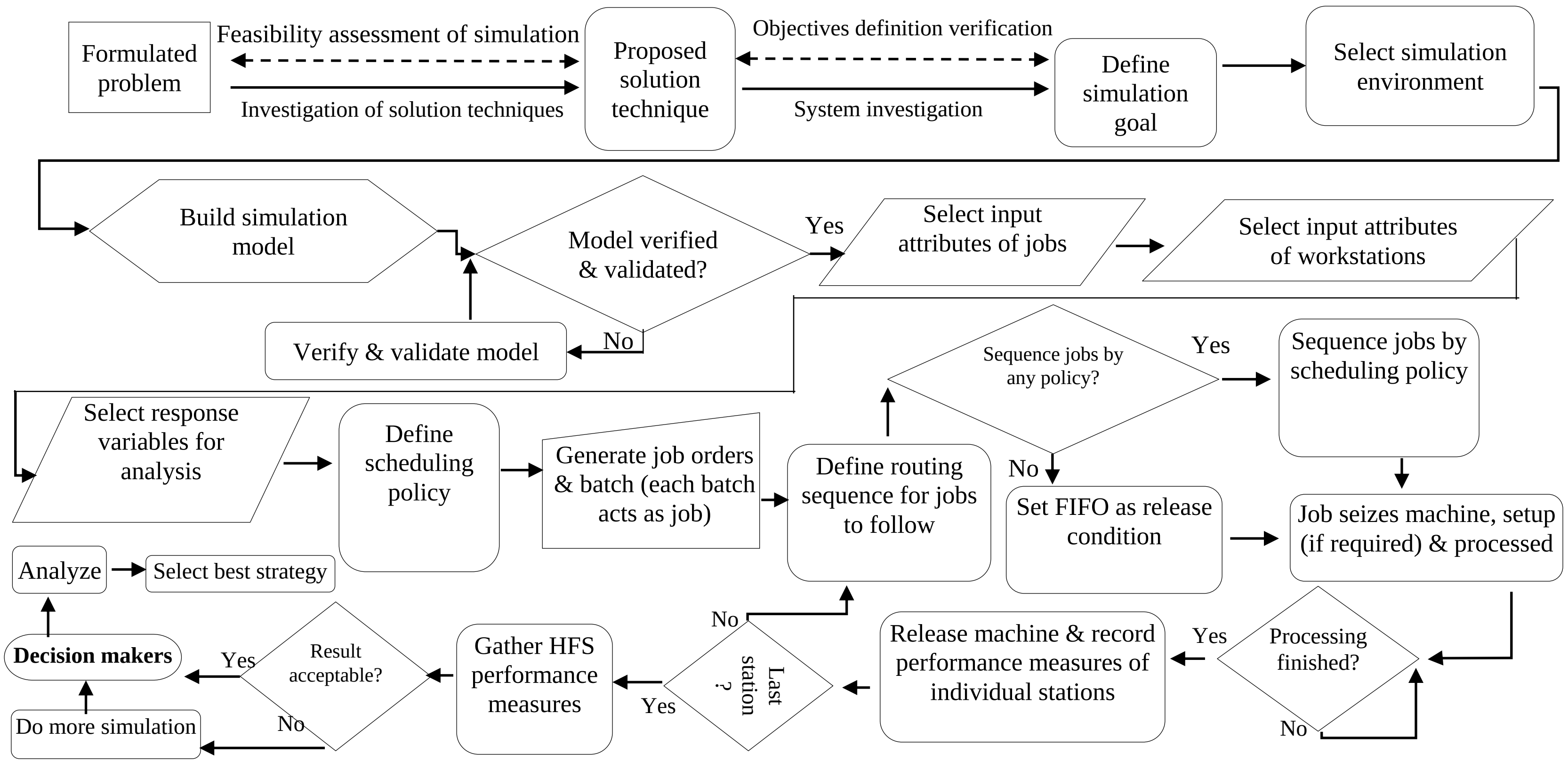
- $C_{max} \geq C_{jSW}$; $\forall j$
  - Makespan should be greater than or equal to the completion time of any job on the last stage
- $C_{jsw} \geq C_{jsw-1} + P_{jsw}$; $\forall j,s,w$
  - Completion time of job j at workstation w of stage s should be greater than or equal to completion time in the preceding workstation (w-1) of stage s
- P_{jml} (O_{jsw}), CT_{jm} (O_{jsw}), r_j , r_{jsw} , C_{jsw} , $DD_j \geq 0$; $\forall j,s,w,m$
  - Restriction that all time units are non-negative

Chapter 4. Methodology

Considering the problem described in Chapter 3 and from the experience of reviewing the existing literatures, this study approaches the simulation based methodology. Simulation is a powerful tool for testing the efficiencies of different scheduling (dispatching) policies without affecting the existing production shop layout or amount of resources. The main purpose of this study is to build a flexible simulation model so that any type of HFS can be mimicked. To the best of our knowledge only simulation can deal such a complex problem.

4.1 Simulation framework

The proposed generic simulation model framework is followed as shown in [Figure 4.1]. The first step starts with identifying the communicated problem. After the feasibility assessment, appropriate solution technique is to be proposed. If the solution methodology proposed is simulation, then goals of the simulation should be elaborated and well described. In the next step, a proper simulation environment should be chosen. After making the simulation model and verification, input parameters for both available jobs and workstations need to be defined. After defining input parameters, proper response variables have to be declared upon which decision makers will be able to perform in depth analysis. The integration of desired job sequencing and dispatching policies would be the next task to accomplish. At the beginning of simulation, orders of jobs should be generated and divided into respective sublots according to pre-defined transfer batch size quantity. The sublots acts as individual jobs throughout whole production layout until completion.



[Figure 4.1] Simulation framework for HFS

Sequence the jobs according to the integrated dispatching policies in each of the workstations buffer queue and process them in the alternative unrelated and dedicated machines. Record the performance measures' values for each of the applied dispatching policies. Analyze and compare the results and select the best combinations of dispatching policies for respective performance measure. By observing the simulation results decision makers can easily decide about the best sequencing policy for the HFS. The implementation of the above mentioned simulation framework is described in the following case study Section.

4.2 Proposed dispatching policies

The proposed simulation methodology is combined with the commonly used job sequencing and priority dispatching policies. In this study, total eleven (11) job sequencing policies are considered for the HFS performance measures. [Table 4.1] presents a brief description and nature of these dispatching policies. *HVF* and *LVF* denotes *high value first* and *low value first* respectively.

[Table 4.1] Job dispatching policies

No.	**Rules**	**Description**	**Type**	**Ranking Criteria**
1	CR	Critical Ratio	Dynamic	HVF
2	MST	Minimum Slack Time	Dynamic	LVF
3	LCT	Longest Cycle Time	Static	HVF
4	SCT	Shortest Cycle Time	Static	LVF
5	FIFO	First In First Out	Static	……
6	LIFO	Last In First Out	Static	……

No.	Rules	Description	Type	Ranking Criteria
7	S/OPN	Slack per Operation	Dynamic	LVF
8	SRCT	Shortest Remaining Cycle Time	Dynamic	LVF
9	LRCT	Longest Remaining Cycle Time	Dynamic	HVF
10	MMOD	Minimum Modified Operation Due Date	Dynamic	LVF

4.2.1 Mathematical measures of dispatching policies

Once the appropriate dispatching policies are chosen, mathematical formula is also need to be defined for those dispatching policies. All the mathematical measures for considered dispatching rules are given below, except FIFO and LIFO, since they are self-evident.

[1] CR = HVF $\left[\frac{\text{Total remaining cycle time}}{\text{Modified due date - Current time}}\right]$ = HVF $\left[\frac{ReCT_{jsw}}{MDD_j - TNOW}\right]$

[2] MST = LVF [Modified due date - Current time - Total remaining cycle time]

$$= \text{LVF} \left[MDD_j - TNOW - ReCT_{jsw}\right]$$

[3] LCT = HVF $\left[\underset{1<j<J}{\overset{Max}{}}\{CT_{jm}(O_{jsw})\}\right]$

[4] SCT = LVF $\left[\underset{1<j<J}{\overset{Min}{}}\{CT_{jm}(O_{jsw})\}\right]$

$$[5]\ \ \mathrm{S/OPN} = \mathrm{LVF}\left[\frac{\text{Modified due date-Current time-Total remaining cycle time}}{\text{Total remaining number of operations}}\right]$$

$$= \mathrm{LVF}\left[\frac{MDD_j - TNOW - ReCT_{jsw}}{ReO_{jsw}}\right]$$

$$[6]\ \ \mathrm{SRCT} = \mathrm{LVF}\left[\underset{1<j<J}{Max}\left\{ReCT_{jsw}\right\}\right]$$

$$[7]\ \ \mathrm{LRCT} = \mathrm{HVF}\left[\underset{1<j<J}{Min}\left\{ReCT_{jsw}\right\}\right]$$

$$[8]\ \ \mathrm{MMOD} = \mathrm{LVF}\left[max\left\{\frac{DD_j}{O_j}, C_{jm}\left(O_{jsw}\right)\right\}\right]$$

At the completion of any operation, a machine becomes free, and the dispatching rule specifies what the machine should do next. One of the options, of course, is to keep the machine idle until all jobs for the current workstation are available and be sequenced by dispatching rules but in the spirit of non-delay schedule, most dispatching rules immediately assign work to the machine as long as work is available and sequencing the rest in the queue.

4.3 Proposed heuristic

As described in Section 3.2, the main objective of this study is to create a system that enables making comparisons between different combinations of considered dispatching policies based on the performance measured mentioned earlier.
Test run is performed based on the parameters described in Section 5.3.5. The simulation is performed by changing the combinations of dispatching rules each time and the related performance output is recorded. Total 5 workstations and

10 dispatching rules are considered in this study. So the major question arises exactly how many combinations are possible to compare the considered performance measures of the HFS.

According to the literatures [Namakshenas & Sahraeian, 2013], [Kadipasaoglu et al., 1997], [Andres, Gomez, & Garcia-Sabater, 2006], if a HFS has n dispatching rules and w number of workstations then:

- n^w number of combinations possible (if repetition allowed) or
- n_{P_w} number of combinations possible (if repetition not allowed)

So if the above mentioned combination policy is followed then we may have to conduct:

- $10^5 = 100{,}000$ number of combinations (if repetition allowed) or
- $10_{P_5} = 30{,}240$ number of combinations (if repetition not allowed)

These huge number of combinatorial simulation experiment might be quite time consuming and hard to achieve. As a result, a near optimal strategy should be addressed to minimize the number of combinations to compare the performance measures and find the near optimal solutions.

In this study, an *individual workstation mean flow time (IWMF)* based heuristic strategy is proposed to find a near optimal combinations of dispatching rules in regard to the performance measures mentioned earlier.

The proposed heuristics is described in the following section.

4.3.1 IWMF heuristic

<ins>*Step 1:*</ins>

1) Iterate the simulation run applying all the dispatching rules at Grinding's [*w(1)*] buffer queue (except FIFO and LIFO) and FIFO at rest of the workstations [*w (2,3,4,5)*].

2) For $Min[F_{avg}(G)]$: set choice, *c(1)*: the dispatching policy performs better (if ties occurs choose one of tied policies arbitrarily)

3) For $Min[F_{avg}]$: set choice *c(2)* combination of dispatching policies performs better

4) For $Min[T_{avg}]$: set choice *c(3)* combination of dispatching policies performs better

5) For $Min[C_{max}]$: set choice *c(4)* combination of dispatching policies performs better

Step 2:

1) Keep choice *c(1)* at *w(1)*, iterate simulation run applying all dispatching rules at *w(2)* and 'FIFO' in w(3,4,5)

2) For $Min[F_{avg}(US)]$: set choice, *c(1)*: combinations of the dispatching policies performs better for *w(1)+w(2)* (if ties occurs choose one of tied policies arbitrarily)

3) Assign temporary choices *c΄(2), c΄(3), c΄(4)* for the composite dispatching rules perform better for step 2 $Min[F_{avg}]$, $Min[T_{avg}]$, $Min[C_{max}]$ respectively

4) Compare step 2 $Min[F_{avg}]$ with step 1 $Min[F_{avg}]$:

   ☐ If step 2 $Min[F_{avg}]$ < step 1 $Min[F_{avg}]$ then,

   Set choice *c(2)*: the combination of dispatching policies perform better for step 2 $Min[F_{avg}]$; otherwise ignore

5) Compare step 2 $Min[T_{avg}]$ with step 1 $Min[T_{avg}]$

   ☐ If step 2 $Min[T_{avg}]$ < step 1 $Min[T_{avg}]$ then,

   Set choice *c(3)*: the combination of dispatching policies perform better for step 2 $Min[T_{avg}]$; otherwise ignore

6) Compare step 2 $Min[C_{max}]$ with step 1 $Min[C_{max}]$:

   - ☐ If step 2 $Min[C_{max}]$ < step 1 $Min[C_{max}]$ then,

     Set choice *c(4)*: the combination of dispatching policies perform better for step 2 $Min[C_{max}]$; otherwise ignore

7) Follow this procedure until last workstation *w(5)* and find the appropriate choices [*c(1), c(2), c(3), c(4)*]

Step 3:

1) Iterate simulation, applying same dispatching rule in all the workstations (w) by using all the considered dispatching rules
2) Find $Min[F_{avg}]$, $Min[T_{avg}]$, $Min[C_{max}]$
3) Compare step 3 $Min[F_{avg}]$ with *c(2)* $Min[F_{avg}]$:

   - ☐ If step 3 $Min[F_{avg}]$ < *c(2)* $Min[F_{avg}]$ then,

     Set choice *c(2)*: the combination of dispatching policies contribute for step 3 $Min[F_{avg}]$; otherwise ignore

4) Compare step 3$Min[T_{avg}]$ with *c(3)* $Min[T_{avg}]$:

   - ☐ If step 3 $Min[T_{avg}]$ < *c(3)* $Min[T_{avg}]$ then,

     Set choice *c(3)*: the combination of dispatching policies contribute for step 3 $Min[T_{avg}]$; otherwise ignore

5) Compare step 3 $Min[C_{max}]$ with *c(4)* $Min[C_{max}]$:

   - ☐ If step 3 $Min[C_{max}]$ < *c(4)* $Min[C_{max}]$ then,

     Set choice *c(4)*: the combination of dispatching policies contribute for step 3 $Min[C_{max}]$; otherwise ignore

Chapter 5. Case Study

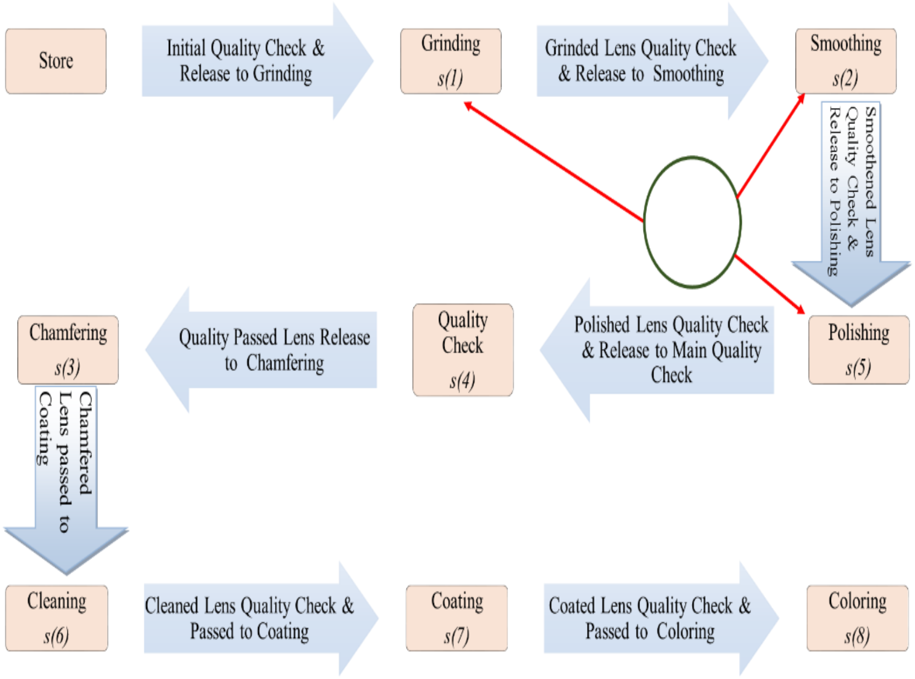
5.1 Introduction

For the above mentioned problem, this thesis considered the case study of *optical lens processing* industry as mentioned earlier in Chapter 1. The raw material of optical lenses is nothing but a piece of *roughly* surfaced glass material. Firstly, the rough surface goes through a sequential processing steps to eradicate surface roughness. After making the surface accurately smoothened and few more steps, they become ready to be assembled into the final products as shown in [Figure 5.1].

[Figure 5.1] Raw optical lens and assembled final products

5.2 Optical lens processing system

Typically lens processing layout comprises of several stages in serial as shown in [Figure 5.1]; from which its clearly seen that it consists of total eight (8) stages in a series starting from the raw material store to the end of coloring Section. It's very important to mention that not all the stages hold the characteristics of a HFS.



[Figure 5.2] Optical lens processing flow chart

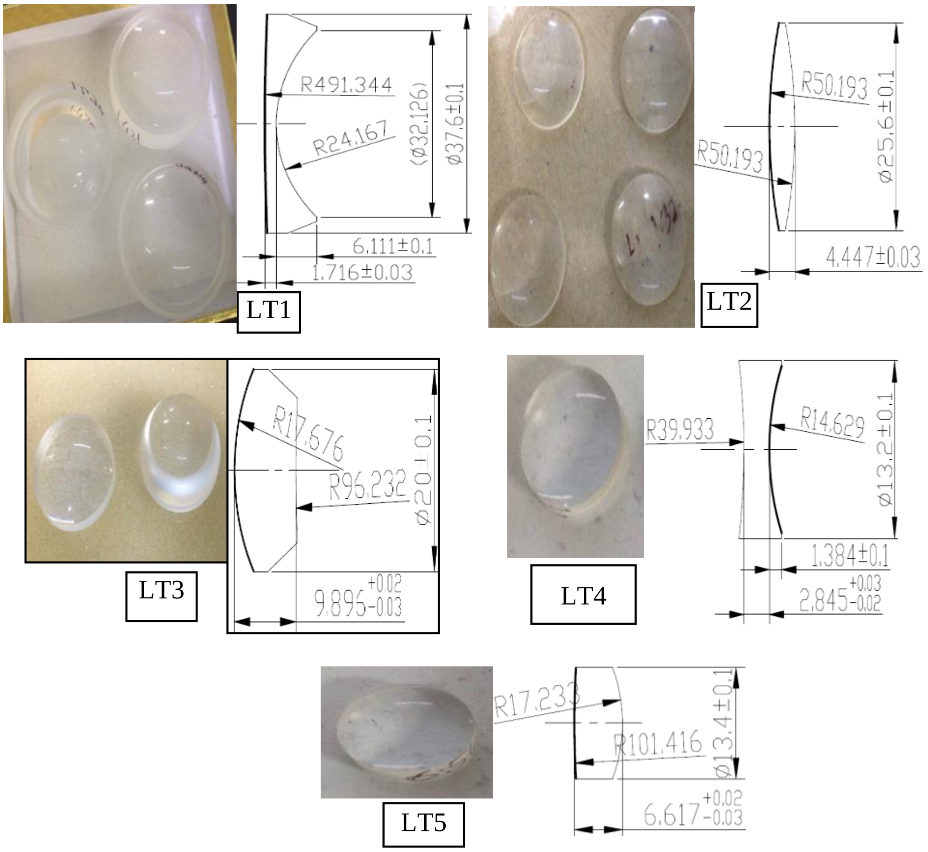
In the [Figure 5.2], the 'focused area' is indicated by three (3) arrow directions, which are the stages named as *grinding*, *smoothing* and *polishing*. These three stages are similar to the characteristics of a HFS. Other stages cannot be compared to HFS structure. As a result, this research is focused on those three sequential stages.

5.3 Design of Experiment

To make a simulation model based on the framework mentioned in Section 4 and to run a sample experiment, a detailed design of experiment is explained as follows:

5.3.1 Analyzed job types

There are several types lenses are being processed in the company but major *five (5)* types of lenses are considered for the experiment. The dimensions of those lenses are shown in [Figure 5.3].



[Figure 5.3] Lens types

Lens types are denoted as follows:

- *LT(j)*= Lens type(*j*)*;* where,
  - *LT(j=1, 2, 3, 4, 5)* means lens type *LT1, LT2, LT3, LT4, LT5* respectively

The lenses are processed by consistent sublots according to a predefined batch sizes. The lenses are carried out through the entire HFS by means of portable trays of different dimensions based on the respective sublot and get processed. The sublot size of each lens type is given in [Table 5.1] and a sample view of a sublot in a tray is shown in [Figure 5.4].

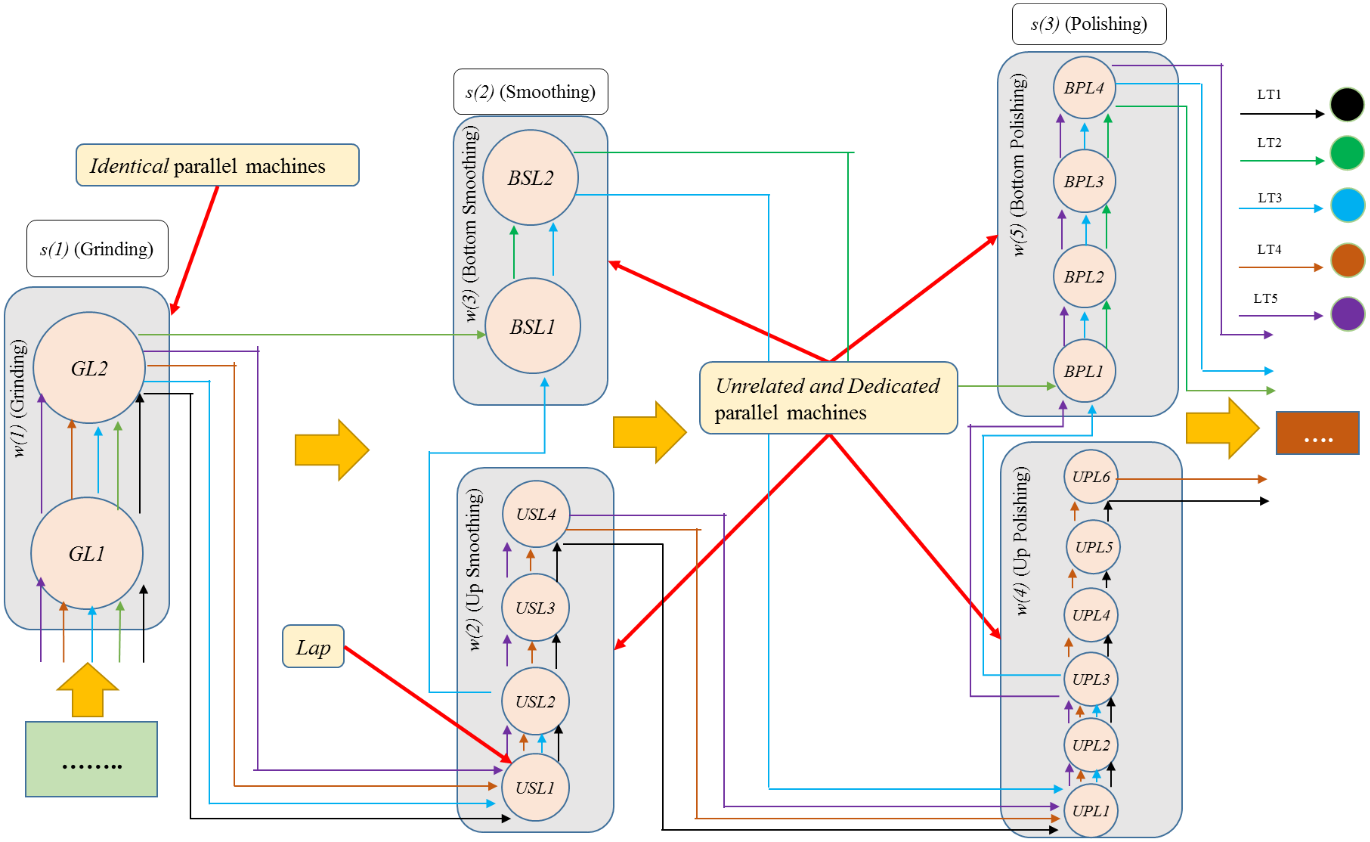
[Table 5.1] Sublot size of lenses

LT(j)	TL_j
LT1	30
LT2	48
LT3	120
LT4	208
LT5	208

[Figure 5.4] Sample view of sublot for LT1

5.3.2 Layout of the analyzed HFS

The layout of the considered optical lens processing hybrid flow shop is shown in [Figure 5.5].



[Figure 5.5] Layout of optical lens processing HFS

The five types of lenses and their processing sequences or routings are indicated by utilizing different *colors* for readers' understanding. For the experiment, total 3 stages are considered as mentioned in Section 5.2, where,

- s= 1, 2, 3 means *Grinding, Smoothing, Polishing* stage respectively

Since at the G*rinding(G)* stage only identical parallel machines are available, so this is referred to as $w = 1$. But in *Smoothing (S)* and *Polishing (P)* stage, the parallel machines are available in the form of *unrelated* and *dedicated* machine set characteristics. Due to the availability of both dedicated parallel machine environment, we divided both *smoothing* and *polishing* stage into 2 workstations for each one of them as shown in [Figure 5.5] where,

- w= 2, 3, 4, 5 means *Up Smoothing (US), Bottom Smoothing (BS), UP Polishing (UP), Bottom Polishing (BP)* workstation respectively

The machines are denoted depending on which workstation they belong to. They are denoted as '*w MC'*; where,

- (w=1, 2, 3, 4, 5) *MC* means *Grinding machine (GMC), Up Smoothing machine (US MC), Bottom Smoothing machine (BS MC), Up Polishing machine (UP MC), Bottom Polishing machine (BP MC)*

Appendix A gives a real time view of all types of machines available in the above mentioned workstations. Another important issue to mention is the existence of sequential sub-machines within the machines as shown in [Figure 5.5], which are typically called *lap* in optical lens processing industries. Each individual workstation's machines have different number of laps inside it and lenses pass through these laps sequentially while being processed. It is noteworthy to mention that lenses do not always necessarily to pass through all available laps of a particular machine (e.g. *LT3*, *LT5*; see [Figure 5.5]). these laps can be denoted as '*(w)L(l)*', means lap l of the machine located at workstation w.

For example:

- (w=2)*L3 = USL3* means *lap 3* of *Up Smoothing machine* (*US MC*)

Similarly, all other laps are denoted as shown in [Figure 5.5]. APPENDIX A shows a real time view of *laps* in the machines.

5.3.3 Simulation model setup

There are several methods to create the simulation model on computer. Different kinds of programming languages like FORTRAN, C/C++, Python, Java, BASIC etc. had been used in several literatures [Altiok & Melamed, 2010] or one of the several commercially available tools can be utilized. Overall simulation tools can be classified into 3 basic classes such as (1) *general purpose simulation languages*, (2) *simulation front-ends and* (3) *simulation packages*. General purpose languages require the user to be proficient in programming along with being a competent simulationist. Simulation front-ends are the essential interface programs between user and simulation language being used. The most advanced of all is the simulation packages available today, integrating similar terminologies common in the manufacturing industries. Moreover, simulation packages offer graphical representations and animation as well.

Discrete event simulation with ARENA$^{®}$ *v14.70 (academic)* simulation package is used in this study for model development and experiment. ARENA$^{®}$ is a powerful flexible simulation tool to mimic a HFS that can accurately represent the system virtually. ARENA$^{®}$ employs an object-oriented design for entirely graphical model development. ARENA$^{®}$ has a natural and consistent modeling methodology due to its flowchart style model building regardless of detail or complexity. ARENA$^{®}$ is built on SIMAN language. While creating simulation model graphically, ARENA® automatically generates the underlying SIMAN codes to perform simulation runs [Takus & Profozich, 1997]. Another advantage of ARENA® is availability of interaction with many applications such as *Microsoft Access, Excel* with its built-in spreadsheet data interface. Furthermore, with the support of integrated *Visual Basic for Applications (VBA®)*, there is actually no limit on creating interfaces and programs. Because of its such a huge impact, ARENA® is now being taught in many *Industrial Engineering* schools worldwide. The *Discrete event simulation* model used in this system is customized and integrated with *VBA®*. *Model based time* units are set as *minutes*. Model warm-up period is set for 1 minute. Number of replications is 1, if stochastic features are used otherwise 1 when deterministic criteria are considered. Replication terminates when processed batch counts reach the production target of *t* unit time horizon.

5.3.4 Assumptions for experiment

The flexible simulation model is actually robust for any type HFS, but for the sample experiment certain HFS characteristics are considered among the criteria mentioned in [Figure 2.2]. The assumptions for the sample experiment are explained as follows:

[1] All types of lenses are ready to be released simultaneously at the first workstation.

[2] All the machines in each of the workstations are available at the beginning.

[3] Each machine can process one type lens at a time.

[4] Job pre-emption is not allowed; once a machine start processing a sublot of a particular lens type, it must finish that sublot before taking another sublot of any lens type.

[5] Although probabilistic parameters are calculated for each of the stochastic criteria but only deterministic measures (i.e., mean values) are used for the experiment and result analysis.

[6] Sequence independent set-up time is considered and integrated separately along with processing time in the model.

[7] Identical parallel machines at (s=1) and unrelated, dedicated parallel machines at (s=2 & 3) are considered (as found in optical lens processing industry) .

[8] Consistent sublots are considered throughout the whole HFS system.

5.3.5 Attributes for simulation experiment

This part is dedicated to attributes and parameters required for the simulation run experiment. To properly implement and design the simulation model, values of related attributes need to be set. An attribute is a characteristic of the entities with values assigned. Entities differ from each other with different values assigned. These assigned values of attributes provide the basis to calculate statistics and also offer programming flexibilities for the modeler. The assigned values of attributes are subject to change depending on the scenario of the simulation run, brings the flexibility for the experiment, which is the core objective of this thesis.

The lens company usually runs for *2 shifts* a day, *22 days* a month. The sample simulation experiment was performed for *1day 1 shift* production target quantity of all types of lenses. That’s why, first we set the production target of all the 5 types of lenses for a *1 day 1 shift* based on the master production schedule from company’s previous data as shown in [Table 5.2].

[Table 5.2] Production target of lenses

LT(j)	**d_{jT} (t=22 days) [2 shifts/day] [480 mins/shift]**	**d_{jt} (t=1 day) [2 shifts/day] [480 mins/shift]**	**d_{jt} (t=1 shift)**	**TL_j**	**d_{jt} [sublot] (t=1 shift or 480 mins)**
LT1	50000	2273	1136	30	38
LT2	50000	2273	1136	48	24
LT3	50000	2273	1136	120	10
LT4	50000	2273	1136	208	6
LT5	50000	2273	1136	208	6
d_{JT}	**250000**	**11364**	**5682**		**84**

For the pilot experiment we considered different number of machines in each workstation. Capacity requirement planning was not studied in this study since there is a specific machine space constraint. The number of machines considered reasonably so that we can obtain all types of performance measures mentioned in objectives [see Section 3.2]. Values of this attribute is shown in [Table 5.3].

[Table 5.3] Number of machines

s	***w***	***w MC***	***M_{sw}***
G	G	G MC	3
S	US	US MC	2
	BS	BS MC	2
P	UP	UP MC	14
	BP	BP MC	5

Since in practice usually a single machine operator is assigned to each of the machines available in each workstation, so the number of workers in each station is assumed to be exactly same according to the number of machines in that workstation.

Each type of lens has two (2) surfaces, belong to the radiuses of 2 different spheres. These surfaces are denoted by *R1* and *R2* as shown in [Figure 5.3]. The processing time of the lenses is found to be followed by normal distribution from the *SPSS* experiment using the previous sample data of production report. The estimated normal distribution parameters *mean (μ)* and *standard deviation (*σ) of processing times P_{jml} (O_{jsw}) for each lap in the machines are presented in [Table 5.4].

Usually in manufacturing or production industries, there can be two types of machine breakdowns such as (1) *count-based and* (2) *time-based*. Sometimes machine operator needs to change the *machine tools* due to *tool tear down* issue after a certain quantity of jobs are processed. Such breakdowns can be identified as count-based breakdowns. On the contrary, machine malfunctioning or other reasons of machine failures can be noted as time-based breakdowns. In optical lens processing company, count-based breakdowns are most frequently to happen, eventually this kind of breakdown is considered in the experiment. The data from the previous production report is tested for fitting the proper distribution of this breakdown and uniform distribution is considered to be followed. The uniform distribution parameters *(a)* and *(b)* for count-based machines breakdown are presented in [Table 5.5]. *a* and *b* define range of probable machine breakdown and the units of *a* and *b* is number of pieces of lenses. Wherever there is a breakdown, there must be a time taken to *repair* the breakdown. In literature, this is called as *mean time to repair (MTTR)*. *Log-normal* distribution is considered for MTTR. The parameters *log-mean (μ)* and *long-standard deviation (s*) is calculated to be 7.98 minutes and 2.34 minutes respectively. For *sequence independent setup time*, this study considers normal distribution and the parameters *mean (μ)* and *standard deviation (*□) are calculated as 10 minutes and 5 minutes respectively.

[Table 5.4] Normal distribution parameters for lenses' lap processing time of machine [P_{jml} (O_{jsw}) unit= seconds]

LT(j)	***Surface***	***G MC***				***US MC***								***BS MC***				***UP MC***												***BP MC***							
		GL1		***GL2***		***USL1***		***USL2***		***USL3***		***USL4***		***BSL1***		***BSL2***		***UPL1***		***UPL2***		***UPL3***		***UPL4***		***UPL5***		***UPL6***		***BPL1***		***BPL2***		***BPL3***		***BPL4***	
		μ	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***	***μ***	***σ***
LT1	R1	30	5			15	5	25	5									250	20	250	20	250	20														
	R2			30	5					15	5	25	5											200	20	200	20	200	20								
LT2	R1	20	5											15	5	25	5													120	20	120	20				
	R2			20	5									15	5	25	5																	120	20	120	20
LT3	R1	15	5			10	5	20	5									200	10	200	10	200	10														
	R2			15	5									10	5	20	5													100	30	100	30	100	30	100	30
LT4	R1	15	5			10	5	20	5									150	20	150	20	150	20														
	R2			15	5					10	5	20	5											150	20	150	20	150	20								
LT5	R1	20	5			10	5	20	5									150	20	150	20	150	20														
	R2			20	5					10	5	20	5																	60	20	60	20	60	20	60	20

[Table 5.5] Uniform distribution parameters for machine breakdown

s	***w***	***w MC***	***a***	***b***
G	G	G MC	2000	4000
S	US	US MC	1800	3000
	BS	BS MC	1200	2000
P	UP	UP MC	1500	2800
	BP	BP MC	1700	2500

5.4 Model development

As mentioned earlier, in this study, ARENA® *v14.70* tool is used to develop the simulation model according to the framework described in Section 4.1. The capabilities of this software, as described in Section 5.3.3, are effectively utilized. The detailed flow chart to make the model is given in APPENDIX B, [Figure B.1]. The whole simulation model is divided into several submodels as described in the following Sections

5.4.1 Job orders creation and routing

Job creation model is responsible to create job orders (i.e. all lens types) to the main model. The *routing* module then direct the jobs to the workstations according to the routing sequence defines in the *sequence* module. A sample job creation submodel for Grinding and a sample routing by sequence for LT1 is shown in APPENDIX B, [Figure B.2]

5.4.2 Workstation design

A sample workstation submodel (Grinding) built in the model is shown in Appendix B, [Figure B.4]. All other workstations ($w=2,3,4,5$) follow the sample design principle.

5.4.3 Queue modules

As described in Section 5.3.2, there are total 5 workstations within 3 stages in the HFS. In this simulation model, queue element is defined for each one of the workstations. These queue elements referred to the buffer of the workstations, upon which the dispatching rules is applied later. The generated queue elements are defined within a queue *set*. The implementation is shown in APPENDIX B, [Figure B.3].

5.4.4 Interface control objects

Five interface control boxes are added in the model as a means of five combo box control objects. The list of these combo boxes comprises the chosen dispatching policies. This combo boxes helps the user to easily interact with the simulation model by choosing the desired dispatching rules and see the impact on the HFS performances. The sample integration of dispatching rules with these combo boxes are shown in Appendix B, [Figure B.5].

5.4.5 Integration of dispatching rules

The job dispatching rules are integrated with the simulation model with the help of *visual basic for applications* (*VBA*). These rules are applied to each of the workstations buffer, where jobs (lens) of different types come and get sequenced by dispatching rules and seized the machine. The sample *VBA code* for the integration of all dispatching policies in regard to *Grinding* workstation's buffer is shown in [APPENDIX C].

Chapter 6. Results and Analysis

6.1 Experiment criteria

The simulation is iterated by changing the combinations of dispatching policies for the available workstations (w). This thesis valued *real time decision support system.* The main focus is to make a *day-to-day* decision support system regarding appropriate combinations of dispatching policies (*composite dispatching rules*) for currently available workstations. For such real time *day-to-day* decision making purpose, data with only fixed or deterministic values are enough to input into the simulation model and thus to decide the appropriate scheduling policies for the *particular day of operation*. Some of the attributes [see Section 5.3.5] varies from day to day, so approximate *probability distributions* are formulated for those attributes [e.g. processing time, P_{jml} (O_{jsw})]. Although these random factors are *integrated* into the simulation model but in the spirit of real time decision making, *deterministic scenarios* are considered by using *mean values* of those *attributes.* For *day-to-day decision making,* analysis with random variables and averaging the output, is not essential. As a result, *one simulation replication* is considered to be enough for *each combination of dispatching rule*.

6.2 Experiment procedure

According to the heuristics procedure (IWMF) described in Section 4.3, simulation model is iterated several times. The iteration results are shown in [Table 6.1]. Total 56 combinations of dispatching rules are performed to compare and find the better or near optimal composite dispatching strategy. For each combination of dispatching rules, individual *workstation's mean flow time*

[$F_{avg}(w)$], and HFS's *mean flow time* [F_{avg}], *mean tardiness* [T_{avg}], *makespan* [C_{max}] values are taken from the simulation outcome to analyze.

The *step 1* comparisons according to the IWMF heuristic is shown in [Figure 6.1]. From the comparisons four choices [*c(1), c(2), c(3), c(4)*] are set as described in Section 4.3 in regard to the performance measures mentioned in Section 3.2.

Similarly, IWMF heuristic *step 2* is explained in both [Figure 6.2] and [Figure 6.3] sequentially.

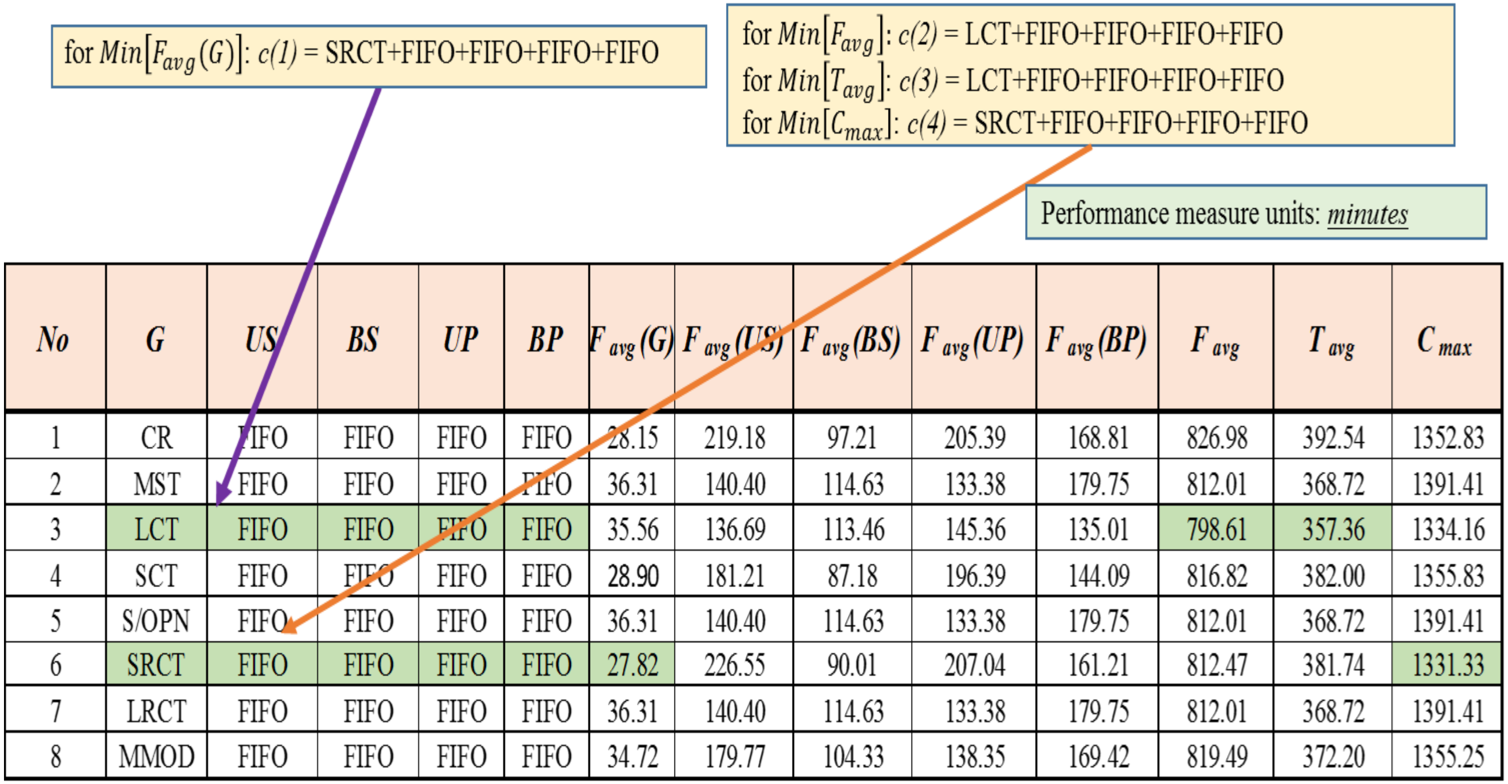
Having the desired choices after *step 2*, IWMF heuristic *step 3* is performed as shown in the [Figure 6.4].

[Table 6.1] Composite dispatching rules and respective performance values

No	***G***	***US***	***BS***	***UP***	***BP***	F_{avg} ***(G)***	F_{avg} ***(US)***	F_{avg} ***(BS)***	F_{avg} ***(UP)***	F_{avg} ***(BP)***	F_{avg}	T_{avg}	C_{max}
1	CR	FIFO	FIFO	FIFO	FIFO	28.15	219.18	97.21	205.39	168.81	826.98	392.54	1352.83
2	MST	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
3	LCT	FIFO	FIFO	FIFO	FIFO	35.56	136.69	113.46	145.36	135.01	798.61	357.36	1334.16
4	SCT	FIFO	FIFO	FIFO	FIFO	28.90	181.21	87.18	196.39	144.09	816.82	382.00	1355.83
5	S/OPN	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
6	SRCT	FIFO	FIFO	FIFO	FIFO	27.82	226.55	90.01	207.04	161.21	812.47	381.74	1331.33
7	LRCT	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
8	MMOD	FIFO	FIFO	FIFO	FIFO	34.72	179.77	104.33	138.35	169.42	819.49	372.20	1355.25
9	SRCT	CR	FIFO	FIFO	FIFO	27.82	102.21	98.11	188.57	73.17	672.47	246.93	1232.16
10	SRCT	MST	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
11	SRCT	LCT	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.68	126.59	744.57	311.80	1275.00
12	SRCT	SCT	FIFO	FIFO	FIFO	27.82	102.21	98.11	189.66	102.17	686.19	258.30	1201.33
13	SRCT	FIFO	FIFO	FIFO	FIFO	27.82	226.55	90.01	207.04	161.21	812.47	381.74	1331.33
14	SRCT	LIFO	FIFO	FIFO	FIFO	27.82	140.97	91.82	183.50	87.08	699.91	271.21	1348.50
15	SRCT	S/OPN	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
16	SRCT	SRCT	FIFO	FIFO	FIFO	27.82	102.21	98.11	190.36	187.12	726.93	298.27	1306.50
17	SRCT	LRCT	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
18	SRCT	MMOD	FIFO	FIFO	FIFO	27.82	113.26	111.31	168.29	101.11	683.95	256.80	1178.50

No	**G**	**US**	**BS**	**UP**	**BP**	F_{avg} **(G)**	F_{avg} **(US)**	F_{avg} **(BS)**	F_{avg} **(UP)**	F_{avg} **(BP)**	F_{avg}	T_{avg}	C_{max}
19	SRCT	CR	CR	FIFO	FIFO	27.82	102.21	97.57	204.32	65.32	678.72	253.18	1223.50
20	SRCT	CR	MST	FIFO	FIFO	27.82	102.21	107.12	205.04	87.16	693.37	267.83	1238.66
21	SRCT	CR	LCT	FIFO	FIFO	27.82	102.21	84.40	174.99	97.11	668.61	243.08	1249.83
22	SRCT	CR	SCT	FIFO	FIFO	27.82	102.21	107.12	205.04	87.16	693.37	267.83	1238.66
23	SRCT	CR	FIFO	FIFO	FIFO	27.82	102.21	98.11	188.57	73.17	672.47	246.93	1232.16
24	SRCT	CR	LIFO	FIFO	FIFO	27.82	102.21	93.82	178.94	84.41	667.90	242.37	1227.83
25	SRCT	CR	S/OPN	FIFO	FIFO	27.82	102.21	107.12	205.04	87.16	693.37	267.83	1238.66
26	SRCT	CR	SRCT	FIFO	FIFO	27.82	102.21	84.40	174.99	97.11	668.61	243.08	1249.83
27	SRCT	CR	LRCT	FIFO	FIFO	27.82	102.21	107.12	205.04	87.16	693.37	267.83	1238.66
28	SRCT	CR	MMOD	FIFO	FIFO	27.82	102.21	107.12	205.04	87.16	693.37	267.83	1238.66
29	SRCT	CR	SRCT	CR	FIFO	27.82	102.21	84.40	186.63	96.88	676.45	250.91	1220.00
30	SRCT	CR	SRCT	MST	FIFO	27.82	102.21	84.40	194.18	96.50	681.81	256.27	1243.33
31	SRCT	CR	SRCT	LCT	FIFO	27.82	102.21	84.40	194.18	96.50	681.81	256.27	1243.33
32	SRCT	CR	SRCT	SCT	FIFO	27.82	102.21	84.40	173.20	95.89	666.61	241.07	1205.33
33	SRCT	CR	SRCT	FIFO	FIFO	27.82	102.21	84.40	174.99	97.11	668.61	243.08	1249.83
34	SRCT	CR	SRCT	LIFO	FIFO	27.82	102.21	84.40	192.66	94.85	679.84	254.30	1233.00
35	SRCT	CR	SRCT	S/OPN	FIFO	27.82	102.21	84.40	194.18	96.50	681.81	256.27	1243.33
36	SRCT	CR	SRCT	SRCT	FIFO	27.82	102.21	84.40	173.20	95.89	666.61	241.07	1205.33
37	SRCT	CR	SRCT	LRCT	FIFO	27.82	102.21	84.40	194.18	96.50	681.81	256.27	1243.33
38	SRCT	CR	SRCT	MMOD	FIFO	27.82	102.21	84.40	173.20	95.89	666.61	241.07	1205.33

No	**G**	**US**	**BS**	**UP**	**BP**	F_{avg} **(G)**	F_{avg} **(US)**	F_{avg} **(BS)**	F_{avg} **(UP)**	F_{avg} **(BP)**	F_{avg}	T_{avg}	C_{max}
39	SRCT	CR	SRCT	SCT	CR	27.82	102.21	84.40	173.20	97.85	667.54	242.00	1205.33
40	SRCT	CR	SRCT	SCT	MST	27.82	102.21	84.40	173.20	97.85	667.54	242.00	1205.33
41	SRCT	CR	SRCT	SCT	LCT	27.82	102.21	84.40	173.20	97.85	667.54	242.00	1205.33
42	SRCT	CR	SRCT	SCT	SCT	27.82	102.21	84.40	173.20	91.93	664.72	239.18	1200.33
43	SRCT	CR	SRCT	SCT	FIFO	27.82	102.21	84.40	173.20	95.89	666.61	241.07	1205.33
44	SRCT	CR	SRCT	SCT	LIFO	27.82	102.21	84.40	173.20	97.85	667.54	242.00	1205.33
45	SRCT	CR	SRCT	SCT	S/OPN	27.82	102.21	84.40	173.20	97.85	667.54	242.00	1205.33
46	SRCT	CR	SRCT	SCT	SRCT	27.82	102.21	84.40	173.20	91.93	664.72	239.18	1200.33
47	SRCT	CR	SRCT	SCT	LRCT	27.82	102.21	84.40	173.20	97.85	667.54	242.00	1205.33
48	SRCT	CR	SRCT	SCT	MMOD	27.82	102.21	84.40	173.20	91.93	664.72	234.52	1200.33
49	CR	CR	CR	CR	CR	28.15	106.25	95.31	151.74	73.87	660.17	239.18	1189.33
50	MST	MST	MST	MST	MST	36.31	138.93	122.95	179.43	197.42	854.40	410.52	1354.08
51	LCT	LCT	LCT	LCT	LCT	35.56	135.21	123.09	81.61	106.62	742.18	298.55	1257.00
52	SCT	SCT	SCT	SCT	SCT	28.90	88.72	132.91	209.13	54.14	734.97	300.82	1253.41
53	S/OPN	S/OPN	S/OPN	S/OPN	S/OPN	36.31	138.93	122.95	179.43	197.42	854.40	410.52	1354.08
54	SRCT	SRCT	SRCT	SRCT	SRCT	27.82	102.21	84.40	176.03	156.54	695.95	267.29	1193.50
55	LRCT	LRCT	LRCT	LRCT	LRCT	36.31	138.93	122.95	179.43	197.42	854.40	410.52	1354.08
56	MMOD	MMOD	MMOD	MMOD	MMOD	34.72	97.22	126.12	218.21	115.33	801.21	355.83	1264.25



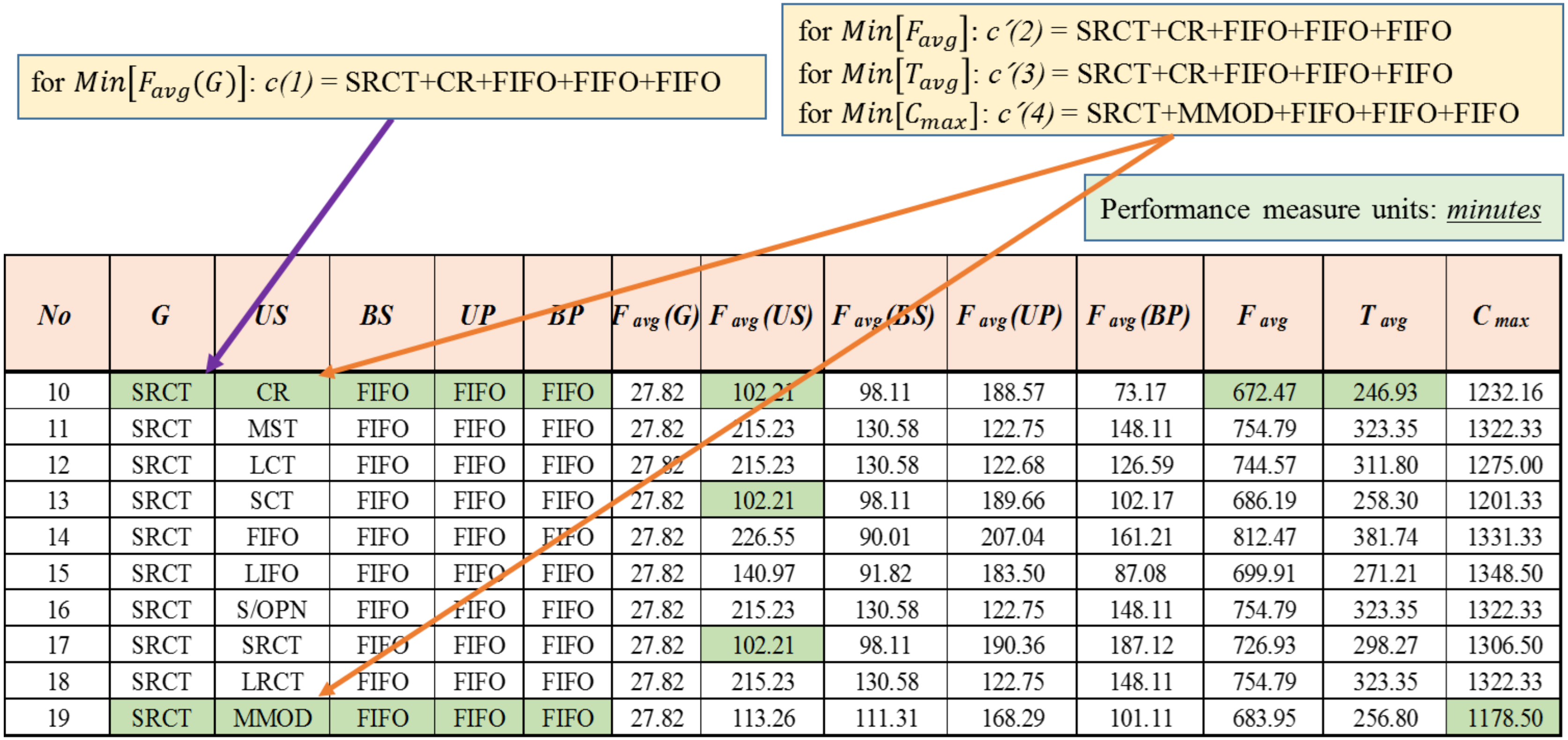
for $Min[F_{avg}(G)]$: *c(1)* = SRCT+FIFO+FIFO+FIFO+FIFO

for $Min[F_{avg}]$: *c(2)* = LCT+FIFO+FIFO+FIFO+FIFO
for $Min[T_{avg}]$: *c(3)* = LCT+FIFO+FIFO+FIFO+FIFO
for $Min[C_{max}]$: *c(4)* = SRCT+FIFO+FIFO+FIFO+FIFO

Performance measure units: *minutes*

No	*G*	*US*	*BS*	*UP*	*BP*	$F_{avg}(G)$	$F_{avg}(US)$	$F_{avg}(BS)$	$F_{avg}(UP)$	$F_{avg}(BP)$	F_{avg}	T_{avg}	C_{max}
1	CR	FIFO	FIFO	FIFO	FIFO	28.15	219.18	97.21	205.39	168.81	826.98	392.54	1352.83
2	MST	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
3	LCT	FIFO	FIFO	FIFO	FIFO	35.56	136.69	113.46	145.36	135.01	798.61	357.36	1334.16
4	SCT	FIFO	FIFO	FIFO	FIFO	28.90	181.21	87.18	196.39	144.09	816.82	382.00	1355.83
5	S/OPN	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
6	SRCT	FIFO	FIFO	FIFO	FIFO	27.82	226.55	90.01	207.04	161.21	812.47	381.74	1331.33
7	LRCT	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
8	MMOD	FIFO	FIFO	FIFO	FIFO	34.72	179.77	104.33	138.35	169.42	819.49	372.20	1355.25

[Figure 6.1] IWMF step 1 comparison and making choice of composite dispatching rules



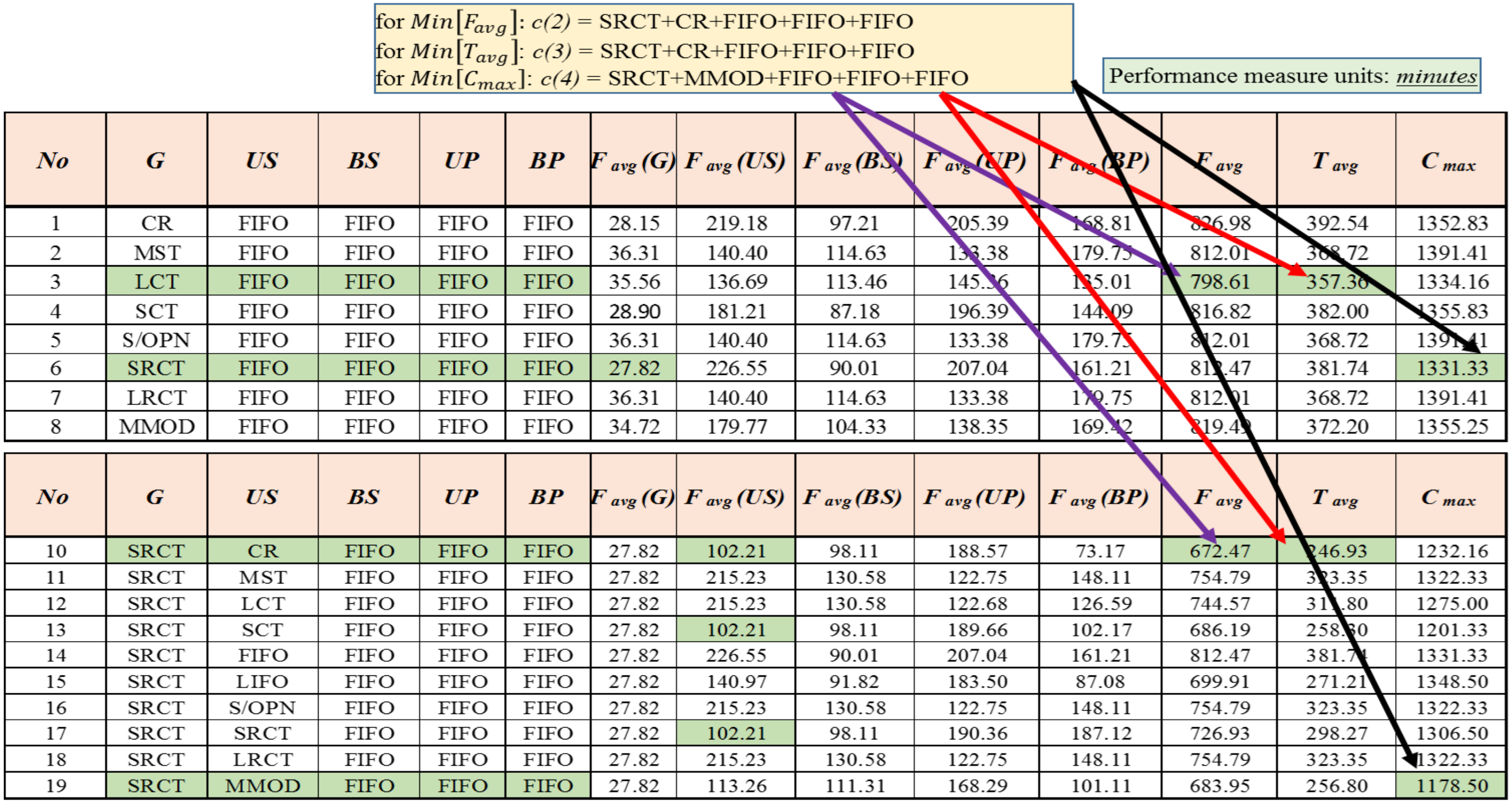
for $Min[F_{avg}(G)]$: $c(1)$ = SRCT+CR+FIFO+FIFO+FIFO

for $Min[F_{avg}]$: $c'(2)$ = SRCT+CR+FIFO+FIFO+FIFO
for $Min[T_{avg}]$: $c'(3)$ = SRCT+CR+FIFO+FIFO+FIFO
for $Min[C_{max}]$: $c'(4)$ = SRCT+MMOD+FIFO+FIFO+FIFO

Performance measure units: *minutes*

No	***G***	***US***	***BS***	***UP***	***BP***	$F_{avg}(G)$	$F_{avg}(US)$	$F_{avg}(BS)$	$F_{avg}(UP)$	$F_{avg}(BP)$	F_{avg}	T_{avg}	C_{max}
10	SRCT	CR	FIFO	FIFO	FIFO	27.82	102.21	98.11	188.57	73.17	672.47	246.93	1232.16
11	SRCT	MST	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
12	SRCT	LCT	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.68	126.59	744.57	311.80	1275.00
13	SRCT	SCT	FIFO	FIFO	FIFO	27.82	102.21	98.11	189.66	102.17	686.19	258.30	1201.33
14	SRCT	FIFO	FIFO	FIFO	FIFO	27.82	226.55	90.01	207.04	161.21	812.47	381.74	1331.33
15	SRCT	LIFO	FIFO	FIFO	FIFO	27.82	140.97	91.82	183.50	87.08	699.91	271.21	1348.50
16	SRCT	S/OPN	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
17	SRCT	SRCT	FIFO	FIFO	FIFO	27.82	102.21	98.11	190.36	187.12	726.93	298.27	1306.50
18	SRCT	LRCT	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
19	SRCT	MMOD	FIFO	FIFO	FIFO	27.82	113.26	111.31	168.29	101.11	683.95	256.80	1178.50

[Figure 6.2] IWMF step 2 comparison and making choices of composite dispatching rules



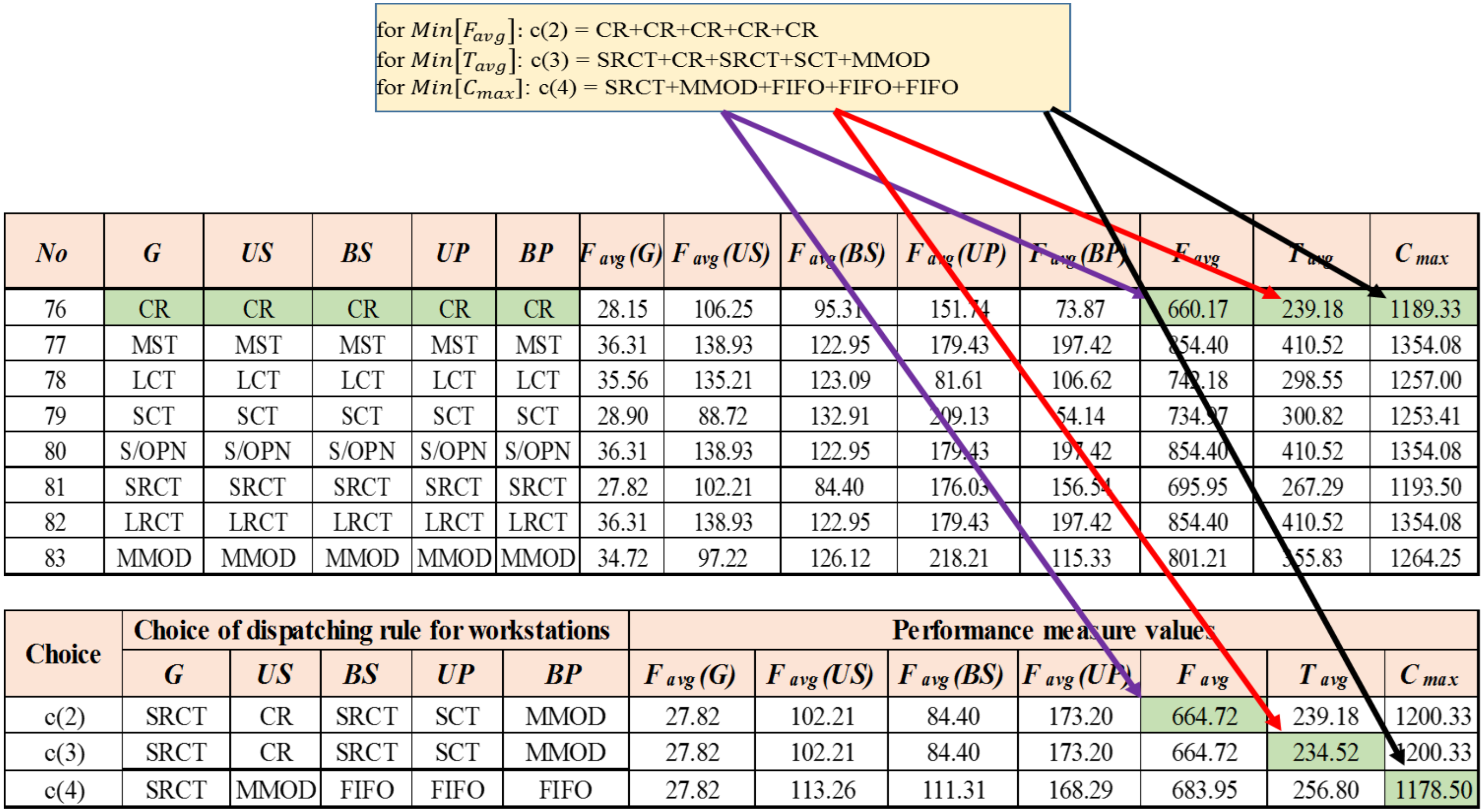
for $Min[F_{avg}]$: *c(2)* = SRCT+CR+FIFO+FIFO+FIFO
for $Min[T_{avg}]$: *c(3)* = SRCT+CR+FIFO+FIFO+FIFO
for $Min[C_{max}]$: *c(4)* = SRCT+MMOD+FIFO+FIFO+FIFO

Performance measure units: *minutes*

No	***G***	***US***	***BS***	***UP***	***BP***	F_{avg} ***(G)***	F_{avg} ***(US)***	F_{avg} ***(BS)***	F_{avg} ***(UP)***	F_{avg} ***(BP)***	F_{avg}	T_{avg}	C_{max}
1	CR	FIFO	FIFO	FIFO	FIFO	28.15	219.18	97.21	205.39	168.81	826.98	392.54	1352.83
2	MST	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
3	LCT	FIFO	FIFO	FIFO	FIFO	35.56	136.69	113.46	145.36	135.01	798.61	357.36	1334.16
4	SCT	FIFO	FIFO	FIFO	FIFO	28.90	181.21	87.18	196.39	144.09	816.82	382.00	1355.83
5	S/OPN	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
6	SRCT	FIFO	FIFO	FIFO	FIFO	27.82	226.55	90.01	207.04	161.21	812.47	381.74	1331.33
7	LRCT	FIFO	FIFO	FIFO	FIFO	36.31	140.40	114.63	133.38	179.75	812.01	368.72	1391.41
8	MMOD	FIFO	FIFO	FIFO	FIFO	34.72	179.77	104.33	138.35	169.42	819.49	372.20	1355.25

No	***G***	***US***	***BS***	***UP***	***BP***	F_{avg} ***(G)***	F_{avg} ***(US)***	F_{avg} ***(BS)***	F_{avg} ***(UP)***	F_{avg} ***(BP)***	F_{avg}	T_{avg}	C_{max}
10	SRCT	CR	FIFO	FIFO	FIFO	27.82	102.21	98.11	188.57	73.17	672.47	246.93	1232.16
11	SRCT	MST	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
12	SRCT	LCT	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.68	126.59	744.57	311.80	1275.00
13	SRCT	SCT	FIFO	FIFO	FIFO	27.82	102.21	98.11	189.66	102.17	686.19	258.30	1201.33
14	SRCT	FIFO	FIFO	FIFO	FIFO	27.82	226.55	90.01	207.04	161.21	812.47	381.74	1331.33
15	SRCT	LIFO	FIFO	FIFO	FIFO	27.82	140.97	91.82	183.50	87.08	699.91	271.21	1348.50
16	SRCT	S/OPN	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
17	SRCT	SRCT	FIFO	FIFO	FIFO	27.82	102.21	98.11	190.36	187.12	726.93	298.27	1306.50
18	SRCT	LRCT	FIFO	FIFO	FIFO	27.82	215.23	130.58	122.75	148.11	754.79	323.35	1322.33
19	SRCT	MMOD	FIFO	FIFO	FIFO	27.82	113.26	111.31	168.29	101.11	683.95	256.80	1178.50

[Figure 6.3] Comparison between step 1 and step 2 and making choices of composite dispatching rules



for $Min[F_{avg}]$: c(2) = CR+CR+CR+CR+CR
for $Min[T_{avg}]$: c(3) = SRCT+CR+SRCT+SCT+MMOD
for $Min[C_{max}]$: c(4) = SRCT+MMOD+FIFO+FIFO+FIFO

No	*G*	*US*	*BS*	*UP*	*BP*	F_{avg} *(G)*	F_{avg} *(US)*	F_{avg} *(BS)*	F_{avg} *(UP)*	F_{avg} *(BP)*	F_{avg}	T_{avg}	C_{max}
76	CR	CR	CR	CR	CR	28.15	106.25	95.31	151.74	73.87	660.17	239.18	1189.33
77	MST	MST	MST	MST	MST	36.31	138.93	122.95	179.43	197.42	854.40	410.52	1354.08
78	LCT	LCT	LCT	LCT	LCT	35.56	135.21	123.09	81.61	106.62	742.18	298.55	1257.00
79	SCT	SCT	SCT	SCT	SCT	28.90	88.72	132.91	209.13	54.14	734.97	300.82	1253.41
80	S/OPN	S/OPN	S/OPN	S/OPN	S/OPN	36.31	138.93	122.95	179.43	197.42	854.40	410.52	1354.08
81	SRCT	SRCT	SRCT	SRCT	SRCT	27.82	102.21	84.40	176.03	156.54	695.95	267.29	1193.50
82	LRCT	LRCT	LRCT	LRCT	LRCT	36.31	138.93	122.95	179.43	197.42	854.40	410.52	1354.08
83	MMOD	MMOD	MMOD	MMOD	MMOD	34.72	97.22	126.12	218.21	115.33	801.21	355.83	1264.25

Choice	**Choice of dispatching rule for workstations**					**Performance measure values**						
	G	*US*	*BS*	*UP*	*BP*	F_{avg} *(G)*	F_{avg} *(US)*	F_{avg} *(BS)*	F_{avg} *(UP)*	F_{avg}	T_{avg}	C_{max}
c(2)	SRCT	CR	SRCT	SCT	MMOD	27.82	102.21	84.40	173.20	664.72	239.18	1200.33
c(3)	SRCT	CR	SRCT	SCT	MMOD	27.82	102.21	84.40	173.20	664.72	234.52	1200.33
c(4)	SRCT	MMOD	FIFO	FIFO	FIFO	27.82	113.26	111.31	168.29	683.95	256.80	1178.50

[Figure 6.4] Comparisons between step 3 and results from step2 and making final choices of composite dispatching rules

6.2.1 Observations

The ultimate choices of *composite dispatching rules* for each of the considered performance measures are shown in [Table 6.2] for *1 day 1 shift* criteria.

[Table 6.2] Final choice of dispatching strategy for each workstation

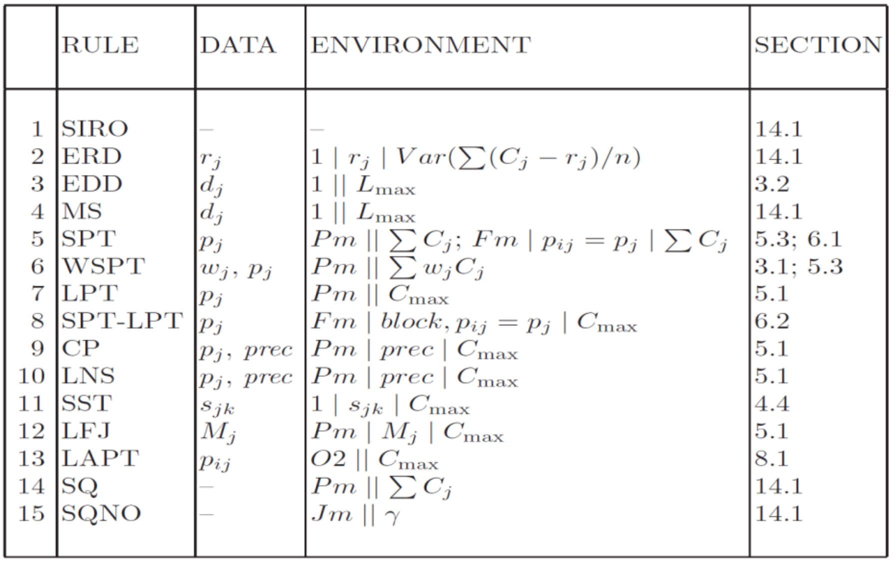
Objective	**choices**	**Choice of dispatching rule for workstations (*w*)**					**Performance measure values (*mins*)**		
							***t* = 1 day, 1 shift**		
		G	***US***	***BS***	***UP***	***BP***	F_{avg}	T_{avg}	C_{max}
$Min[F_{avg}]$	*c*(2)	CR	CR	CR	CR	CR	660.17	239.18	1189.33
$Min[T_{avg}]$	*c*(3)	SRCT	CR	SRCT	SCT	MMOD	664.72	234.52	1200.33
$Min[C_{max}]$	*c*(4)	SRCT	MMOD	FIFO	FIFO	FIFO	683.95	256.80	1178.50

The following observations can be made from the above outcome:

- Dominant strategy isn't found for the performance measures considered
- Significant differences performance measures values cannot be seen because the simulation experiment is performed for *1 day 1 shift* production target. Significant differences can be obtained if the experiment is run for more time horizon.

Other than those above mentioned observations, another interesting analysis can be made. It is well known that shortest processing time, SPT (here shortest cycle time, SCT) works well for mean flow time performance measure in flow shop or job shop environment, though it was not the case in this study. Obviously, a proper justification is need in this respect. Actually SPT (or, SCT) is known to minimize the mean flow time at a single station shop under conditions of deterministic operating times [Klafehn, Weinroth, & Boronico, 1996]. [Baker & Trietsch, 2013] mentioned that flow time is minimized by shortest processing time (SPT) sequencing in single machine sequencing. SPT is relatively effective when due dates are very tight but not when due dates are loose. Thus, a particular experimental comparison might find SPT performance to be good or bad, depending on how tight the due dates are set [Baker & Trietsch]. [Pinedo, 2008] also given an overview of better known dispatching rules lead to optimal schedules in certain machine environments, as shown in [Table 6.3]. From this overview, it is seen that SPT works well in case of parallel machine flow shop with makespan criteria.

[Table 6.3] dispatching rules for certain machine environments



	RULE	DATA	ENVIRONMENT	SECTION
1	SIRO	–	–	14.1
2	ERD	r_j	$1 \mid r_j \mid Var(\sum(C_j - r_j)/n)$	14.1
3	EDD	d_j	$1 \mid\mid L_{\max}$	3.2
4	MS	d_j	$1 \mid\mid L_{\max}$	14.1
5	SPT	p_j	$Pm \mid\mid \sum C_j;\ Fm \mid p_{ij} = p_j \mid \sum C_j$	5.3; 6.1
6	WSPT	$w_j,\ p_j$	$Pm \mid\mid \sum w_j C_j$	3.1; 5.3
7	LPT	p_j	$Pm \mid\mid C_{\max}$	5.1
8	SPT-LPT	p_j	$Fm \mid block, p_{ij} = p_j \mid C_{\max}$	6.2
9	CP	$p_j,\ prec$	$Pm \mid prec \mid C_{\max}$	5.1
10	LNS	$p_j,\ prec$	$Pm \mid prec \mid C_{\max}$	5.1
11	SST	s_{jk}	$1 \mid s_{jk} \mid C_{\max}$	4.4
12	LFJ	M_j	$Pm \mid M_j \mid C_{\max}$	5.1
13	LAPT	p_{ij}	$O2 \mid\mid C_{\max}$	8.1
14	SQ	–	$Pm \mid\mid \sum C_j$	14.1
15	SQNO	–	$Jm \mid\mid \gamma$	14.1

On the other hand, it is hard to claim that SPT (or SCT) will also outperform other dispatching policies in a complex flow shop or job shop (e.g. HFS or HJS) due to the existence of *parallel machines*. Since in HFS (or HJS) each workstation depends on its predecessor workstations and there are the cases of having unequal parallel machine capacities, so mean flow time (F_{avg}) in one workstation can actually affect the mean flow time of the successor workstation's mean flow time. For example, in this study, as mentioned in [Table 5.3], *w(1)* (Grinding) comprises with 3 *identical parallel (alternate)* machines, whereas *w(2)* (Up Smoothing) and *w(3)* (Bottom Smoothing) each has 2 *identical parallel (alternate)* machines. So, obviously lenses in *w(1)* exiting faster (compared to single machine case) due to availability of parallel machines, and arriving to the next successive workstations [*w(2)* and *w(3)*]. It is also noteworthy to mention that while SCT is being used in *w(1)*, obviously lenses will tend to exit w(1) with *lesser* mean flow time compared to other dispatching rules (e.g. LCT). As a result, lenses would tend to arrive successive workstations *earlier* which eventually assign *shorter arrival time* to the arriving sublots of lenses and as flow time of job j at workstation w of stage s is calculated by $F_{jsw} = C_{jsw} - ArrT_{jsw}$, so shorter arrival time would definitely *increase* the successive workstations' mean flow time. That is why for such a complex architecture like HFS or hybrid job shop it is difficult to anticipate the optimal or best dispatching policy without a *simulation* scheme specially when the choices comprise with a combination of elementary dispatching rules. Because of such reasons, the best dispatching policy choice might vary from case by case in HFS or hybrid job shop like in this study, a combination of (CR+ CR+ CR+ CR+ CR) dispatching policy shows better efficiency for mean flow time performance measure.

The simulation is run again for *5 days 2 shifts* criteria to really investigate if there are any significant differences arise from longer production run phenomena. The corresponding outcome of this simulation run is shown in [Table 6.4]. It is clearly seen that after running the model for more production time horizon, the gap of performance measure values increased among the three (3) best choices of composite dispatching rules. The significance of the performance measures' differences between 1 day 1 shift and 5 days 2 shifts production run models is shown in [Table 6.5].

[Table 6.4] Performance measure values for final choices [*5 days 2 shifts*]

Objective	**Choices**	**Choice of dispatching rule for workstations (*w*)**					**Performance measure values (*mins*)**		
							***t* = 5 days, 2 shifts**		
		G	***US***	***BS***	***UP***	***BP***	F_{avg}	T_{avg}	C_{max}
$Min[F_{avg}]$	*c(2)*	CR	CR	CR	CR	CR	6087.89	1945.93	12069.00
$Min[T_{avg}]$	*c(3)*	SRCT	CR	SRCT	SCT	MMOD	6196.52	1891.62	12076.83
$Min[C_{max}]$	*c(4)*	SRCT	MMOD	FIFO	FIFO	FIFO	7963.71	3534.53	11439.58

[Table 6.5] Percentage (%) differences of performance measure values from 1 day 1 shift to 5 days, 2 shifts

Performance measures	**Choices**	**Compared to**		
		c(2)	*c(3)*	*c(4)*
F_{avg}	*c(2)*	…..	2.16%(-)	38.58%(-)
T_{avg}	*c(3)*	1.03%(-)	……	33.76%(-)
C_{max}	*c(4)*	12.89%(-)	12.82%(-)	

From the above Table, it's to be noted that the difference of mean flow time (F_{avg}) between *c(2)* and *c(3)* is increased to 2.16%. Between *c(2)* and *c(4)* its increased to 38.58%. On the other hand, difference of mean tardiness (T_{avg}) between *c(3)* and *c(2)* is increased to 1.03%. Between *c(3)* and *c(4)* it's increased to 33.76%. Difference of makespan (C_{max}) between *c(4)* and *c(2)* is increased to 12.89%. between *c(4)* and *c(3)* it's increased to 12.82%. Although even after running for 5 days 2 shifts, some of the differences are still not much significant. But it's believed to show more significant differences if the model is run for longer production periods (e.g., 1 month, 6 months or even 1 year).

6.3 Varying batch size effects

As mentioned in Section 3.2, another objective of this study is to analyze the impact of varying batch size on *mean machine utilization rate* and *total machine setup time*. The experiment is done by taking 5 instances of different batch sizes for all types of lenses as shown in [Table 6.6].

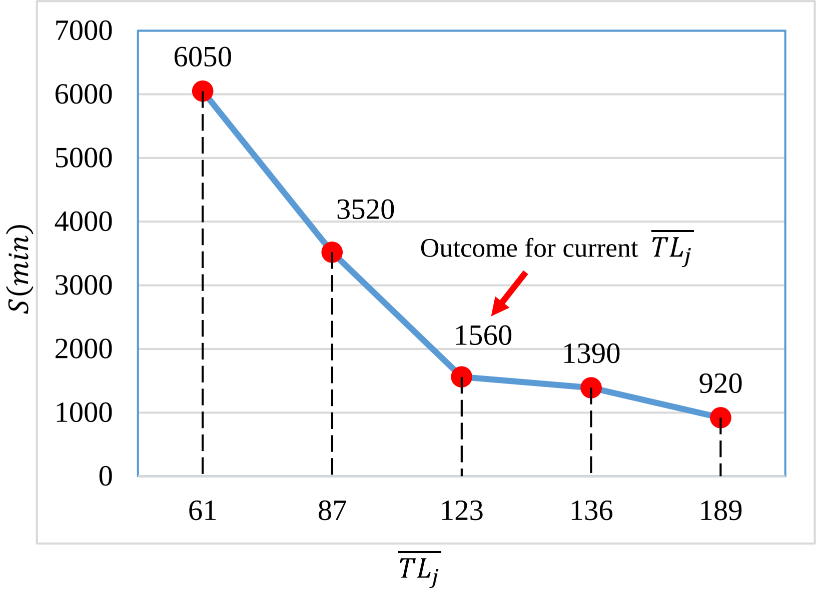
[Table 6.6] Performance measures with regard to varying batch sizes

Instances	***LT(j)***	**TL_j**	**$\overline{TL_j}$**	**Work Station**	**Machine Setup time *(min)***	**Total Machine Setup time *S (min)***	**Mc Utilization *(U_m)***	**U_{avg} (%)**
1	LT1	30	123	G	650	1560	0.6	60%
	LT2	48		US	300		0.7	
	LT3	120		BS	140		0.56	
	LT4	208		UP	280		0.44	
	LT5	208		BP	190		0.68	
2	LT1	25	87	G	1460	3520	0.8	73%
	LT2	30		US	610		0.9	
	LT3	80		BS	430		0.7	
	LT4	150		UP	570		0.52	
	LT5	150		BP	450		0.75	
3	LT1	15	61	G	2640	6050	0.99	85%
	LT2	24		US	1140		0.99	
	LT3	60		BS	580		0.78	
	LT4	104		UP	1000		0.6	
	LT5	104		BP	690		0.88	

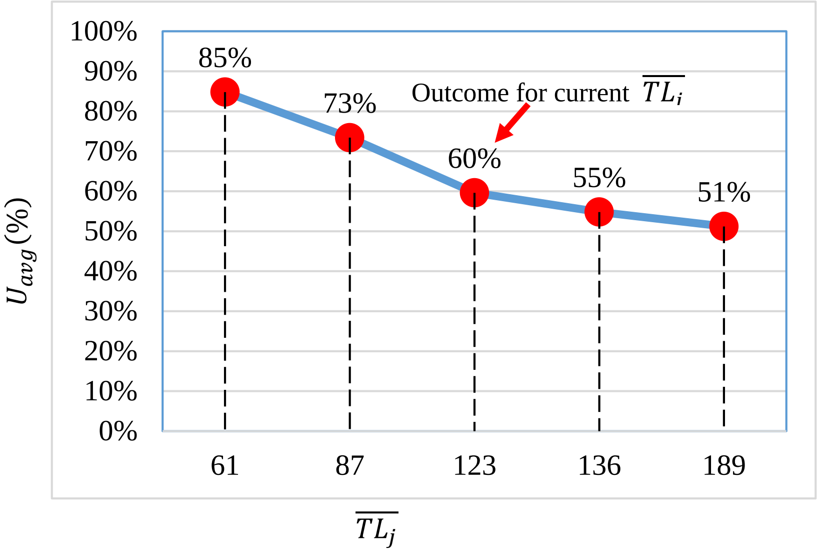
Instances	***LT(j)***	TL_j	$\overline{TL_j}$	**Work Station**	**Machine Setup time *(min)***	**Total Machine Setup time *S (min)***	**Mc Utilization *(U_m)***	U_{avg} *(%)*
4	LT1	40	136	G	540	1390	0.55	55%
	LT2	60		US	280		0.69	
	LT3	140		BS	130		0.49	
	LT4	220		UP	260		0.4	
	LT5	220		BP	180		0.61	
5	LT1	55	189	G	330	920	0.54	51%
	LT2	100		US	190		0.67	
	LT3	190		BS	100		0.48	
	LT4	300		UP	170		0.32	
	LT5	300		BP	130		0.55	

6.3.1 Observations

The outcomes shown in [Table 6.6] are plotted in the following figures:



[Figure 6.5] Average batch size, $\overline{TL_j}$ vs Total machine setup time, $S(min)$



[Figure 6.6] Average batch size, $\overline{TL_j}$ vs Mean machine utilization, $U_{avg}(\%)$

The observations made from [Figure 6.5] and [Figure 6.6], are as follows:

- Decreasing the batch size of each lenses' sublot has a negative impact on total machine setup time but positive impact on mean machine utilization. That means if the average batch size is decreased, then the total machine setup time in the system will increase because of the frequent change og the machine tools. On the contrary, the mean machine utilization will increase because of the lesser waiting time of the subsequent machines in the HFS to get the next sublot with lesser batch sizes.
- Increasing the batch size of each lenses' sublot has a positive impact on total machine setup time but negative impact on machine utilization rate. That means if the average batch size is increased, then the total machine setup time in the system will decrease because of less frequent change of machine tools. On the contrary, mean machine utilization will decrease because of greater waiting time of the subsequent machines in the HFS to get the next sublot with bigger batch sizes.

Chapter 7. Conclusion

7.1 Contribution

A *flexible simulation modeling framework* is developed to support *real time decision making* for HFS scheduling problem. Most of the commonly used dispatching rules are integrated in to the simulation model and analyzed by the simulation framework. A heuristic named *IWMF* is used to mitigate the challenge of how to reduce the number of combinations of integrated dispatching policies into different workstations and thus to reduce the number of comparisons among them with regard to the performance measures. Near optimal composite dispatching strategy is established for each of the performance measures considered in this study. From this study it is revealed that it is much wiser to use well established dispatching policies for enhancing the performances of the HFS rather than just using FIFO or any unplanned random strategy. By utilizing the developed simulation interface in this study, production manager shall be able to set the better job scheduling policy for day-to-day operations of multiple job orders in under any certain type of HFS characteristic. At last, effect of varying batch size on *total machine setup time* and *mean machine utilization* is shown. It is found that *total machine setup time* and *mean machine utilization* heavily depend on *varying batch size* and these two performance measures show a certain pattern based on the varying batch size. The simulation modelling framework used in this study, can actually help researchers in future to further investigate any type of HFS and to find near optimal or optimal strategy regarding to the HFS scheduling problem.

7.2 Limitations

This study focuses on *real time decision making phenomena* for HFS scheduling problem. Although possible *random or uncertain* attributes and their *distributions* are integrated into the simulation model but in regard to the day-to-day *decision making* of HFS scheduling problem*, real time deterministic values* are used in the simulation experiment. Machine breakdown criteria were not considered in the experiment only to avoid inconsistent outcome.

The heuristic used in this study required to be verified with more experiments comprising more number of combinations among the dispatching rules. Decision making pattern under stochastic environment is not identified by the studied model. *Dominant composite dispatching policy* has not been found for considered performance measures.

Although certain observations are made regarding the varying batch size impact on certain performance measures, but optimum batch quantity for the consistent sublot is not proposed in this study.

7.3 Future work

All the possible uncertainties like machine breakdown, machine setup time variations etc. needs to be tested in the simulation model for decision making of HFS scheduling problem. Statistical analysis like sensitivity and variance analysis needs to be shown if randomness of the parameters is considered. Since the obtained dispatching strategy is near optimal, so there is always scope for further improvement. Proposed heuristics should be improved to get more closer to the optimal strategy. Optimum batch quantity is to be found for more appropriate decision making regarding to the HFS scheduling problem. Last but not least if varying batch quantity show a certain impact on the HFS performances then at the same time a *flexible job holder* to be designed for carrying the sublot with *varying batch size.*

Bibliography

Adler, L., Fraiman, N., Kobacker, E., Pinedo, M., Plotnicoff, J. C., & Wu, T. P. (1993). BPSS: a scheduling support system for the packaging industry. *Operations research, 41*(4), 641-648.

Al-Turki, U. M., Arifusalam, S., El-Seliaman, M., & Khan, M. (2011). *Resource allocation, batching and dispatching in a stochastic flexible job shop.* Paper presented at the Advanced Materials Research.

Al-Turki, U. M., Saleh, H., Deyab, T., & Almoghathawi, Y. (2012). *Resource Allocation and Job Dispatching for Unreliable Flexible Flow Shop Manufacturing System.* Paper presented at the Advanced Materials Research.

Allahverdi, A., & Tatari, M. F. (1996). Simulation of different rules in stochastic flowshops. *Computers & Industrial Engineering, 31*(1), 209-212.

Altiok, T., & Melamed, B. (2010). *Simulation modeling and analysis with Arena*: Academic press.

Andres, C., Gomez, P., & Garcia-Sabater, J. P. (2006). *Comparing dispatching rules in dynamic hybrid flow shops.* Paper presented at the Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE Conference on.

Baker, K. R. (1974). *Introduction to sequencing and scheduling*: John Wiley & Sons.

Baker, K. R., & Trietsch, D. Simulation Models for the Dynamic Job Shop. *Principles of Sequencing and Scheduling*, 249-371.

Baker, K. R., & Trietsch, D. (2013). *Principles of sequencing and scheduling*: John Wiley & Sons.

Boschetti, M. A., Maniezzo, V., Roffilli, M., & Röhler, A. B. (2009). *Matheuristics: Optimization, simulation and control.* Paper presented at the International Workshop on Hybrid Metaheuristics.

Brah, S. A. (1996). A comparative analysis of due date based job sequencing rules in a flow shop with multiple processors. *Production Planning & Control, 7*(4), 362-373.

Brah, S. A., & Wheeler, G. E. (1998). Comparison of scheduling rules in a flow shop with multiple processors: A simulation. *Simulation, 71*(5), 302-311.

Burtseva, L., Parra, R. R., & Yaurima, V. (2010). *Scheduling methods for hybrid flow shops with setup times*: INTECH Open Access Publisher Rijeka.

Chtourou, H., Masmoudi, W., & Maalej, A. (2005). An expert system for manufacturing systems machine selection. *Expert Systems with Applications, 28*(3), 461-467.

de Fatima Morais, M., Boiko, T. J. P., dos Santos Coelho, L., da Rocha, R. P., & Paraíso, P. R. (2014). Multicriteria hybrid flow shop scheduling problem: literature review, analysis, and future research. *Independent Journal of Management & Production, 5*(4), 1004-1031.

Grangeon, N., Tanguy, A., & Tchernev, N. (1999). Generic simulation model for hybrid flow-shop. *Computers & Industrial Engineering, 37*(1-2), 207-210.

Guinet, A., Solomon, M., Kedia, P., & Dussauchoy, A. (1996). A computational study of heuristics for two-stage flexible flowshops. *International Journal of Production Research, 34*(5), 1399-1415.

Gupta, J., Hariri, A., & Potts, C. (1997). Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. *Annals of Operations Research, 69*, 171-191.

Gupta, J. N. (1988). Two-stage, hybrid flowshop scheduling problem. *Journal of the Operational Research Society, 39*(4), 359-364.

Johnson, L. A., & Montgomery, D. C. (1974). *Operations research in production planning, scheduling, and inventory control* (Vol. 6): Wiley

New York.

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

Jun, S., & Park, J. (2015). A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry. *Expert Systems with Applications, 42*(15), 6196-6204.

Kadipasaoglu, S., Xiang, W., & Khumawala, B. (1997). A comparison of sequencing rules in static and dynamic hybrid flow systems. *International Journal of Production Research, 35*(5), 1359-1384.

Kelton, W., Sadowski, R., & Sturnock, D. (2004). Simulation with Arena 3rd International Edition: McGraw-Hill, New York.

Klafehn, K., Weinroth, J., & Boronico, J. (1996). *Computer simulation in operations management*: Greenwood Publishing Group.

Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. *Computers & Industrial Engineering, 37*(1-2), 57-61.

Minella, G., Ruiz, R., & Ciavotta, M. (2008). A review and evaluation of multiobjective algorithms for the flowshop scheduling problem. *INFORMS Journal on Computing, 20*(3), 451-471.

Moccellin, J., & Santos, M. (2000). An adaptive hybrid metaheuristic for permutation flowshop scheduling. *Control and Cybernetics, 29*(3), 761-771.

Montoya-Torres, J. R., Solano-Charris, E. L., & Muñoz-Villamizar, A. (2016). Assessing the Performance of Dispatching Policies for Hybrid Flowshop Manufacturing Systems. *IFAC-PapersOnLine, 49*(31), 109-113.

Moon, I., Lee, S., & Bae, H. (2008). Genetic algorithms for job shop scheduling problems with alternative routings. *International Journal of*

Production Research, 46(10), 2695-2705.

Morais, M. d. F., & Moccellin, J. V. (2010). Constructive heuristics methods to minimizing work in process in environment production hybrid flow shop with asymmetric sequence dependent setup times. *Gestão & Produção, 17*(2), 367-375.

Morton, T., & Pentico, D. W. (1993). *Heuristic scheduling systems: with applications to production systems and project management* (Vol. 3): John Wiley & Sons.

Namakshenas, M., & Sahraeian, R. (2013). Real-time scheduling of a flexible manufacturing system using a two-phase machine learning algorithm. *International Journal of Engineering-Transactions C: Aspects, 26*(9), 1067.

Paul, R. (1979). A production scheduling problem in the glass-container industry. *Operations research, 27*(2), 290-302.

Pinedo, M. (2002). *Scheduling : theory, algorithms, and systems / Michael Pinedo* (2nd ed.. ed.). Upper Saddle, N.J.: Upper Saddle, N.J. : Prentice Hall.

Pinedo, M. (2008). *Scheduling : theory, algorithms, and systems / Michael L. Pinedo* (3rd ed.. ed.). New York ; London: New York ; London : Springer.

Ribas, I., Leisten, R., & Framiñan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research, 37*(8), 1439-1454.

Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research, 205*(1), 1-18.

Salvador, M. S. (1973). *A solution to a special class of flow shop scheduling problems.* Paper presented at the Symposium on the theory of scheduling and its applications.

Shannon, R., & Johannes, J. D. (1976). Systems simulation: the art and science. *IEEE transactions on systems, man, and cybernetics, 6*(10), 723-724.

Sioud, A., Gagné, C., & Gravel, M. (2014). Metaheuristics for Solving a Hybrid Flexible Flowshop Problem with Sequence-Dependent Setup Times. In P. Siarry, L. Idoumghar & J. Lepagnot (Eds.), *Swarm Intelligence Based Optimization: First International Conference, ICSIBO 2014, Mulhouse, France, May 13-14, 2014. Revised Selected Papers* (pp. 9-25). Cham: Springer International Publishing.

Sriskandarajah, C., & Sethi, S. P. (1989). Scheduling algorithms for flexible flowshops: worst and average case performance. *European Journal of Operational Research, 43*(2), 143-160.

T'kindt, V., & Billaut, J.-C. (2006). *Multicriteria scheduling: theory, models and algorithms*: Springer Science & Business Media.

Takus, D. A., & Profozich, D. M. (1997). ARENA software tutorial: Institute of Electrical and Electronics Engineers (IEEE).

Tsubone, H., Ohba, M., Takamuki, H., & Miyake, Y. (1993). A production scheduling system for a hybrid flow shop—a case study. *Omega, 21*(2), 205-214.

Uetake, T., Tsubone, H., & Ohba, M. (1995). A production scheduling system in a hybrid flow shop. *International Journal of Production Economics, 41*(1-3), 395-398.

Verma, S., & Dessouky, M. (1999). Multistage hybrid flowshop scheduling with identical jobs and uniform parallel machines. *Journal of scheduling, 2*(3), 135-150.

Vignier, A., Billaut, J., & Proust, C. (1999). Hybrid flowshop scheduling problems: State of the art. *Rairo-Recherche Operationnelle-Operations Research, 33*(2), 117-183.

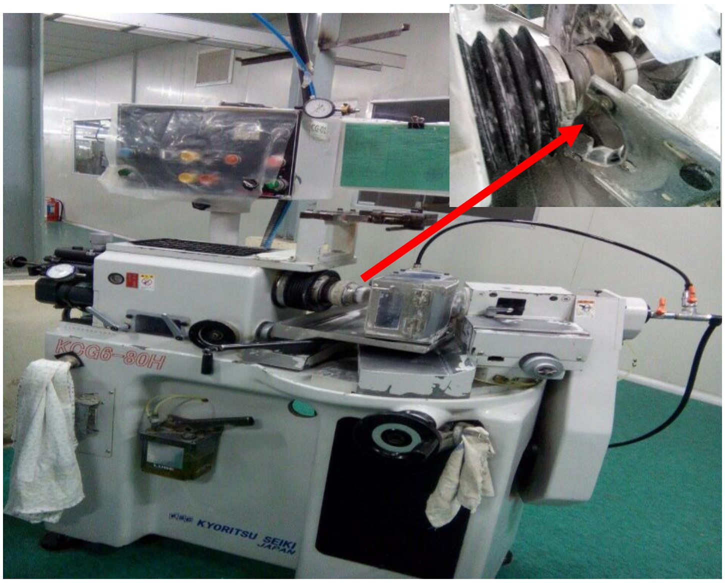
Wang, L., Pan, Q.-K., & Tasgetiren, M. F. (2010). Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms. *Expert Systems with Applications, 37*(12), 7929-7936.

Yenisey, M. M., & Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega, 45*, 119-135.

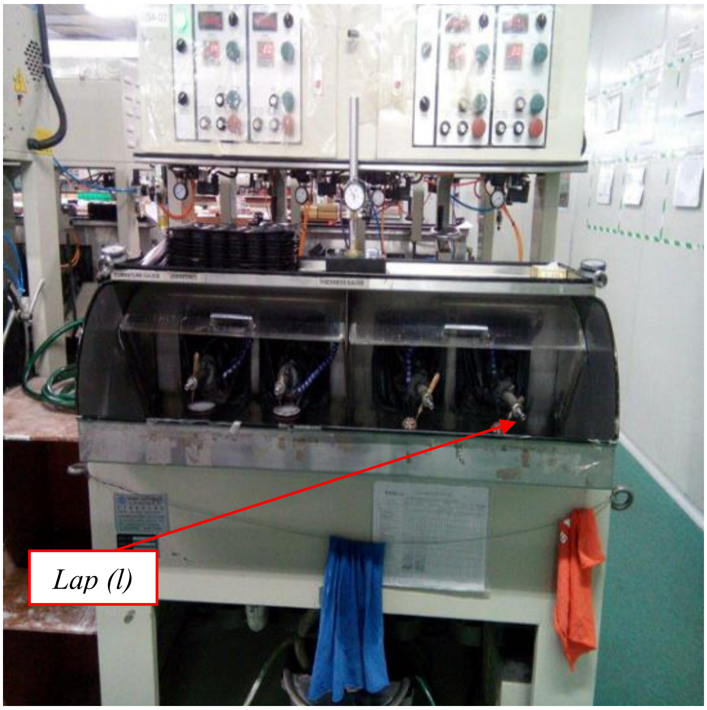
Yücel, N., Şen, D., & Kılıç, S. (2004). *Integrating an agent based flexible manufacturing cell framework with simulation: A pilot implementation.* Paper presented at the Proceedings of The Eleventh International Conference on Machine Design and Production.

APPENDIX A

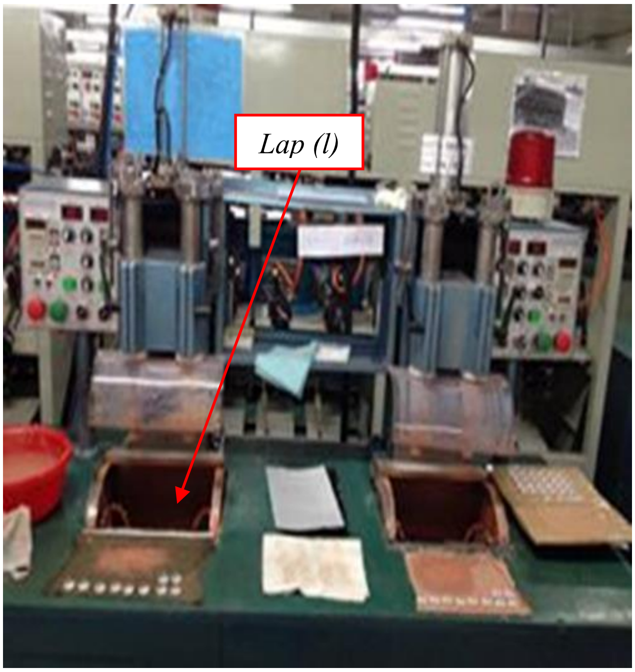
[Figure A.1] Identical parallel G MC



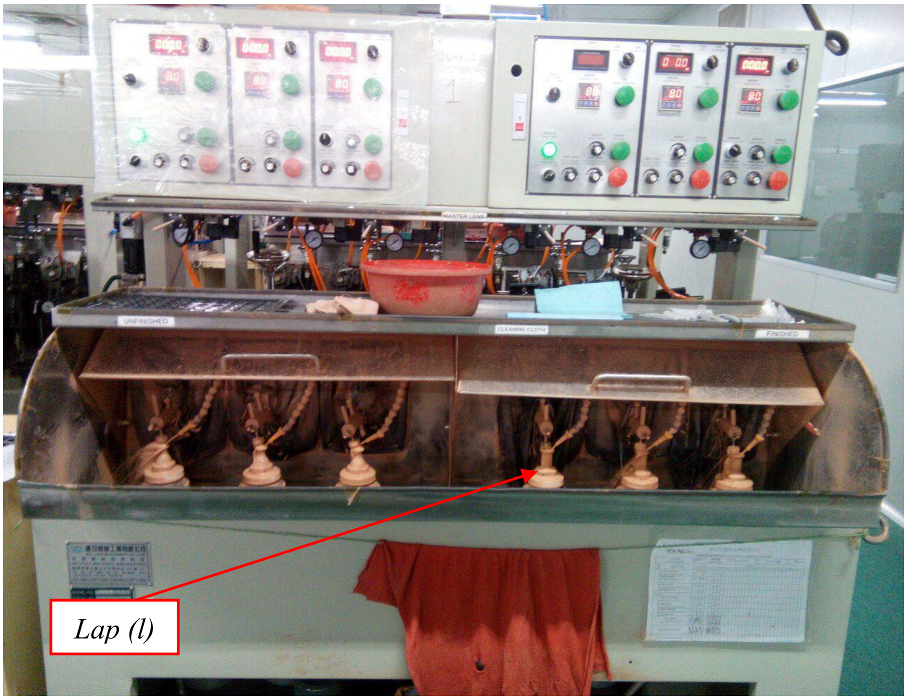
[Figure A.2] G MC



[Figure A.3] US MC



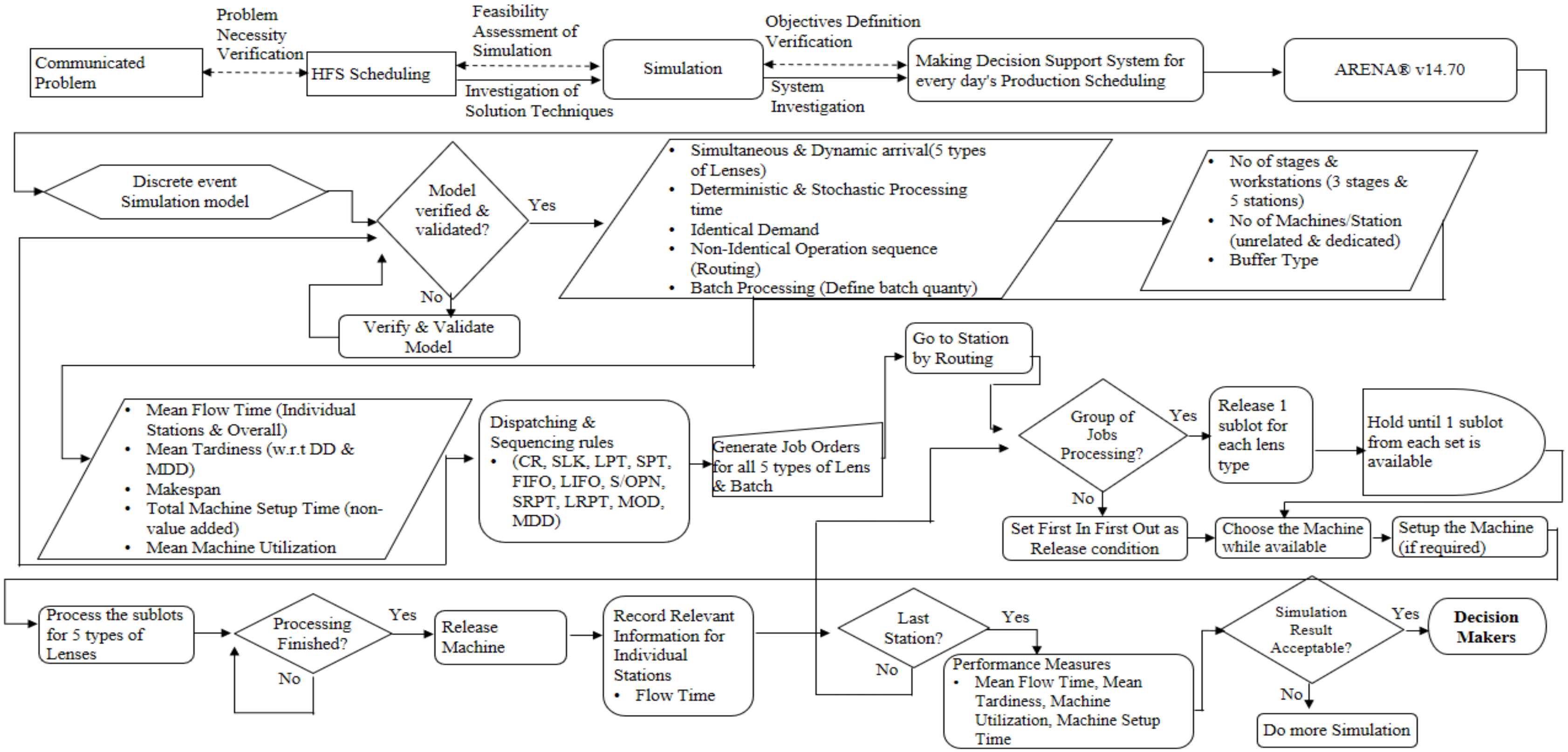
[Figure A.4] BS MC



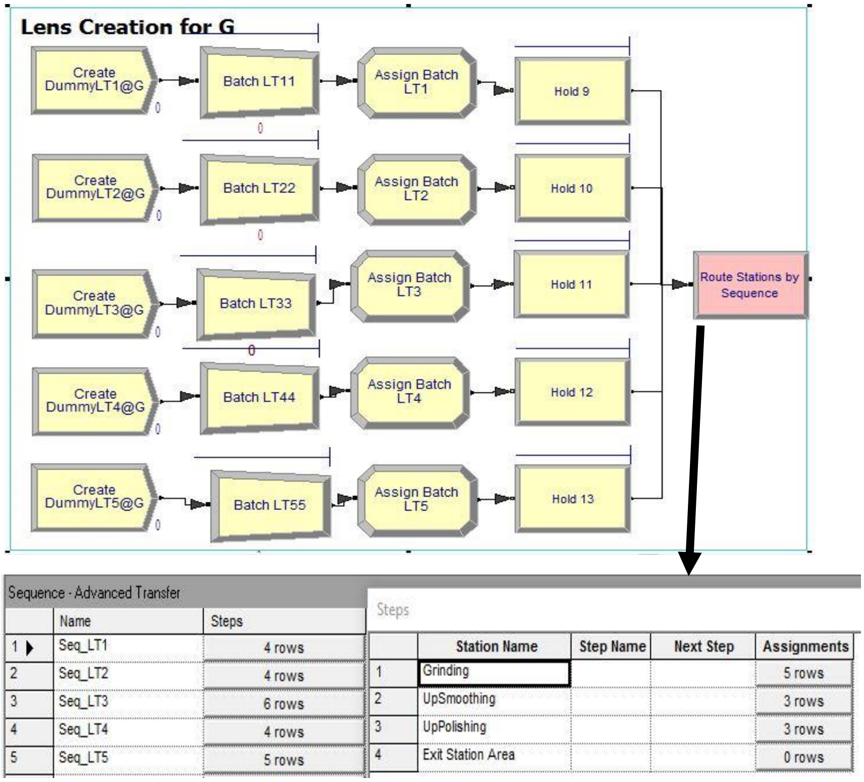
[Figure A.5] UP MC

[Figure A.6] BP MC

APPENDIX B



[Figure B.1] Basic flowchart for the developed model of HFS



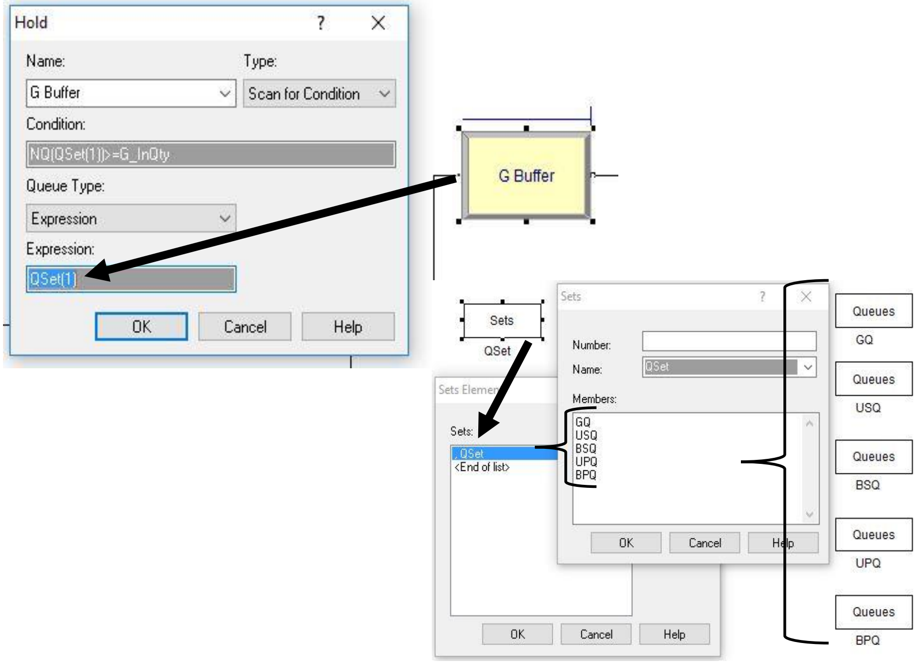
Sequence - Advanced Transfer

	Name	Steps
1	Seq_LT1	4 rows
2	Seq_LT2	4 rows
3	Seq_LT3	6 rows
4	Seq_LT4	4 rows
5	Seq_LT5	5 rows

Steps

	Station Name	Step Name	Next Step	Assignments
1	Grinding			5 rows
2	UpSmoothing			3 rows
3	UpPolishing			3 rows
4	Exit Station Area			0 rows

[Figure B.2] Lens creation and routing model



Name: G Buffer

Type: Scan for Condition

Condition: NQ(QSet(1))>=G_InQty

Queue Type: Expression

Expression: QSet(1)

Sets Elements

Sets:
QSet
<End of list>

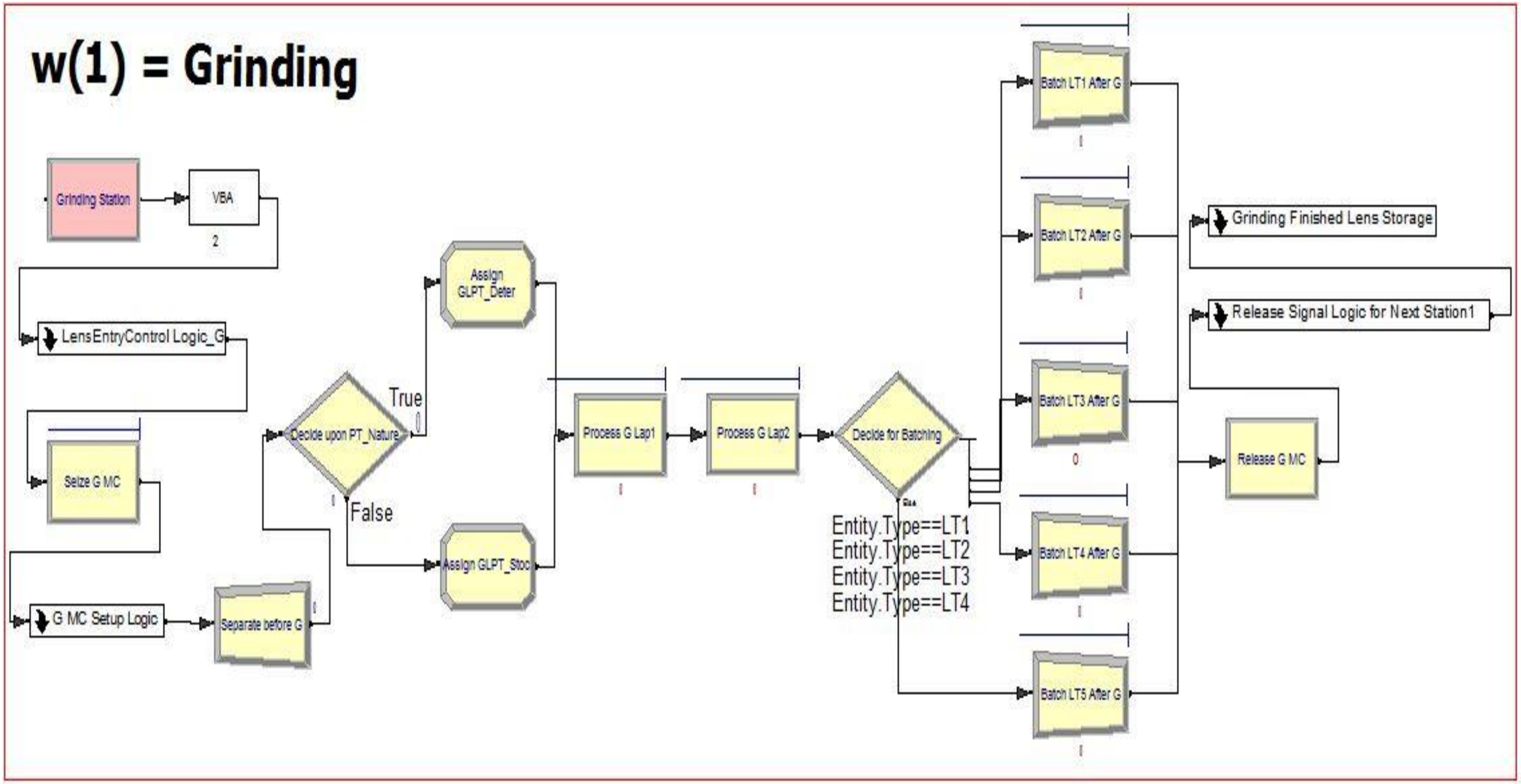
Sets

Number:

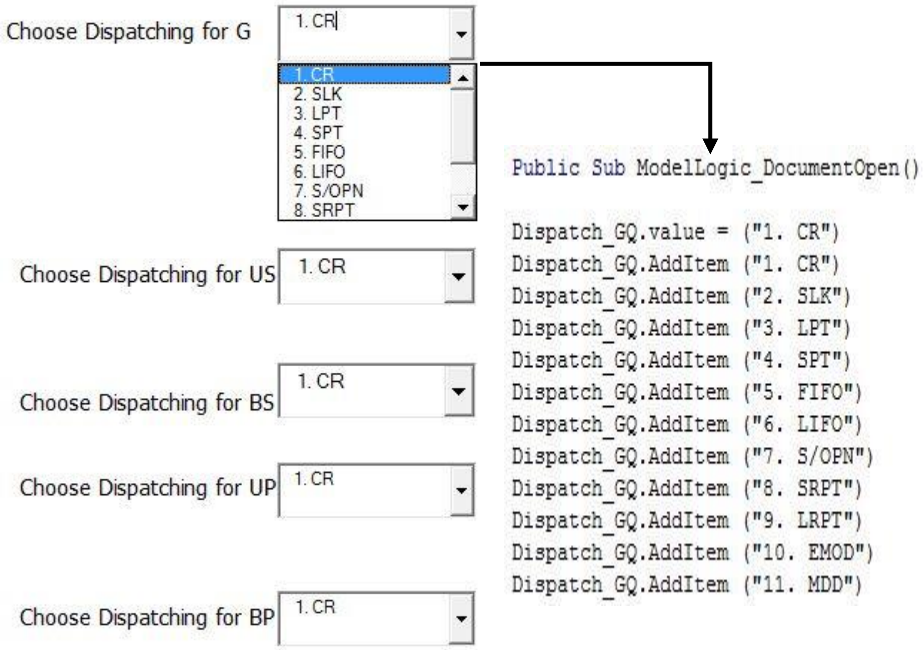
Name: QSet

Members:
GQ
USQ
BSQ
UPQ
BPQ

[Figure B.3] Queue model integration



[Figure B.4] Sample workstation (*w=1=Grinding*) design model



```
Public Sub ModelLogic_DocumentOpen()

Dispatch_GQ.value = ("1. CR")
Dispatch_GQ.AddItem ("1. CR")
Dispatch_GQ.AddItem ("2. SLK")
Dispatch_GQ.AddItem ("3. LPT")
Dispatch_GQ.AddItem ("4. SPT")
Dispatch_GQ.AddItem ("5. FIFO")
Dispatch_GQ.AddItem ("6. LIFO")
Dispatch_GQ.AddItem ("7. S/OPN")
Dispatch_GQ.AddItem ("8. SRPT")
Dispatch_GQ.AddItem ("9. LRPT")
Dispatch_GQ.AddItem ("10. EMOD")
Dispatch_GQ.AddItem ("11. MDD")
```

[Figure B.5] Interface integration for dispatching policies

APPENDIX C

```
Private Sub Dispatch_GQ_Change()
        Dim mObj, mObj1 As Module
        Dim i, i1 As Integer
        Set MyModel = ThisDocument.Model
        i = MyModel.Modules.Find(smFindTag, "GBufferQ")

        If i > 0 Then
                Set mObj = MyModel.Modules.Item(i)
        Else
                ' If the module was not found, display a message and exit
                MsgBox "Did not find module with tag 'GBufferQ'."
                Exit Sub
        End If

        If Dispatch_GQ.value = ("1. CR") Then
        With mObj
                .Data("Ranking") = "HVF"
                .Data("RankExp") = "(ReProcT/(DD-TNOW))"
        End With
  '     New Rule
        ElseIf Dispatch_GQ.value = ("2. SLK") Then
        With mObj
                .Data("Ranking") = "LVF"
                .Data("RankExp") = "(DD-TNOW-ReProcT)"
                End With
  '     New Rule
        ElseIf Dispatch_GQ.value = ("3. LCT") Then
        With mObj
                .Data("Ranking") = "HVF"
                .Data("RankExp") = "(ProcT)"
                End With
  '     New Rule
        ElseIf Dispatch_GQ.value = ("4. SCT") Then
        With mObj
                .Data("Ranking") = "LVF"
                .Data("RankExp") = "(ProcT)"
                End With
'     New Rule
        ElseIf Dispatch_GQ.value = ("5. FIFO") Then
        With mObj
                .Data("Ranking") = "FIFO"
                .Data("RankExp") = ""
                End With
```

```
'      New Rule
        ElseIf Dispatch_GQ.value = ("6. LIFO") Then
        With mObj
              .Data("Ranking") = "LIFO"
              .Data("RankExp") = ""
              End With
  '     New Rule
        ElseIf Dispatch_GQ.value = ("7. S/OPN") Then
        With mObj
              .Data("Ranking") = "LVF"
              .Data("RankExp") = "(DD-TNOW-ReProcT)/ReProcO"
              End With
                '     New Rule
        ElseIf Dispatch_GQ.value = ("8. SRCT") Then
        With mObj
              .Data("Ranking") = "LVF"
              .Data("RankExp") = "(ReProcT)"
              End With
    '     New Rule
        ElseIf Dispatch_GQ.value = ("9. LRCT") Then
        With mObj
              .Data("Ranking") = "HVF"
              .Data("RankExp") = "(ReProcT)"
              End With
     '     New Rule
        ElseIf Dispatch_GQ.value = ("10. EMOD") Then
        With mObj
              .Data("Ranking") = "LVF"
              .Data("RankExp") = "(EMOD)"
              End With
                   '     New Rule

End If
End Sub
```

초록

오늘날 생산 라인은 수 많은 생산 공정이 존재하여 복잡한 레이아웃으로 구성되어 있다. 아직까지 hybrid flow shop (HFS)에 대한 명확한 정의가 내려져 있지 않지만, 일반적으로 여러 병렬 기계로 구성되어 있는 생산 환경을 HFS라고 한다. HFS는 생산환경에 따라 특성이 다 다르기에 복잡한 형태를 지니며, 다양한 목적함수들 간의 tradeoff가 발생할 수 있는 가능성 때문에 의사결정이 쉽지 않다. 그러나 지금까지의 HFS 관련 연구들은 단일 기준 목적에 집중하고 있어 HFS의 복잡도와 중요성을 감안하여 볼 때, 현실과 맞지 않는다는 한계점을 지니고 있다.

현재 산업계에서는 시뮬레이션을 통해 생산시스템을 모델링하고, 생산시스템의 특성 변화에 따른 성과의 변동을 예측하고 그에 대처할 수 있는 방안을 마련하고 있다. HFS 스케줄링 문제는 생산 공정에 관여하는 모든 요소들을 이용하여 해답을 찾는 문제이기 때문에, 시뮬레이션 유연성은 HFS 문제를 푸는 데에 있어 최적의 솔루션을 제공하는 도구로 사용될 수 있다. 그러나 HFS 문제를

시뮬레이션을 이용하여 풀기 위해서는 생산 공정에 관여하는 모든 직원들의 높은 시뮬레이션 숙련도가 요구된다. 이러한 문제는 개개인의 요구에 맞춘 인터페이스 설계와 시뮬레이션 지원 프로그램과의 통합을 통해 해결될 수 있기 때문에 본 연구에서는 HFS 문제를 해결하기 위해 유연한 시뮬레이션 모델링 프레임워크를 제안하고자 한다. 또한 시뮬레이션을 통해 일반적으로 통용되고 있는 작업 순서와 디스패칭 정책의 영향도를 여러 가지의 성과 측정치를 이용하여 파악하고자 한다. 결과적으로 생산 공정에서의 효율적인 디스패칭 정책을 제시하고, HFS를 해결하기 위한 다양한 배치 사이즈를 도출하고자 한다.

주요어: Hyrbid flow shop (HFS), 시뮬레이션, Dispatching rules, 다양한 베치 크기

학번: 2015-23296