2021

BALANCING TRADE-OFFS IN ONE-STAGE PRODUCTION WITH PROCESSING TIME UNCERTAINTY

Jingjing You
University of Kentucky, jingjing_you@yahoo.com
Author ORCID Identifier:
https://orcid.org/0000-0001-7062-0983
Digital Object Identifier: https://doi.org/10.13023/etd.2021.127

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Jingjing You, Student

Dr. Wei Li, Major Professor

Dr. Alexandre Martin, Director of Graduate Studies

BALANCING TRADE-OFFS IN ONE-STAGE PRODUCTION WITH PROCESSING TIME UNCERTAINTY

THESIS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering in the College of Engineering at the University of Kentucky

By

Jingjing You

Lexington, Kentucky

Director: Dr. Wei Li, Professor of Mechanical Engineering

Lexington, Kentucky

2021

https://orcid.org/0000-0001-7062-0983

ABSTRACT OF THESIS

BALANCING TRADE-OFFS IN ONE-STAGE PRODUCTION WITH PROCESSING TIME UNCERTAINTY

Stochastic production scheduling faces three challenges, first the inconsistencies among key performance indicators (KPIs), second the trade-offs between the expected return and the risk for a portfolio of KPIs, and third the uncertainty in processing times. Based on two inconsistent KPIs of total completion time (*TCT*) and variance of completion times (*VCT*), we propose our trade-off balancing (ToB) heuristic for one-stage production scheduling. Through comprehensive case studies, we show that our ToB heuristic with preference α=0.0:0.1:1.0 efficiently and effectively addresses the three challenges. Moreover, our trade-off balancing scheme can be generalized to balance a number of inconsistent KPIs more than two. Daniels and Kouvelis (DK) proposed a scheme to optimize the worst-case scenario for stochastic production scheduling and proposed the endpoint product (EP) and endpoint sum (ES) heuristics to hedge against processing time uncertainty. Using 5 levels of coefficients of variation (*CV*s) to represent processing time uncertainty, we show that our ToB heuristic is robust as well, and even outperforms the EP and ES heuristics on worst-case scenarios at high levels of processing time uncertainty. Moreover, our ToB heuristic generates undominated solution spaces of KPIs, which not only provides a solid base to set up specification limits for statistical process control (SPC) but also facilitates the application of modern portfolio theory and SPC techniques in the industry.

KEYWORDS: Key performance indicators, Modern portfolio theory, Statistical process control, Trade-off Balancing, Heuristics.

Jingjing You

04/12/2021
Date

BALANCING TRADE-OFFS IN ONE-STAGE PRODUCTION WITH PROCESSING TIME UNCERTAINTY

By

Jingjing You

Dr. Wei Li

Director of Thesis

Dr. Alexandre Martin

Director of Graduate Studies

04/12/2021

Date

ACKNOWLEDGMENTS

I would like to start by giving my deepest gratitude to my academic advisor, Dr. Wei Li, for taking me on as his student and guiding me to build the ability of critical thinking. He always inspires me to raise questions and instructs me to explore the solutions gradually. Whenever I am perplexed by a problem, he always provides me constructive comments promptly. Moreover, he gave me timely guidance when the progress of my thesis was slow so that I can finish the program I once thought was unattainable. I also want to thank Dr. Jawahir and Dr. Badurdeen for their guidance in and out of class.

In addition to the technical and instrumental assistance above, I received strong support from my family. I am truly grateful to my parents, my brother, father-in-law, mother-in-law, and my husband for their ongoing encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Background

Flow shop scheduling is critical in manufacturing because it affects the performance of the whole production process. A flow shop is defined as n jobs processed sequentially on m machines, in which each job has a specified processing time on each machine and each machine can process one job at a time. The basic characteristic of permutation flow shop scheduling is that the job order is the same on each machine, so determining a sequence achieving one or several objectives is the key point in flow shop scheduling. Flow shop scheduling can be regarded as a special case of job shop scheduling. In a job shop, machines are not ordered sequentially, and the job order can be different on each machine. In addition to manufacturing, flow shop scheduling can be widely applied to other fields, such as transportation, medical and health care, supply chain, etc., in which operations are sequentially carried out in achieving an objective. The 4th industrial revolution has higher requirements for scheduling because of the following two reasons. The first reason is based on the limitation of resources. Initially, in the first stage of the industrial revolution, the requirement on scheduling was not urgent, as resources were relatively sufficient for simple and standard tasks. In recent years, the emergence of mass production systems for highly customized products increases the production volume, requires more resources, and intensifies the need for scheduling. More researchers recognize that it is necessary to do an in-depth and extensive research on scheduling to achieve certain production objectives with limited production resources. Generally, the goal of production scheduling is to allocate competing tasks to scarce resources over time, in achieving some objectives (Pinedo, 2012). In flow shop scheduling, we have some classic objectives to achieve, such as minimize maximum completion time, min(*MCT*), minimize total completion time, min(*TCT*), and minimize idle times or setup times, etc. The second is based on trade-offs in multi-objective optimization. Although Garey et al. (1976) proved that flow shop scheduling to min(*MCT*) is a nondeterministic polynomial complete (*NP*-complete) problem, a substantial volume of research papers were published in the literature about scheduling, and various heuristic algorithms and exact algorithms were proposed in dealing with different production issues. One of the objectives to develop scheduling methods is to reduce the computation time or the complexity, but few address the trade-offs as in this thesis for robust production scheduling.

Scheduling for one-stage production is important from the following three perspectives. First, from the perspective of multi-objective optimization, we can draw tight and solid bounds for individual variables based on solutions to one-stage production problems. In analyzing individual objectives involved in multi-objective optimization, we need the solution space to individual objectives, which can be used to set up the solid upper and lower limits of individual objectives. These bounds make it more efficient or effective to solve multi-objective optimization problems. This is more pronounced when some optimization problems for multi-stage flow shop scheduling are *NP*-complete, that is, the optimal solution cannot be obtained. Second, from the perspective of interpretation, we can explain the solutions both quantitatively and qualitatively, making it easy to deal with multi-stage production problems. Quantitatively, a factory or an entire assembly line can be modeled as a one-stage production, for which the due date is set up for delivery and can be decomposed as completion times for substages. Qualitatively, we can define the stability of a stage as the variation range of total completion time in a certain stable range.

Third, from the perspective of computation speed, we can get good solutions in time, making it possible to control the whole process in real-time. One-stage production scheduling is meaningful, not only to address the production requirement for the whole manufacturing process but also to address the complexity in a bottleneck operation. Therefore, our research on one-stage production provides insights into both scale-up and scale-down production scheduling problems.

1.2 Challenges and Motivations

The following three challenges motivate us on one-stage production scheduling. The first challenge is trade-offs exist among inconsistent key performance indicators (KPIs) for production. For example, minimizing the total completion time (*TCT*) and minimizing the variance of completion times (*VCT*) are inconsistent with each other. The second one is trade-offs exist between minimizing the mean and minimizing the standard deviation for a quadratic optimization problem. In terms of modern portfolio theory (Markowitz, 1952), optimizing the expected return of a portfolio is inconsistent with minimizing the risk. The third one is uncertainties exist in the real production environment, such as stochastic processing times, that is being with variation in processing times. Given stochastic processing times, Daniels and Kouvelis (1995) proposed their heuristics to maximize the minimum deviation from the upper bound, that is, optimization for the worst-case scenario. However, optimizing the worst-case scenario does not necessarily optimize the average performance.

1.3 Contribution

We have made contributions in this thesis work from the following three aspects. First, we develop a trade-off balancing (ToB) heuristic for one-stage production scheduling. Although the current version of our ToB heuristic balances trade-offs between min (*TCT*) and min (*VCT*), the sequencing scheme of our ToB heuristic can be extended to balance trade-offs among multiple KPIs. Second, for robust scheduling with stochastic processing times, we show that balancing the trade-offs between the mean and the variance of an objective function is more appropriate than optimizing the worst-case scenario. Third, integrating our ToB heuristic with statistical process control (SPC) techniques, our approach of trade-off balancing can provide solid specification limits of individual KPIs for production control. Through case studies, we verify that our ToB heuristic generates stable production performance for robust production control against stochastic processing times.

The rest of this thesis is organized as follows. In Chapter Two, we provide a thorough literature review on flow shop scheduling for single- and multi-objective optimization, and the necessary information on modern portfolio theory, stochastic control theory, and statistical process control. In Chapter Three, we introduce our methodology for developing our ToB heuristic. In Chapter Four, we provide and analyze the results of empirical case studies. In Chapter Five, we draw conclusions and future work.

CHAPTER 2. LITERATURE REVIEW

To better understand the three challenges described in Chapter One, we provide a literature review in Chapter Two, organized as follows. In Section 2.1, we introduce the definition of production scheduling and discuss the differences between flow shops and job shops, including one-stage and multi-stage production, based on which readers can not only distinguish different types of production lines, but also apply our methodology to suitable process settings. In Section 2.2, we discuss heuristics for multi- and one-stage flow shop scheduling, where the one-stage flow shop scheduling will affect the solution space of the multi-stage flow shop scheduling problems. In Section 2.3, we introduce multi-objective optimization and relative advantages and disadvantages. In Section 2.4, we introduce the modern portfolio theory (MPT) and its application to our research. In Section 2.5, we introduce the basic concept of stochastic control, based on which we further discuss the advantage and the disadvantage in Daniels and Kouvelis (1995)'s scheme for robust scheduling. Finally, in Section 2.6, we provide mathematical definitions of the statistical process control (SPC) and discuss how some process capability indices can be applied to trade-off balancing.

2.1 Scheduling

Scheduling is to allocate the competing jobs to limited resources in achieving some objectives (Pinedo, 2016). The objectives related to manufacturing can be minimizing makespan, minimizing production cost, maximizing equipment utilization, minimizing idle times, and those related to customers' satisfaction can be minimizing the delivery tardiness and minimizing transportation cost. Scheduling has a wide range of applications in many fields, such as enterprise management, transportation, aerospace, medical and health care, modern flexible manufacturing systems, and so on. As processes differ on process settings, job characteristics, and evaluation criteria, R.L. Graham et al. (1979) classified scheduling problems by using a triplet scheme of $\alpha|\beta|\gamma$, where α denotes the process settings, β denotes the job characteristics, and γ specifies the evaluation criterion. Suppose that *n* jobs J_j(*j*=1, …, *n*) need to be processed on m machines M_i(*i*=1, …, *m*), and p_{ij} denotes the processing time of job *j* on machine *i*.

For process settings or the machining environment, there are three categories:

i. single machine or multi-machine ($\alpha = 1$ or m).

ii. parallel machines: identical ($\alpha =$ P), uniform ($\alpha =$ Q), unrelated ($\alpha =$ R).

iii. multi-operation models: Flow Shop ($\alpha =$ F), Open Shop ($\alpha =$ O), Job Shop ($\alpha =$ J), Mixed (or Grouped) Shop ($\alpha =$ X).

For job characteristics, there are six categories:

i. Whether the preemption is allowed ($\beta_1 = pmtn$presence of preemption).

ii. The presence of resource constraints (s limited resources: $\beta_2 = s$, only a single resource: β_2 =1, no limited resources: $\beta_2 = 0$).

iii. The precedence relation (arbitrary: $\beta_3 = prec$, rooted tree: $\beta_3 = tree$, no precedence relation: $\beta_3 = 0$).

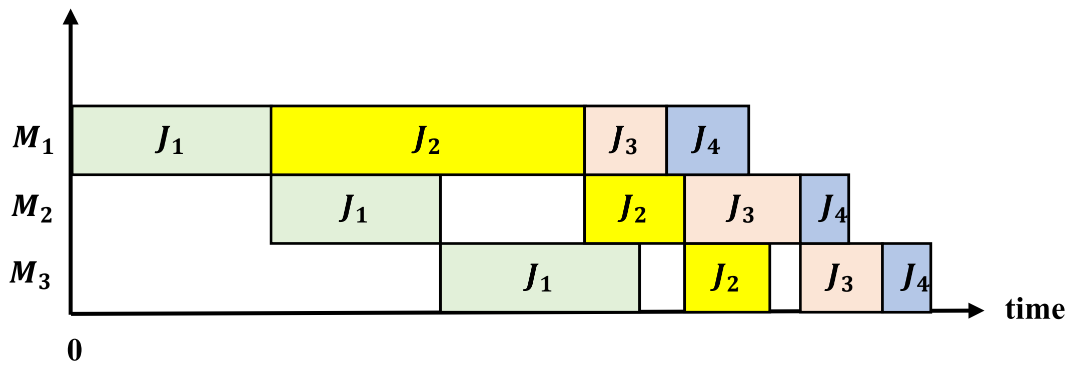
iv. $\beta_4 = r_j$, r_j presence of release dates. $\beta_4 = 0$: we assume that $r_j = 0$.

v. $\beta_5 = m_j \leq \overline{m}$: A constant upper bound on m_j is specified. $\beta_5 = 0$: No such bound is specified.

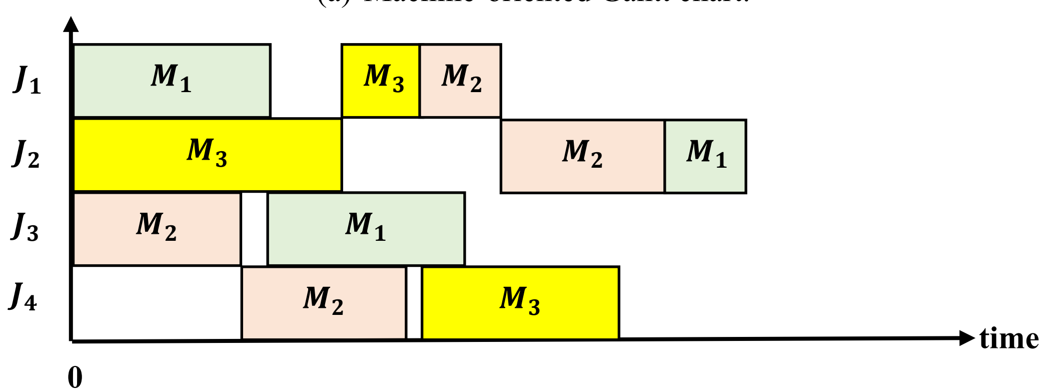
vi. $\beta_6 = p_{ij} = 1$: The processing times are unit. $\beta_6 = \underline{p} \leq p_{ij} \leq \overline{p}$: Constant lower and upper bounds on p_{ij} are specified. $\beta_6 = 0$: No such bounds are specified.

These categories of job characteristics help us classify the problem by showing specific job characteristics and the relationships between the job.

For evaluation criteria, Brucker (2006) proposed a method that is useful for presenting cost function. The author denoted γ as a cost function and the completion time of job J_j by C_j. Then Brucker (2006) associated the cost with $f_j(C_j)$. The total cost function was expressed by the following function: $f_{max}(C) \coloneqq \max\{f_j(C_j)|j = 1, \dots, n\}$ or $\sum f_j(C) \coloneqq \sum_{j=1}^{n} f_j(C_j)$ is called bottleneck objectives and sum objectives, respectively. Thus, we can model any scheduling problem as a cost function that aims to find a feasible sequence that minimizes the total cost function. The Gantt chart can show the operation process intuitively. We can present production scheduling in two ways as shown in Figure 1. Figure 1(a) is mainly for the machine-oriented perspective, where shows the operation on each machine. Figure 1(b) is mainly for the job-oriented perspective, where shows that all machines on which each job is operated.



(a) Machine-oriented Gantt chart.



(b) Job-oriented Gantt chart.

Figure 1: Machine-oriented and Job-oriented Gantt charts.

Job shop scheduling is one type of scheduling and flow shop scheduling can be regarded as a special case of job shop scheduling. As the most common and complex scheduling problem in the real production environment, job shop scheduling can be described as n jobs with different processing times to be scheduled on m machines with different process routings. The character of a job shop is jobs need to be processed in a specific order aiming to achieve a single objective or multiple objectives and each job needs to separately occupy one machine for processing. Once the process begins, it cannot be interrupted before the last job being finished on the last machine and each machine can only process one job at any time. In a job shop, the job order can be the same or different on each machine. The advantages of a job shop are as follows. Setting up with one or two machines initially is not difficult, and not difficult to add, change or remove machines as necessary. However, because of this flexibility, it is difficult to automate and schedule due to the lack of consistency and standardization. The difference between a job shop and a flow shop is that in a flow shop, the job order is the same on each machine. If some jobs are not processed on some machines, the corresponding processing time should be zero. The advantages of a flow shop are it is easy to automate, measure and optimize since the job order is the same on each machine. Compared with job shops, the disadvantage of flow shops is that they have less flexibility. Moreover, they need more initial work to set up a flow shop because determining a sequence achieving one or several objectives is the key point of the flow shop scheduling problem. Therefore, mathematical algorithms are applied to generate schedules and improve the effectiveness and efficiency of flow shop scheduling problems. Depending on the advantage and the disadvantage of job shops and flow shops, the designer can use either one type or combination of both types to design the layout of the operating environment.

2.2 Heuristics for multi-and one-stage flow shop scheduling

It has been almost 70 years since the pioneer paper published by Johnson in 1954 on flow shop scheduling, whose algorithm is for 2-machine production with an objective to minimize makespan. In the beginning of flow shop production scheduling, researchers mainly used branch & bound techniques (Gupta and Stafford, 2006). However, for a production line with $m \geq 3$ machines, the flow shop scheduling problem is *NP*-complete (Garey et al., 1976). Although it is difficult to find an optimal solution for *NP*-hard problems, many researchers put time and effort into developing scheduling theory and heuristics. Most of the literature initially focused on minimizing the maximum completion time (*MCT*). NEH heuristic (Nawaz et al., 1983) is the best constructive heuristic to achieve the optimal solution for *MCT* minimization. Many researchers generated heuristics and algorithms to achieve other objectives depending on the scheme of the NEH heuristic. Gradually, with the complexity of actual production, algorithms for optimizing other objectives were generated. For example, the heuristic algorithm to minimize total flowtime (Rajendran, 1993, and Liu and Reeves, 2001) and the heuristic algorithm to minimize completion time variance (Kubiak, 1993). In this section, we provide literature reviews on multi- and one-stage flow shop scheduling.

2.2.1 Heuristics for multi-stage flow shop scheduling

A flow shop is a workshop that n jobs are processed in the same order on m machines (Pinedo, 2016). It is generally required that the flow direction of the job is consistent rather than that each job must be processed on each machine. If some jobs are not processed by some machines, the corresponding processing time should be set to zero. The most common performance measurement is makespan (maximum completion time) minimization and total completion time (*TCT*) minimization. Given the processing time of job j on machine i, i.e., p_{ij}, we denote C_j as the completion time of job j on the last machine m. Therefore, the maximum completion time (C_{max}) is the completion time of the last job n on the last machine m, and the total completion time ($\sum C_j = \sum_{j=1}^{N} C_{j,m}$) is the sum of completion times of all jobs on the last machine. Except for the above two basic performance measurements, there are several other classical performance metrics. For example, the measurement based on the due date will be the lateness (L_j), the tardiness (T_j), and the unit penalty (U_j) (Pinedo, 2016). They are defined as

$$L_j = C_j - d_j, \tag{1}$$

$$T_j = \max(L_j, 0), \text{ and} \tag{2}$$

$$U_j = \begin{cases} 1, if\ C_j > d_j \\ 0, otherwise \end{cases} \tag{3}$$

Garey et al. (1976) and Hoogenveen and Kawaguchi (1999) proved that MCT minimization and *TCT* minimization are *NP*-complete for a production problem with m>2 machines. However, the effort on seeking near-optimal solutions never stops. NEH heuristic, LR heuristic, and FF heuristic are three classic heuristics for multi-stage flow shop scheduling problem optimization. NEH heuristic (Nawaz et al., 1983) is the best constructive heuristic to achieve the optimal solution for *MCT* minimization. LR heuristic (Liu and Reeves, 2001)) and FF heuristic (Fernandez-Viagas and Framinan, 2015) are considered as two of the best constructive heuristic to achieve the optimal solution for *TCT* minimization (Li et al., 2019). We will review some other heuristic algorithms for multi-stage flow shop scheduling with different problem sets or perspectives.

Campbell et al. (1970) developed the CDS heuristic based on Johnson's rule. First, the CDS algorithm separated *m* machines into two groups, that is one group had (*m*-1) machines and the other had one machine, and then applied Johnson's rule to the group with (*m*-1) machines to find the target sequence with minimum makespan.

Woo and Yim (1998) proposed a heuristic based on the job insertion strategy of the NEH heuristic (Nawaz et al., 1983) to solve the problem with 5, 10, 15, 20 machines. Although it showed that the heuristic needed more computation time than other heuristics, it outperformed CDS (Campbell et al., 1970), NEH (Nawaz et al., 1983), and Rajendran's algorithm (Rajendran, 1993).

Rajendran and Chaudhuri (1990) proposed two heuristic algorithms for the flow shop problem to minimize *TCT*. The experiment was designed with multi-stage varying from 3 to 25. The procedures of the first heuristic are shown as follows:

Step.1: Let σ be the available partial schedule, π be the set of unscheduled jobs, and n' be the number of jobs in the set σ. Arranging the job in ascending order of value $T_j = \sum_{j=1}^{m}(m - j + 1)p_{ij}$, where p_{ij} is the processing time of job j on machine i. If there exists a tie ranking, put the job with the least value of $T_j' = \sum_{j=1}^{m} p_{ij}$ first.

Step.2: Select the first job in the array and put it into the σ, thus n'=1, then update the array. Select the second job in the array and put it in the pth position of σ according to $[n'+1]/2 \leq p \leq [n'+1]$. Sequence the job in ascending order of value $\sum_{j=2}^{n'+1}(n' + 2 - j)d_{[j-1],[j]}$ in the σ, where $d_{[j-1],[j]}$ is the minimum delay on the first machine between the start of job [*j*-1] and job [*j*].

Step.3: Continue to step 2 and update array and σ until there is no job in π.

Another heuristic is similar with the one mentioned above but ranks the sequence according to the value of T_j'.

Compare to Bonney and Gundry (1976), King and Spachis (1980), and the RANDOM selection rule, these two heuristics gave optimal or near-optimal solutions.

Followed Rajendran and Chaudhuri (1990), Rajendran and Chaudhuri (1992) proposed three heuristic algorithms that achieve the following three objectives, respectively.

i. $\sum_{j=2}^{m} \max[q(\sigma a, j - 1) - q(\sigma, j), 0]$,

ii. $\sum_{j=2}^{m} \text{abs}[q(\sigma a, j-1) - q(\sigma, j)]$,

iii. $\sum_{j=2}^{m} \text{abs}[q(\sigma a, j-1) - q(\sigma, j)] + \sum_{j=1}^{m} q(\sigma a, j)$.

According to two phases of experimentation, the three heuristic algorithms proved better performance than Gupta's MINIT algorithm (Gupta, 1972), Miyazaki's algorithm (Miyazaki, 1978), and Ho and Chang's algorithm (Ho and Chang, 1991).

Bertolissi (2000) Proposed a heuristic using a comparison algorithm and insertion algorithm for the flow time minimization objective. The comparison algorithm is as follows: The job sequence is generated by comparing each pair of temporary flow times. First, the initial pair of jobs is set as the pair that has the smallest flow times. Then, doing the same operation on the rest pairs of flow times. In the meantime, marking the starting job of the pair so that each job has its number of marks. Second, ranging all jobs in decreasing number of its marks. If there exists a tie ranking, sequencing the job in nonincreasing of total processing time. The insertion algorithm used to improve the performance of the initial sequence is the same as the insertion method of Rajendran and Chaudhuri's heuristics (Rajendran and Chaudhuri, 1990). According to the results of experiments, this heuristic had an identical computational time as RC heuristic (Rajendran and Chaudhuri, 1990) and BG heuristic (Bonney and Gundry, 1976), but was better than RC heuristic (Rajendran and Chaudhuri, 1990) and BG heuristic (Bonney and Gundry, 1976). Furthermore, the quality of the schedule was improved effectively.

In analyzing the NEH heuristic, Framinan et al. (2003) proposed a heuristic aiming to minimize flowtime i.e., *TCT*, and showed that their special modifications of NEH heuristic could improve the performance of the sequence not only on the quality of the approximation but also on the application range from small-scale to large-scale. The essence of the heuristic was the pairwise exchange scheme.

Aldowaisan and Allahverdi (2004) generated six heuristics for the flow time minimization objective. The six heuristics (PH1, PH2, PH3, PH4, *i*PH*i*, PH*i*(*p*)) depended on three schedule criteria:

i. on the criteria that choose between two stoppages. PH1 and PH2 terminate after 10 replications while PH3 and PH4 terminate either after 10 replications or a worse solution appeared.

ii. on the criteria that choose between two insertion approaches. PH1 and PH3 used Nawaz et al. (1983) insertion method while PH2 and PH4 used Rajendran and Ziegler's insertion method (Rajendran and Ziegler, 1997).

iii. on the criteria that exchange the adjacent pairwise procedure. *i*PH*i* and PH*i*(p) applied the criteria for further improvement.

PH1(*p*) was recommended because it outperformed two heuristics proposed by Rajendran and Chaudhuri (1990) and the genetic algorithm generated by Chen et al. (1996).

As discussed above, many heuristic algorithms were based on or modified some classical heuristics. For example, the CDS algorithm regrouped m machines into two subgroups. Such a strategy gives us a profound insight into the importance of returning the one-stage flow shop scheduling problem. Modeling a bottleneck as a one-stage or split multi-stage production line into several production lines helps us reduce the complexity of the problem. The solution precision of a one-stage problem will affect the solution space of a multi-stage problem. Therefore, it is important to improve the efficiency and effectiveness of heuristics for one-stage production problems. In section 2.2.2, we review some literature on one-stage flow shop scheduling.

2.2.2 Heuristics for one-stage flow shop scheduling

In a one-stage production environment, given *N* jobs for processing, we omit the machine index, the processing time of job j on the single machine will be p_j (Gupta and Kyparisis, 1987). We can generate *N* factorial (*N*!) possible sequences for one-stage production scheduling. For deterministic *N*-job 1-stage production problems,

$$MCT = C_{max} = C_N = \sum_{j=1}^{N} p_j \quad (4)$$

is a constant, and there is no difference in *N!* possible sequences. But the total completion time,

$$TCT = \sum C_j = \sum_{j=1}^{N} C_j = \sum_{j=1}^{N} (N - j + 1) \cdot p_j \quad (5)$$

is a weighted sum of processing times and different weights will present different performances. Li et al. (2014) proved that the shortest processing time (SPT) rule, arranging p_j in nondecreasing order, generates an optimal solution to min (*TCT*) for deterministic problems. The mean flow time equals TCT/N which drives other KPIs in scheduling, such as work-in-process inventories and the mean waiting time in the process, etc. Because the job sequence is independent on processing times for some one-stage scheduling problems, such as maximum lateness minimization and maximum tardiness minimization, the optimal solution to min (L_{max}) and min (T_{max}) is to order the jobs in a nondecreasing of due dates (Shabtay and Steiner, 2007).

Customer satisfaction is directly affected by the quality of service received, so minimizing the variance of performance and giving a uniform response to customer's requests are usually desirable. Such measurements have a strong correlation with customer satisfaction (Merten and Muller, 1972). To improve the quality of products and services, the company is also pursuing to provide the same service for customers. Minimizing the variance of completion times (*VCT*) is one type of minimizing the variance of performance, defined as the variance among completion times of *N* jobs. Generally, min (*VCT*) is *NP*-hard. The following research findings are useful on min (*VCT*).

Merten and Muller (1972) analyzed the variance of flow time and variance of waiting time on a single machine production problem. Firstly, they analyzed the minimization of mean flow time and the mean waiting time in the single machine production environment. The analysis process is as follows:

Given *n* independent jobs are to be processed one at a time in sequence on a single machine. The number of possible permutations for sequencing the jobs is *n*!. Let *R* = (i_1, i_2, …, i_n) be that element of Π which is the set of all permutations of the first *n* integers where integer *i* is in the *j*th position for *j* = 1, 2, …, *n* and R' = (i_n, i_{n-1},…, i_2, i_1). And let p_i be the processing time for each job *i* and *u*(*i*) be the weight for describing the relative importance of job *i*. The reason why they reverse *R* to get the antithetical schedule, i.e., R' is to check whether these two schedules result in a minimum mean flow time (*FM*) and mean waiting time (*WM*) and maximum *FM* and *WM*, respectively. The *FM* and the *WM* are expressed in the formulas, $FM(R) = \sum_{j=1}^{n} u(i_\mathrm{j})F(R, i_\mathrm{j})$ and $WM(R) = \sum_{j=1}^{n} u(i_j)W(R, i_j)$. Given the properties of the sequence that minimizing *FM* and the sequence that minimizing *WM*, Merten and Muller (1972) found that the optimal solution to the *FM* and the *WM* can be achieved by the same job schedule, that is the schedule $R_{solution}$ gave minimums of *FM* and *WM* while $R'_{solution}$ gave maximums of *FM* and *WM*.

The introduction of performance measures of *FM* and *WM* was aiming to contrast to the performance measures of the variance of flow time (*FV*) and the variance of waiting time (*WV*) expressed as follows, $FV(R) = \sum_{j=1}^{n} u(i_j)(F(R, i_j) - FM(R))^2$ and $WV(R) = \sum_{j=1}^{n} u(i_j)(W(R, i_j) - WM(R))^2$. Merten and Muller (1972) proved that the sequence that minimizes the *FV* is antithetical to the sequence that minimizes the *WV*, although the minimum values of the two variance measures are equal.

Eilon and Chowdhury (1977) focused on the waiting time variance minimization problem in the single machine. They proved that the optimal sequence should be *V*-shaped and an algorithm was given accordingly. To improve the performance of the algorithm, a heuristic method was developed especially for the scenario with several jobs. Their work has inspired many researchers to study the variance of completion times or related fields.

Kanet (1981) also modeled this type of problem as minimizing the variation in flow time, i.e., min (*VCT*), aiming to reduce the fluctuation of the treatment of jobs (customers) such as the variation of service time (time in the system) and the variation of waiting time for service (time before operation) of each job. The author found an alternative way to min (*VCT*), which is equivalent to measure the total absolute differences in completion times (*TADC*): $TADC = \sum_{i=1}^{n} \sum_{j=i}^{n} \mid C_j - C_i \mid = \sum_{j=1}^{n} (j-1)(n-j+1) \cdot p_j$, where the weight is $-j^2 + nj + 2j - n - 1$, which independent on processing times. Clearly, the weight is a quadratic function with a maximum value at $j = \frac{(n+2)}{2}$. Assume that a single stage with *n* jobs available at time zero for production, obviously, given the characteristic above, the optimal sequence has three properties:

i. the job with the maximum value of processing time should be scheduled first.

ii. the sequence is *V*-shaped regarding processing times.

iii. Let *k* is the position in the schedule of the job with the smallest processing time, if *n* is even, *k*=(*n*+2)/2, if *n* is odd, *k*=(*n*+1)/2.

To achieve the optimal *TADA* schedule, Kanet (1981) generated two methods. The first one is the *GEN* method: Set *S* be the final sequence. Arrange all jobs in descending order of processing times. Consider the sequence as *AS*. Assign the first in *AS* to be last, the second in *AS* to be first, the third to be last but one, the fourth to be second, and so on, until assigning all jobs so that generate *S*. The second one is the *SMV* method: Set *U* be the set of unscheduled jobs. Regard the smallest job in *S* as *k* and the largest job in *U* as *i*. Compute the variance of completion times of temporary sequence $\boldsymbol{S'}$ and $\boldsymbol{S''}$, where $\boldsymbol{S'}$ is generated by inserting job *i* to the immediate left of *k* and $\boldsymbol{S''}$ is generated by inserting job *i* to the immediate right of *k*. If the variance of completion times of temporary sequence $\boldsymbol{S'}$ is less than or equal to that of temporary sequence $\boldsymbol{S''}$, set $S = \boldsymbol{S'}$. Otherwise, $S = \boldsymbol{S''}$. Continue the process until all jobs are scheduled. According to 7 cases study, it was showed that *SMV* was a simple method to find the optimal solution and when the job number is less than five and outperformed the heuristic given by Eilon and Chowdhury (1977).

Schrage (1975) presented four theorems and three corollaries for the single machine environment with a finite number of jobs. However, they were not suitable for the weighted time calculation method just except for the first corollary (Merten and Muller, 1972). The followings are the summary. Let a(*i*) be the index for $p_{a(1)\geq}p_{a(2)\geq}p_{a(3)\geq}\ldots$.

Theorem 1: the property of finite sequence that aims to achieve the optimal solution for minimizing the variation of completion times is scheduling the job with the largest processing time.

Theorem 2: Reversing the last *n*-1 jobs will not change the variance of the schedule.

Theorem 3: When the job number is larger than 3, to generate a sequence that minimizes the variance of completion times should have the properties $p_2 \geq p_3$ and $p_n \geq p_{n-1}$.

Theorem 4: When the job number is equal to 5, there are two solutions for minimizing the variance of completion times, i.e., a(1), a(2), a(5), a(4), a(3) and a(1), a(3), a(4), a(5), a(2).

Correspondingly, three corollaries are as following,

i. The optimal solution of minimizing the variance of completion times for two job problems is processing the longest job first.
ii. As long as processing the longest job first, the schedule will achieve a minimum value of the variance of completion times in the three jobs system.
iii. As long as processing the longest job first and the shortest job third, the schedule will achieve a minimum value of the variance of completion times in the four jobs system.

Bagchi (1989) thought that no efficient algorithm exists now for an optimal solution both for minimizing the variation of completion times and for minimizing the variation of waiting times. But the properties for this type of problem were summarized (the properties were also held by *TADC* and *TADW*):

i. The sequence that achieves an optimal solution for min (*VCT*) is antithetical to the sequence that achieves an optimal solution for the variance of waiting times (*VWT*) (Merten and Muller, 1972).
ii. The value of *VCT* of any sequence is the same as the value of *VWT* of the antithetical sequence (Merten and Muller, 1972).

iii. The value of *VWT* of the dual part which is from the schedule that the *SMV* method generated is the same (Eilon and Chowdhury, 1977).

iv. The sequence that achieves an optimal solution for min (*VCT*) has the property that the job with the largest processing time was ordered first.

v. The sequence that achieves an optimal solution for min (*VWT*) is *V*-shaped.

Using Kanet's method (Kanet, 1981), Bagchi (1989) firstly proposed an alternative way to min (*VWT*), which is equivalent to measure the total absolute differences in waiting times (*TADW*): $TADW = \sum_{i=1}^{n} \sum_{j=i}^{n} | W_j - W_i | = n^2(VWT) = \sum_{j=1}^{n} j(n-j) \cdot p_j$. Clearly, *TADW* is minimized by sorting weights $j(n-j)$ in a non-ascending order and the processing times in a non-descending order. Secondly, the weighted method was giving to find the optimal solution for dual objectives.

Overall, for a one-stage production problem, the SPT rule generates an optimal solution for min (*TCT*) (Li et al., 2014). The sequence that minimizes the variance of flow time is antithetical to the sequence that minimizes the variance of waiting time (Merten and Muller, 1972). The optimal sequence should be *V*-shaped for the waiting time variance minimization problem in the single machine (Eilon and Chowdhury, 1977). In conclusion, for inconsistent KPIs, we cannot find one sequence that simultaneously achieves optimal solution for each KPI optimization, which is one factor that triggers us to develop our ToB heuristic to balance the trade-offs between two inconsistent objectives.

2.3 Heuristics for multi-objective optimization

Although almost all multi-objective optimization problems are *NP*-hard and only a few can be solved by polynomial time (Pinedo, 2016), it is necessary to search for a near-optimal solution for multi-objective optimization problem in the real complex production environment. For example, achieving an optimum solution for patient flow time minimization leads to lower utilization of the periop process (Li et al., 2018).

A multi-objective optimization problem (MOP) with several minimizing objectives can be defined as follows (Li and Ma, 2016), min $F(x) = (f_1(x), f_2(x), \dots, f_k(x))$, subject to $x \in \Omega$, where the solution x is a vector of discrete decision variables and Ω is the decision space. Two procedures to solve a multi-objective optimization problem (Ciavotta et al., 2013). One is "priori" approach, which is giving each objective a preference, i.e., a weight, to generate a single weighted linear function. The other is "posteriori" approach, aiming to find out a set of solutions (Pareto front). The decision-maker just picks one solution from the Pareto front.

Followings are the literature reviews on the heuristics that using "priori" approach or "posteriori" approach.

Dhingra and Chandna (2010) proposed HAS algorithms based on the NEH heuristic, to find an optimal sequence to minimize the weighted sum of total weighted tardiness, total weighted earliness, and makespan. NEH insertion technique and six generating rules were considered when proposing HAS algorithms. From the results of experiment which instances were derived by Taillard (1993), HAS algorithms were superior to others with weights were setting as (0.33, 0.33, 0.33), (0.25, 0.25, 0.5), (0.5, 0.25, 0.25) and (0.25, 0.5, 0.25) for multi-objective function.

Using the local search technique, Li and Ma (2016) also presented a novel multi-objective memetic search algorithm (MMSA) to find an optimal schedule with makespan and total flowtime minimization objectives. First, the NEH heuristic-based method was applied for initializing the population, and individuals in the population were considered as P_L. Then search for a non-dominated solution in P_L and put them into P_E. Second, the global search method, the further local search method, and the update method were used to generate a set of P_L and update P_L. Then continue to update A={$P_L \cup P_E$}, P_E=A, until the Pareto optimal set was generated. The experimental results showed that MMSA was better than NNMA (Chiang et al., 2011), MOLSD (Li and Li, 2015), MOMAD (Ke et al., 2014), and RIPG (Minella et al., 2011).

Chandrasekaran et al. (2007) generated a particle swarm optimization (PSO) algorithm for solving the multi-objective flow shop scheduling problem, i.e., minimizing makespan, flow time, and completion time variance simultaneously. Generally, the PSO algorithm solves continuous non-linear optimization problems, mimicking the behavior of birds and their patterns of information exchange. The experiment was prepared to solve problems with jobs ranging from 20 to 500 and machines ranging from 5 to 20 and did not compare to other algorithms. The result was a Pareto solution set whose performance can be improved by increasing the number of iterations.

Bagchi (1989) used the weighted method to simultaneously minimize the mean and the variation of flow time and waiting time in single-machine systems. Given p_j with j = 1, …, N for processing times of N jobs processing on the non-preemptive one-stage scenario, the author modeled the bicriterion scheduling problems as a cost function of the mean and variance of completion times, and considered total absolute differences in completion times (*TADC*) as a measure of the variation and total completion time (*TCT*) as a representative for mean completion time. The cost function with preference α are as following,

$$Z^c = \alpha(TCT) + (1-\alpha)(TADC) = \sum_{j=1}^{n} w_{j,\alpha}^{c}\, p_{[j]},\ 0 \leq \alpha \leq 1 \tag{6}$$

$$Z^W = \alpha(TW) + (1-\alpha)(TADW) = \sum_{j=1}^{n} w_{j,\alpha}^{w}\, p_{[j]},\ 0 \le \alpha \le 1 \tag{7}$$

where $w_{j,\alpha}^{c} = (2\alpha - 1)(n+1) + j\{2 - 3\alpha + n(1-\alpha)\} - j^2(1-\alpha)$.

where $w_{j,\alpha}^{w} = \alpha n + j\{n - \alpha(1+n)\} - j^2(1-\alpha)$, *TW* is total waiting time.

From the literature review above, we found that the articles with multi-objective optimization most focused on deterministic scheduling, that is there is no variation in processing times. However, uncertainty is everywhere in a real production scenario, which results in stochastic scheduling. Although Daniels and Kouvelis (1995) proposed EP and ES heuristics to hedge against processing time uncertainty by optimizing the worst-case scenarios, we raise a question that does optimize the worst-case scenarios also optimize the expected value of a KPI? We will answer the question in Chapter 4.

2.4 Modern portfolio theory (MPT)

Markowitz (1952) developed the modern portfolio theory for investment, the objective of which is to maximize the expected return for a given level of risk. Assuming a number of *K* assets are available in the market, each of which has a return of R_k for *k* = 1, …, *K*, we need to invest 100% capital onto *K* assets with two objectives, to maximize the expected return (*E*) and to minimize the variance of the portfolio return (σ^2). The expected return is

$$E = \sum_{k=1}^{K} w_k R_k = W^T R \tag{8}$$

where w_k is the weight or the percentage of capital invested on an asset *k*, with $\sum_{k=1}^{K} w_k = 1$, *W* and *R* are the vector of portfolio weights and the vector of expected returns respectively, and *T* stands for transpose. The variance of the portfolio return is

$$\sigma^2 = \sum_{k=1}^{K} w_k^2 \sigma_k^2 + \sum_{k=1}^{K} \sum_{c=1, c\neq k}^{K} w_k w_c \sigma_k \sigma_c \rho_{kc} = W^T \Sigma W \tag{9}$$

where ρ_{kc} is the correlation coefficient between the returns on assets k and c, Σ is a *K* by *K* covariance matrix for the returns on the assets in the portfolio. If ρ_{kc}=0, it means that all the asset pairs are uncorrelated. While if ρ_{kc}=1, it means that all the asset pairs are positively correlated.

Regarding KPIs as assets, the MPT model can be applied to balance trade-offs among KPIs in one-stage production.

The efficient portfolio frontier offers analytical advice for risk-averse investors to make decisions that allocating capital to different assets. Any point on the efficient frontier means, for a given risk σ, the expected return cannot be further maximized, or for a given expected return *E*, the risk cannot be further minimized. However, when it comes to stochastic variables, the MPT model does not work unless the actual processing times are unknown in advance and we measure the performance in terms of the mean of processing time.

2.5 Stochastic Control

Generally, a stochastic process is a sequence of random variables that are related by time *T*. Both the sequence and each of the random variables can be continuous or discrete. The control theory is applied to a stochastic process, namely stochastic control.

Stochastic control theory deals with the system that with uncertainty or disturbance and aims at answering the following questions (Astrom, 1970):

i. What are the statistical properties of the system variables?
ii. How to adjust the unknown parameters of the system to optimize the system under the given criteria?
iii. How to find a control law aiming to minimize the criterion?

The first question is the most basic question of stochastic control theory based on probability and statistics techniques. We can use some properties to describe the random variable, such as probability distribution for a discrete random variable and probability density function for the continuous random variable, or expected value, variance, covariance and correlation. Because of the statistical properties of the system variables, generally, researchers hope to evaluate the endpoint of the stochastic process and find an optimal method to maximize or minimize the expected value of the random variable. As for the second and third questions, the variation range and distribution of each known and unknown parameter are important. They will affect the solution space for cost function, such as equations (6) and (7), which means that it is necessary to address single objective (single parameter) production problems. With the variation in production, we can measure its mean and variance as our control objectives.

In the real production scenario, randomness and uncertainty are everywhere, such as the skill levels of operators, the condition of stages, machine operating environment changes, or raw material quality parameters fluctuation which brings about the uncertain job processing times. Several methods have been used to describe the uncertainty in scheduling problem, such as Probability distribution function when the historical data is available, Fuzzy description, on the contrary, when probabilistic information is not ready, and the method that Daniels and Kouvelis (1995) (DK) used, which is bounded form, i.e., lower and upper limits $[\underline{p_j}, \overline{p_j}]$. DK proposed a scheme for one-stage production robust scheduling against uncertainty in processing times, which to maximize the minimum deviation from the upper bound of total completion time. The scheme was presented in two heuristics, the endpoint production (EP) and the endpoint sum (ES), which schedule the expected processing time due to the variation of actual processing times. They described the uncertainty in processing times by an interval $p_j \in [L_j, U_j]$, where L_j and U_j are the lower bound and the upper bound for the processing time of job j, respectively. Moreover, they use these two bounds to sequence the jobs in nondecreasing order. Since we are comparing our methods with DK's, it is necessary to introduce and explain their heuristics in detail. The following are the detailed steps for EP and ES heuristics.

Given p_j with j = 1, …, N for processing times of N jobs producing on the one-stage scenario:

Step 1. Calculate the lower bound L_j, the average or expectation E_j, and the upper bound U_j for stochastic processing times.

Step 2. Sort L_j according to the shortest processing time (SPT) rule, which is the lower bound of *TCT*, denote as *LB*(*TCT*).

Step 3. Sort U_j according to the longest processing time (LPT) rule, which is the upper bound of *TCT*, denote as *UB*(*TCT*).

Step 4. Keep the rest N – 2 jobs in the same position and exchange the positions of two jobs, j and l in the sequence S1 = [1, 2, …, j, …, l, …, N], we get sequence S2 = [1, 2, …, l, …, j, …, N].

According to equation (5), we get,

- The worst-case (*WC*) of *TCT* occurs to S1 under the condition of U_j and L_l, which has the effect of

  $WC_1 = (N - j + 1) \cdot U_j + (N - l + 1) \cdot L_l$, and (10)

- The worst-case of *TCT* occurs to S2 under the condition of U_l and L_j, which has the effect of

$$WC_2 = (N - j + 1) \cdot U_l + (N - l + 1) \cdot L_j \tag{11}$$

Step 5. If $WC_1 \leq WC_2$, that is

$$(N - j + 1) \cdot U_j + (N - l + 1) \cdot L_l \leq (N - j + 1) \cdot U_l + (N - l + 1) \cdot L_j, \tag{12}$$

we use sequence S1 to process job j earlier than job l, otherwise, we use S2.

We define the difference between L_j and L_l as $dL = L_l - L_j$, and the difference between U_j and U_l as $dU = U_j - U_l$. Figure 2 shows the deviation between the lower bound and upper bound. Then the equation (8) can be expressed as follows,

$$(N - j + 1) \cdot (U_j - U_l) \leq (N - l + 1) \cdot (L_l - L_j) \tag{13}$$

$$(N - j + 1) \cdot dU \leq (N - l + 1) \cdot dL \tag{14}$$

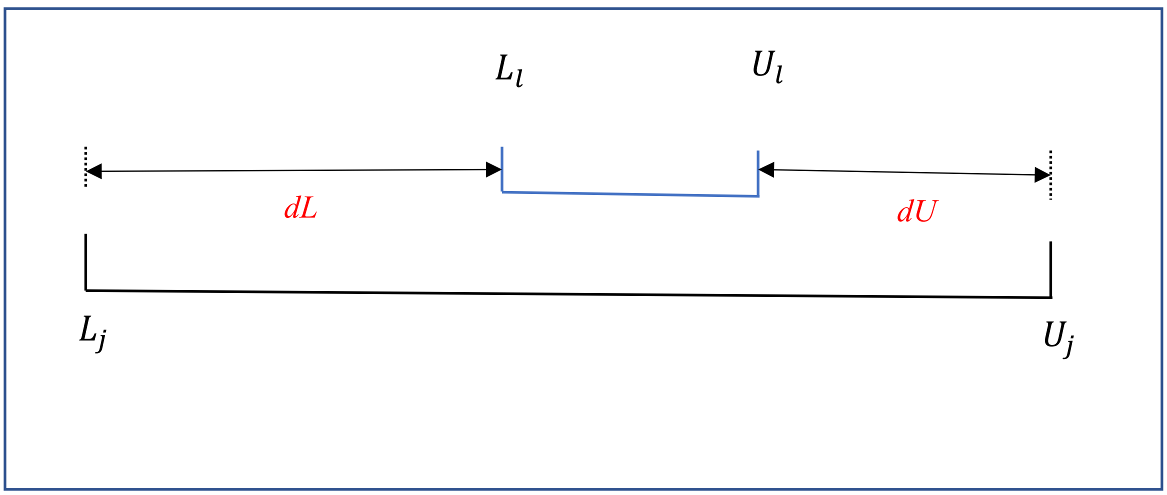


Figure 2: The deviation between lower bound and upper bound.

The programming logic of EP and ES is summarized as follows:

Applying the SPT rule to $L_j \cdot U_j$ and $L_j + U_j$, generating the EP and ES sequences respectively.

Given a sequence π for EP or ES, Figure 3 shows the programming logic of EP and ES heuristics.

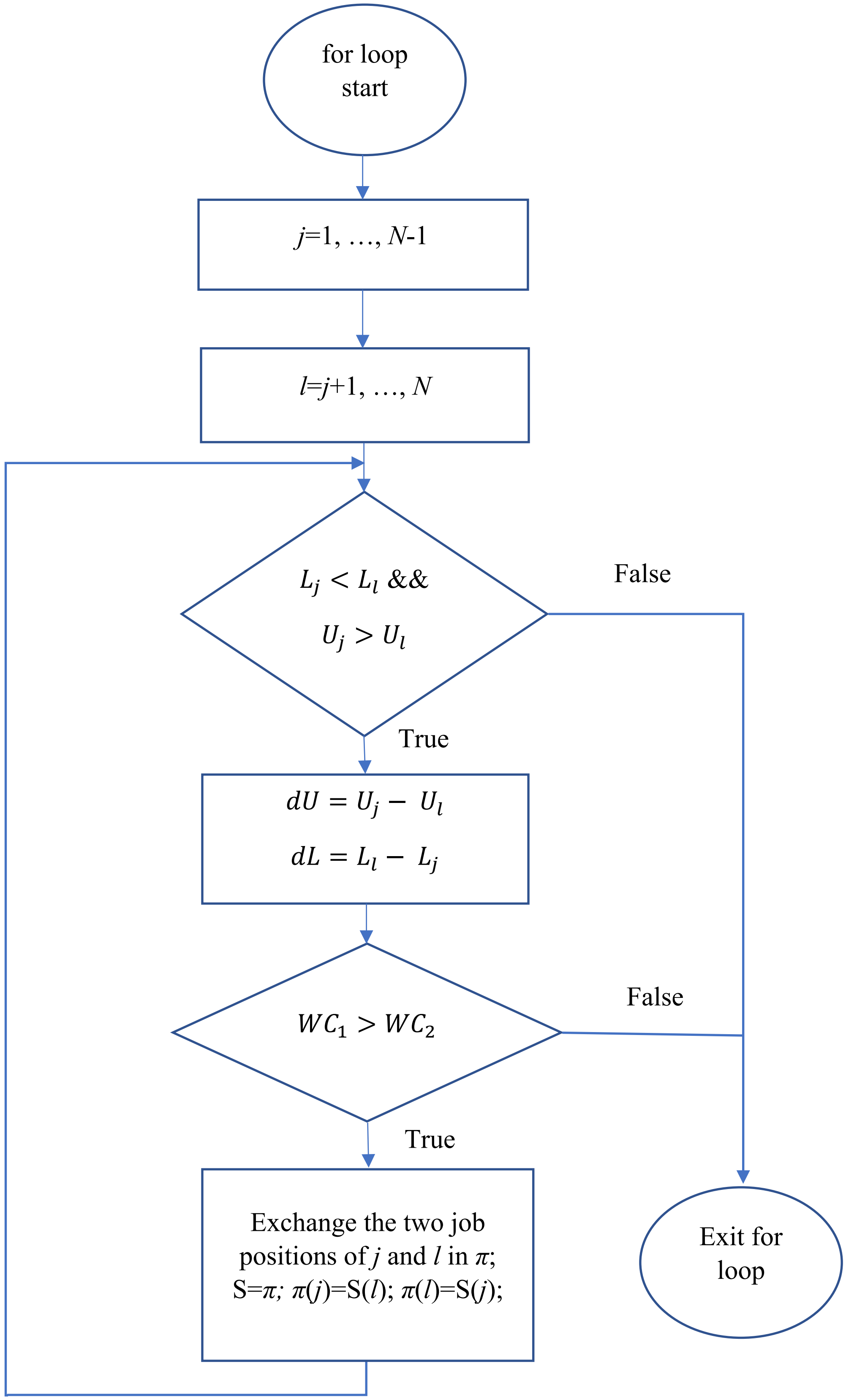


Figure 3: The programming logic of EP and ES heuristics.

In fact, EP and ES heuristics sequence the jobs according to the following rules,

$$L_j \cdot U_j \leq L_{j+1} \cdot U_{j+1}, \text{ and} \tag{15}$$

$$L_j + U_j \leq L_{j+1} + U_{j+1}, \text{ respectively.} \tag{16}$$

To optimize the worst-case performance, DK's scheme is more appropriate and gives useful insight for robust scheduling. However, we challenge DK's scheme that a solution to optimize the worst-case does not necessarily optimize the average expected performance. Therefore, our scheduling scheme for trade-off balancing considers not only the worst-case scenario, but also the average expected performance. Accordingly, our ToB heuristic outperforms DK's EP and ES heuristics on both the worst-case and average performances.

2.6 Statistical process control

Based on the concept of exchangeability, Shewhart (1931) proposed the concept of a state of statistical control which is the precursor to the statistical process control (SPC) method, and successfully promoted and applied it in the communication industry and military industry.

According to John (2003), the SPC is a basic set of tools for process management, improving the process design, enhancing the consistency, reducing production costs, and improving the quality of products from a process by controlling input factors. The benefits of the SPC include but not limited to the following:

i. The application range is very wide. It can be used in any process in which output is measured by certain specifications.
ii. The decision is rational.
iii. The involvement in the improvement process increases the 'awareness' of quality.

iv. The experience of the workforce is enhanced.

v. Leaders are more methodological.

vi. Communication is improved.

Ishikawa (1974) developed 7-QC tools, which are

i. Stratification/Divide and Conquer Method,

ii. Histogram,

iii. Check Sheet/Tally Sheet,

iv. Cause-and-Effect/Fishbone/Ishikawa Diagram,

v. Pareto Chart/80-20 Rule,

vi. Scatter Diagram, and

vii. Control/Shewhart Chart,

and 7-SUPP (Ishikawa, 1974), which are

i. Stratification,

ii. Defect Mapping,

iii. Events Logs,

iv. Flowchart,

v. Progress Centers,

vi. Randomization, and

vii. Sample Size Determination.

These tools help us understand the application breadth of the SPC. The other useful concept is statistical quality control (SQC). The difference between SQC and SPC is the application scope of the above tools. Using these tools to observe the outputs which are dependent factors is the process of SQC while using these tools to control the inputs which are independent factors is the process of the SPC.

Duncan (1959) introduced the general theory of control charts which are the most basic tool for the SPC technique. It not only a tool for illustrating a stage of statistical control and achieving the purpose of control but also a tool for indicating what level the control reached.

A typical control chart contains two parts. One is the centerline which stands for the average value of observed variables. The other part is two control limit lines, i.e., the upper control limit (*UCL*) and the lower control limit (*LCL*). Generally, if all or nearly all samples fall between the control limit, we could consider that the process is in control. But sometimes, the process was under suspicion of being out of control even if all sample points did fall between the control limit (Montgomery, 2009). Figure 4 is an example of such a situation. As shown in Figure 4, all points fall between *UCL* and *LCL*, however, two of these points plotted upper the center line while others fall below the centerline. Clearly, the $\bar{x}$ chart in Figure 4 has no random pattern. The other type of control charts is the *R* charts.

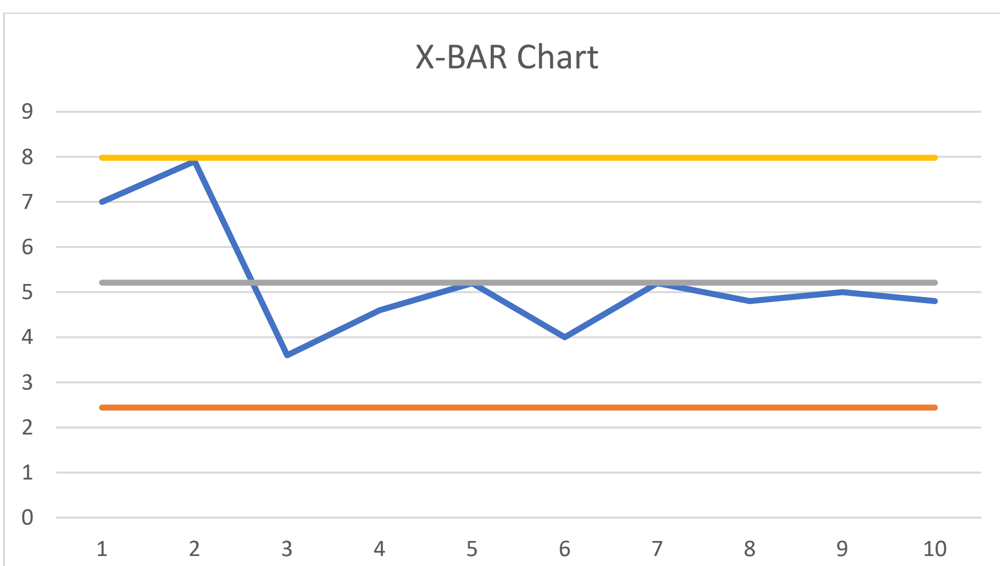


Figure 4: The control chart example.

Montgomery (2009) introduced two Phases of applications of $\bar{x}$ charts and R charts:

Phase one: Suppose that a quality characteristic is normally distributed with mean μ and standard deviation σ, where both mean μ and standard deviation σ are known. If x_1, x_2, …, x_n is a sample of size n, then the average of this sample is $\bar{x} = \frac{x_1+x_2+ \ldots +x_n}{n}$. Suppose that m samples are available, each containing n observations on the quality characteristic. Let $\bar{x}_1$, $\bar{x}_2$, …, $\bar{x}_m$ be the average of each sample. $\bar{\bar{x}} = \frac{\bar{x}_1+\bar{x}_2+ \ldots +\bar{x}_m}{m}$. The range of the sample is $R = x_{max} - x_{min}$. Let R_1, R_2, …, R_m be the ranges of the m samples. The average range is $\bar{R} = \frac{R_1+R_2+ \ldots +R_m}{m}$. Then the control limits for the $\bar{x}$ chart are: UCL $= \bar{\bar{x}} + A_2\bar{R}$ and LCL $= \bar{\bar{x}} - A_2\bar{R}$. The control limits for the R chart are: UCL $= D_4\bar{R}$ and LCL $= D_3\bar{R}$. A_2 is a constant which is tabulated for various sample size. D_3 and D_4 are constants that are tabulated for various values of n. After plotting the $\bar{x}$ chart and the R chart, the next step is analyzing the result. If all points are between the control limits and there is no trend of shifting, we can conclude that the process is in control in the past and use the control limits for process control in the future (Montgomery, 2009).

Phase two: Using the reliable control limits generated by Phase one to monitor future production (Montgomery, 2009).

It is crucial to draw suitable control limits. With narrow control limits, the probability of 'type I error' goes up. However, with wide control limits, the risk of having 'type II error' is increased. Shewhart (1931) introduced a recommendation of setting ± 3*standard deviations for balancing the risk of 'type I error' and 'type II error'.

Leavengood and Reeb (1999) further summarized SPC with two advantages of the application, that is, Defects are effectively prevented by monitoring and controlling variation, and substantial improvement was achieved by improving the performance of the system and avoiding or reducing variation.

In addition to control charts, two process capability indices of C_p and C_{pk} are also extensively used in the industry:

$$C_p = \frac{USL-LSL}{6\hat{\sigma}} \tag{17}$$

$$C_{pk} = \min\,(C_{pu}, C_{pl}) \tag{18}$$

where lower and upper specification limits, [*LSL*, *USL*] are generally determined externally, such as customer preference, $\hat{\sigma}$ is the estimated process standard deviation, C_{pl} and C_{pu} are one-sided process capability ratios. Given estimated sample mean for the population, $\hat{\mu}$, C_{pl} and C_{pu} can be calculated by

$$C_{pl} = \frac{\hat{\mu}-LSL}{3\hat{\sigma}} \text{ for the lower specification only, and} \tag{19}$$

$$C_{pu} = \frac{USL-\hat{\mu}}{3\hat{\sigma}} \text{ for the upper specification only.} \tag{20}$$

We can design reasonable specification limits to help decision-makers choose a heuristic with the largest value of C_p and C_{pk} when facing processing time uncertainty and trade-offs among KPIs.

CHAPTER 3. METHODOLOGY

The presentation of our methodology is organized as follows. In Section 3.1, we provide a problem description with assumptions and notations, followed by a summary of mathematical definitions of total completion time (*TCT*) and variance of completion times (*VCT*). In Section 3.2, we present our ToB heuristic with its formulation scheme and mathematical description. In Section 3.3, we discuss how we integrate the MPT model with our ToB heuristic. In Section 3.4, we introduce how can we apply the SPC techniques for robust production control.

3.1 Problem description

The assumptions for the problem are stated as follows,

i. The stage has been set up at time zero.
ii. All jobs are available at time zero.
iii. Preemption is not allowed.
iv. The job sequence cannot be changed during the operation.
v. The stage just can operate one job at a time.

Notations are used in problem description and formulation are as follows,

N: the number of jobs;

p_j: the processing time of job j on the machine, where $j=1, \ldots, N$;

I: the total instances number, $i = 1, \ldots, I$;

V: the levels of coefficient of variation (CV) in processing times,

$v = 1, \ldots, V$;

S: the samples randomly generated for each CV level, $s = 1, \ldots, S$;

H: the heuristics for sequencing for $h = 1, \ldots, H$;

K: the number of KPIs, $k = 1, \ldots, K$ for general cases, for this thesis work k=1 for TCT and k=2 for VCT;

C_j: the completion time for job j;

w_k: the weight for KPI k.

3.1.1 Total completion time (*TCT*)

For N jobs one-stage production problem, we have N! possible sequences, which is our solution space. Because there is no setup time for the stage and all jobs are available at time zero, we can calculate maximum completion time (MCT) and total completion time (TCT) as following equations. Figure 5 is the Gantt chart for one-stage production problems. As shown in Figure 5, the completion time of the first job equals the processing time of the first job,

$$C_1 = p_1 \tag{21}$$

then we can calculate MCT, also called makespan or C_{max} by

$$C_j = C_{j-1} + p_j, \text{ with } C_0 = 0 \tag{22}$$

$$C_{max} = C_N = \sum_{j=1}^{N} p_j \tag{23}$$

and TCT, also called flow time or $\sum C_j$, by

$$TCT = \sum C_j = \sum_{j=1}^{N} C_j = \sum_{j=1}^{N} (N - j + 1) \cdot p_j \tag{24}$$

which is the sum of weighted processing times. We set $W_1 = (N - j + 1)$.

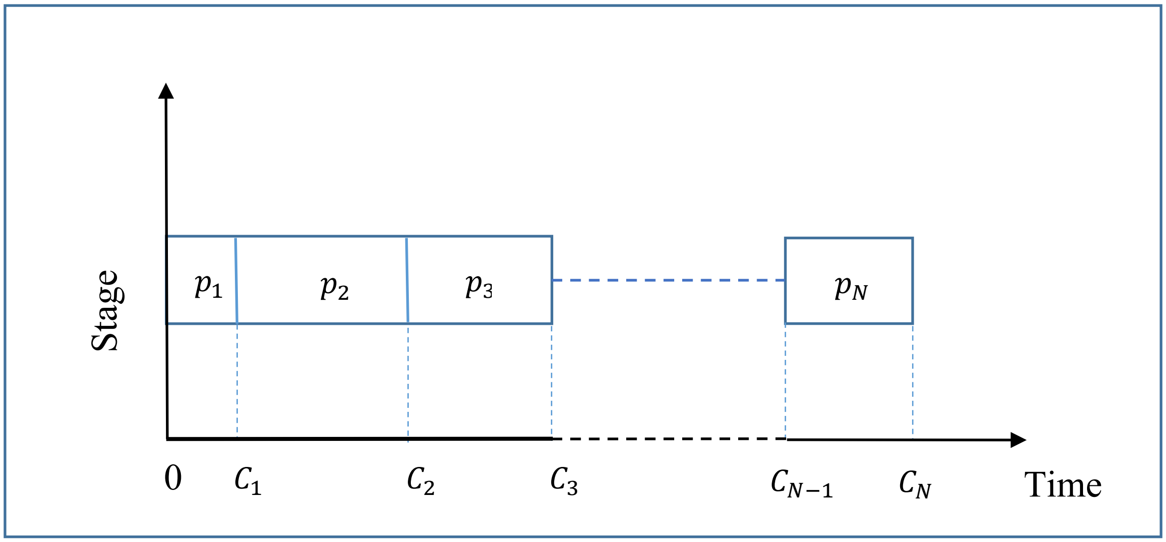


Figure 5: Gantt chart for one-stage production problem.

For deterministic N jobs one-stage production, the processing time of all the jobs on the stage is determined, so C_{max} is a constant. But for *TCT*, its weights $(N - j + 1)$ are independent of processing times. The optimal solution for min (*TCT*) is ordering jobs in a nondecreasing sequence, i.e., the SPT rule.

3.1.2 Variance of completion times (*VCT*)

The other important measurement for the production performance is the variance of completion times (*VCT*), formulated by the following equation,

$$VCT = \frac{1}{N}\sum_{j=1}^{N}(C_j - MFT)^2 \tag{25}$$

where *MFT* is the mean flow time, i.e.,

$$MFT = \frac{TCT}{N} \tag{26}$$

Although Eilon and Chowdhury (1977) have already presented that optimal flow time variance sequence must be *V*-shaped which orders the jobs that before the shortest job in descending order of processing times (LPT rule) and after the shortest job in ascending order of processing times (SPT rule), Kubiak (1993) showed that minimizing *VCT* is *NP*-hard. It has led researchers to conduct extensive and in-depth studies. Kanet (1981) generate an easier way to find the optimal of min(*VCT*), which is minimizing the total absolute differences in completion times (*TADC*), i.e., minimizing

$$TADC = \sum_{j=1}^{N} (j-1)(N-j+1) \cdot p_j \tag{27}$$

which is also the sum of weighted processing times. We set $W_2 = (j-1)(N-j+1)$. W_2 is also independent on processing times.

3.2 ToB Heuristic for one-stage production scheduling

Since W_1 is a first-order equation and W_2 is second-order equation of j, we can tell the inconsistency between min (*TCT*) and min (*VCT*) by equations (24) and (27) which is the first source of trade-offs.

Our ToB heuristic aims to balance the trade-off between the two KPIs, flow time minimization and completion time variance minimization in one-stage production. The scheme is to allocate preference on the two KPIs using modern portfolio theory (MPT), so we introduce α to *TADC* in equation (27) and $(1-\alpha)$ to TCT in equation (24), generating out our ToB heuristic as follows,

$$z = \alpha \cdot TADC + (1-\alpha) \cdot TCT \tag{28}$$

$$= \alpha \cdot \sum_{j=1}^{N} (j-1)(N-j+1) \cdot p_j + (1-\alpha) \cdot \sum_{j=1}^{N} (N-j+1) \cdot p_j$$

$$= \sum_{j=1}^{N} [(j-2)\,\alpha + 1](N-j+1) \cdot p_j \tag{29}$$

The value z is the sum of weighted processing times. We also set weight

$$W_3 = [(j-2)\alpha + 1](N-j+1). \tag{30}$$

Equation (29) is a quadratic function of *j*for a given α and a finite set of *N* jobs. Since the parabola opens downward (the quadratic coefficient$<$ 0), we can take the first-order derivative with respect to *j*, W_3 reaches its maximum at $j = \frac{N+3}{2} - \frac{1}{2\alpha}$. The main scheme of our ToB heuristic is sorting the processing times in descending order and sorting weights in ascending order, then matching the two orders together to get a sequence.

Changing preference α = 0.0 : 0.1 : 1.0, our ToB heuristic generates 11 sequences for trade-off balancing. When α = 0.0 and α = 1.0, according to the equation (28), it makes us completely inclined to min (*TCT*) and min (*VCT*), respectively. The overall computational complexity of our ToB heuristics is only $O(N\log N)$, the same as that of LEPT or SEPT, but much simpler than that of EP or ES.

3.3 Modern portfolio theory (MPT)

Based on the mathematical summary of the MPT model in Chapter Two, our ToB heuristic is formed by allocating preference on the two KPIs. The normalized deviations of the ToB heuristic for *TCT* and *VCT* are plugged into equations (8) and (9) in calculating the expected return (*E*) and the risk (σ^2):

$$E = w_1 \Delta TCT + w_2 \Delta VCT \tag{31}$$

$$\sigma^2 = W^T \Sigma W = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$= w_1^2 \sigma_1^2 + 2 w_1 w_2 \rho \sigma_1 \sigma_2 + w_2^2 \sigma_2^2 \tag{32}$$

with w_1, w_2 $\geq$0 and $w_1 + w_2$= 1.

We have already known that the inconsistency between min (*TCT*) and min (*VCT*) by equations (24) and (27), which means ρ is equal to neither 0, 1 nor –1. The properties of a linear function (31) and a quadratic function (32) show the second source of trade-offs between expected return and risk.

When we have a set of portfolios (several weights), the efficient portfolio frontier will be generated. The efficient portfolio frontier offers analytical advice for making decisions that balancing the trade-offs. Any point on the efficient frontier means, for a given risk σ, the expected return cannot be further maximized, or for a given expected return E, the risk cannot be further minimized. For a portfolio of normalized deviations, the smaller the expected value of E and the smaller the risk of σ, the better the trade-offs balancing.

3.4 Statistical process control (SPC)

According to Chapter Two, Section 2.6, $\bar{x}$ chart and R chart are the two most basic control charts. In our research, given numbers of I instances and numbers of S samples for each instance, the sample mean for an instance on asset k can be calculated by

$$\bar{x}_{k,i} = \frac{1}{S}\sum_{s=1}^{S} d_{k,i,s} \text{ for } i = 1, \ldots, I,\ k = 1, \ldots, K, \tag{33}$$

where $d_{k,i,s}$ is the return from asset k for instance i in sample s. To generate an $\bar{x}$ chart, we need the value of the centerline and two control limits, i.e., the upper control limit (UCL) and the lower control limit (LCL). For $\bar{x}$ chart, the value of centerline is the grand mean of asset k:

$$\overline{X}_k = \frac{1}{I}\sum_{i=1}^{I} \bar{x}_{k,i} \text{ for } k = 1, \ldots, K. \tag{34}$$

The variation range for an instance can be calculated by

$$R_{k,i} = \max_{\{S\}}\left(d_{k,i,s}\right) - \min_{\{S\}}\left(d_{k,i,s}\right). \tag{35}$$

For R chart, the value of the centerline is the average variation range:

$$\bar{R}_k = \frac{1}{I}\sum_{i=1}^{I} \bar{R}_{k,i} \text{ for } k = 1, \ldots, K. \quad (36)$$

The control limits for the $\bar{x}$ chart are:

$$\text{UCL} = \overline{X}_k + A_2\bar{R}_k, \quad (37)$$

$$\text{UCL} = \overline{X}_k + A_2\bar{R}_k, \quad (38)$$

and the control limits for the R chart are:

$$\text{UCL} = D_4\bar{R}_k \quad (39)$$

$$\text{LCL} = D_3\bar{R}_k \quad (40)$$

C_p and C_{pk} are defined in Section 2.6. They are useful to verify if the process is under control. Qualitatively, the larger the C_p and C_{pk}, the better the process is under control. Quantitatively, C_p has a useful practical interpretation, that is $P = (\frac{1}{C_p})100$, the percentage of a specification band used by the process. A large value of P means the process is well controlled. Additionally, because C_{pk} measures the process centering, a large value of C_{pk} indicates the process fluctuates around the target and the process performance is consistent over time.

CHAPTER 4. CASE STUDY

To test the ability of our ToB heuristic in addressing three challenges in one-stage production scheduling, we design our case studies comprehensively. This chapter is organized as follows. In section 4.1, we illustrate the design scheme of our case studies. In section 4.2, we compare the performance of 16 heuristics on single-objective optimization problems and for two types of processing times, one of which is for static processing times, and the other for variation in processing times. In section 4.3, we present the fluctuations of trade-off balancing along with variation in processing times, based on the modern portfolio theory model. We also illustrate how we use our ToB heuristic to further optimize the expected return and the risk for multi-objective optimization. In section 4.4, we present how we use our ToB heuristic to facilitate stochastic production control, in terms of statistical process control (SPC) techniques.

4.1 The design scheme of case studies

4.1.1 A list of variables

i. The number of jobs ranges from $N = 5, 6, \ldots, 10$.

ii. The number of instances is 50 for each job number. The total number of instances is $I = 6\times50 = 300$ for $i = 1, \ldots, I$.

iii. The processing times for each instance are randomly generated following a uniform distribution between [1, 99].

iv. To describe the processing time uncertainty, we introduce a measurement of a probability distribution or frequency distribution, the coefficient of variation (CV). Its mathematical definition is a ratio of the standard deviation σ to the mean μ,

$CV = \frac{\sigma}{\mu}$. Since we can estimate CV by using the ratio of the sample standard deviation to the sample mean, we determined the sample by $p = E(p) + \sqrt{3}E(p)CV(2U - 1)$, where U is a uniform random number from [0, 1], $E(p)$ is the expected value for processing time. We set $CV \leq 1/\sqrt{3}$ to avoid processing times falling below zero which cannot happen in a production scenario. So, the CV changes in the interval [0.1, 0.5] with increments of 0.1. Thus, we have 5 CV levels in total for $v = 1, \ldots, V$.

v. $S = 50$ samples for each CV level and each instance with $s = 1, \ldots, S$. The total number of samples = The total number of instances×The total number of CV levels×The number of samples for each CV level and each instance = 300·×5×50 = 75,000.

vi. $H = 16$ heuristics for evaluation with $h = 1, \ldots, H$. We aimed to compare our ToB heuristic (11 sequences in total) with EP and ES heuristics. Since the shortest expected processing time (SEPT) rule is a classic stochastic scheduling method to address processing time uncertainty in min (TCT), we take it into consideration. And we also take account of the longest expected processing time (LEPT) rule, the antithetical to the sequence generated by the SEPT rule. Our ToB heuristic, SEPT, and LEPT rules sequences the expected processing times, $E(p_j)$, and the last two rules arrange the expected processing time in a nondecreasing order and a nonincreasing order, respectively. As introduced in Section 2.5, EP and ES heuristics operate the sequence of the job on lower and upper bounds of processing times. We also compared our ToB heuristic with the first come first served rule (FCFS) which does not depend on processing times and sequence jobs in the order in which they arrive. Because all jobs are available at time zero, we regard the sample sequence at time zero as the sequence generated by the FCFS rule. In total, 16 sequences are generated for each sample, with 11 sequences by the ToB heuristic, and one by each of the EP, ES, SEPT, LETP, and FCFS methods, respectively.

vii. We focus on two KPIs with k =1 for *TCT* and k =2 for *VCT*.

viii. $y_{k,v,i,s,h}$ is the actual value of a KPI k generated from a heuristic h for instance i in sample s on level v. Generally, since there are different units or scales for different KPIs, normalization should be considered. The units for our data are identical, but we cannot be sure that samples are on the same scale. So, it is necessary to normalize KPIs on the same scale. Min-max feature scaling is a common method to bring the data into the range [0, 1].The best (minimum) and worst (maximum) solutions to minimize total completion time (*TCT*) and completion times variance (*VCT*) are $\max_{\{H\}}(y_{k,v,i,s,h})$ and $\min_{\{H\}}(y_{k,v,i,s,h})$, respectively. The following expression will be used for results analysis of the case studies:

Normalized Deviation (*ND*):

$$ND_{k,v,i,s,h} = \frac{y_{k,v,i,s,h} - \min_{\{H\}}(y_{k,v,i,s,h})}{\max_{\{H\}}(y_{k,v,i,s,h}) - \min_{\{H\}}(y_{k,v,i,s,h})} \tag{41}$$

Average Normalized Deviation (*AND*):

$$AND_{k,v,h} = \frac{1}{I}\sum_{i=1}^{I}\left(\frac{1}{S}\sum_{s=1}^{S} ND_{k,v,i,s,h}\right) = \frac{1}{IS}\sum_{i=1}^{I}\sum_{s=1}^{S} ND_{k,v,i,s,h} \tag{42}$$

Maximum Normalized Deviation (*MND*):

$$MND_{k,v,i,h} = \max_{\{S\}}(ND_{k,v,i,s,h}) \text{ for each } CV \text{ level.} \tag{43}$$

Average Maximum Normalized Deviation (*AMND*):

$$AMND_{k,v,h} = \frac{1}{I}\sum_{i=1}^{I} MND_{k,v,i,h} \text{ for all instances.} \tag{44}$$

As for static processing times, there will be no subscript of *v* for the above expressions.

4.1.2 Evaluation scheme

Based on three challenges that motivate us on one-stage production scheduling, we designed our case study. The following strategies corresponded to three challenges we faced.

i. Given the challenge that trade-offs between two KPIs, i.e., the trade-offs between min (*TCT*) and min (*VCT*) in production scheduling. The proposed ToB heuristic was generated to balance the trade-offs. We compare the performance of 16 heuristics on single-objective optimization problems, i.e., min (*TCT*) and min (*VCT*), and for two types of processing times, one of which has static processing times, and the other has variation in processing times. Results are shown in Section 4.2.
ii. Given the challenge that trade-offs between the mean and the variance in multi-objective optimization, we integrate the concept of modern portfolio theory (MPT) into our ToB heuristic to balancing the trade-offs between the expected return and the risk. Firstly, we show the fluctuations of trade-off balancing along with variation in processing times. Then, we also present how our ToB heuristic facilitates multi-objective optimization based on the modern portfolio theory model. Results are shown in Section 4.3.

iii. Given the challenge that uncertainties exist in the real production environment, we use SPC techniques to show fluctuations of process performance, how we set up specification limits and how our ToB heuristic facilitates SPC techniques to generate specification limits. Results are shown in Section 4.4.

4.2 Single-objective optimization

To verify the effectiveness of our ToB heuristic in balancing trade-offs among KPIs, we compare the performance of 16 heuristics on single-objective optimization problems, i.e., min (*TCT*) and min (*VCT*). The following is the result for two types of one-stage production. The first type is the production with static processing times. The other one is the production with variation in processing times. The smallest values will be highlighted in bold in the following subsections.

4.2.1 For static processing times

For the production with static processing times, the results are shown in Table 1 and Table 2. From Table 1, we can tell that, with single-objective min (*TCT*), our ToB heuristic with $\alpha = 0.0$ is the same as the SEPT rule, has the same performance as the EP and ES heuristics with the smallest average normalized deviation (*AND*) and smallest maximum normalized deviation (*MND*). From Table 2, we can tell that, with single-objective min (*VCT*), our ToB heuristic with $\alpha = 0.5$ to 1.0 has the smallest *AND* and the smallest *MND*. Additionally, we can tell the inconsistencies between min (*TCT*) and min (*VCT*) for static processing times from both Table 1 and Table 2, in terms of a small deviation achieved by a heuristic on one KPI, but a large one on the other.

Table 1: *AND* and *MND* from min (*TCT*) for static processing times.

	ToB(α)							EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~0.9	1.0					
AND	**.000**	.001	.115	.236	.390	.610	.764	**.000**	**.000**	**.000**	1.000	.490
MND	**.000**	.038	.287	.363	.484	.895	.986	**.000**	**.000**	**.000**	1.000	.996

Table 2: *AND* and *MND* from min (*VCT*) for static processing times.

	ToB(α)						EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~1.0					
AND	.983	.978	.642	.381	.161	**.000**	.983	.983	.983	.289	.637
MND	1.000	1.000	1.000	.958	.726	**.001**	1.000	1.000	1.000	.540	1.000

4.2.2 For stochastic processing times

We manage the grand *AMND* for each number of jobs, which is across I = 50 instances in Table 3. From Table 3, we can tell that ToB(0.0) heuristic is the same as the SEPT rule, both of which generate the smallest grand *AMND* when the number of jobs is relatively large (N=7, 8, 9, 10). When the number of jobs is relatively small (N=5, 6), the EP and ES heuristics generate the smallest grand *AMND* of 0.262 and 0.237, respectively. Comparatively, the LEPT rule always generates the largest deviations. Our ToB heuristic outperforms other heuristics when the number of jobs is relatively large.

Table 3: *AMND* from min (*TCT*) from the number of jobs perspective.

N	ToB(α)							EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~0.9	1.0					
5	.264	.264	.372	.576	.787	.923	.973	**.262**	**.262**	.264	1.000	.749
6	.241	.241	.342	.670	.807	.913	.957	**.237**	**.237**	.241	1.000	.808
7	**.143**	**.143**	.319	.607	.762	.889	.945	.144	.145	**.143**	1.000	.741
8	**.154**	**.154**	.519	.677	.785	.882	.934	.164	.162	**.154**	1.000	.757
9	**.116**	**.116**	.482	.647	.761	.864	.920	.120	.120	**.116**	1.000	.737
10	**.106**	.121	.542	.650	.767	.849	.904	.107	**.106**	**.106**	1.000	.709

From each *CV* level perspective, we manage the *AMND* across I = 300 instances in Table 4. From Table 4, we can tell that the EP and ES heuristics generate the smallest maximum deviations from min (*TCT*) at *CV* = 0.2 and 0.3, respectively. Our ToB(0.0) heuristic has the same performance as the EP and ES heuristics with the smallest maximum deviation at *CV* = 0.1 and 0.4. However, ToB(0.0) heuristic generates the smallest maximum deviation of 0.425 at *CV* = 0.5 and the smallest grand average of 0.171 across 5 *CV* levels. Same as that in Table 3, the LEPT rule generates the largest deviations at all *CV* levels and the largest grand average.

Table 4: *AMND* from min (*TCT*) from the *CV* levels perspective.

CV	ToB(α)							EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~0.9	1.0					
0.1	**.008**	.010	.169	.333	.514	.721	.841	**.008**	**.008**	**.008**	1.000	.576
0.2	.048	.050	.261	.474	.657	.824	.906	**.047**	**.047**	.048	1.000	.670
0.3	.121	.124	.397	.636	.811	.917	.960	**.120**	**.120**	.121	1.000	.762
0.4	**.250**	.253	.578	.812	.926	.975	.988	**.250**	**.250**	**.250**	1.000	.844
0.5	**.425**	.429	.740	.935	.983	.997	.998	.436	.435	**.425**	1.000	.898
Avg.	**.171**	.173	.429	.638	.778	.887	.939	.172	.172	**.171**	1.000	.750

We manage the grand *AND* for each number of jobs in Table 5 and for individual *CV* level in Table 6. From Table 5, we can tell that ToB(0.0) heuristic generates the smallest deviations from min (*TCT*) for all job numbers and ToB(0.1) heuristic generates the smallest deviations from *N* = 5 to *N* = 9. We can also tell that as the number of jobs increases, the deviation tends to decrease for all heuristics except for the LEPT rule and the FCFS rule. From Table 6, we can tell that our ToB(0.0) heuristic is the same as the SEPT rule, both of which generate the smallest deviation among all heuristics for each of 5 *CV* levels, and the average normalized deviation across 5 *CV* levels is only 0.025. Comparatively, the LEPT rule always generates the largest deviations, and its grand average deviation across 5 *CV* levels is 0.982. The grand average deviations across 5 *CV* levels are 0.027 and 0.026 for the EP and ES heuristics, respectively. Our ToB heuristic outperforms the EP and ES heuristics at all 5 *CV* levels.

Table 5: *AND* from min (*TCT*) from the number of jobs perspective.

N	ToB(α)							EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~0.9	1.0					
5	**.045**	**.045**	.088	.187	.369	.638	.820	.046	.046	**.045**	.963	.485
6	**.036**	**.036**	.078	.253	.396	.612	.754	.038	.037	**.036**	.971	.533
7	**.018**	**.018**	.093	.244	.392	.613	.761	.019	.019	**.018**	.988	.491
8	**.022**	**.022**	.187	.289	.418	.588	.718	.024	.024	**.022**	.985	.498
9	**.015**	**.015**	.187	.293	.418	.588	.713	.017	.017	**.015**	.991	.495
10	**.014**	.020	.228	.319	.436	.571	.688	.016	.016	**.014**	.993	.467

Table 6: *AND* from min (*TCT*) from the *CV* levels perspective.

CV	ToB(α)							EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~0.9	1.0					
0.1	**.001**	.002	.117	.238	.391	.610	.763	.002	.002	**.001**	1.000	.491
0.2	**.007**	.008	.125	.248	.396	.608	.756	.008	.008	**.007**	.997	.492
0.3	**.016**	.017	.134	.259	.403	.604	.747	.017	.017	**.016**	.991	.497
0.4	**.035**	.036	.156	.277	.412	.598	.733	.037	.037	**.035**	.975	.494
0.5	**.065**	.066	.185	.300	.423	.589	.712	.069	.068	**.065**	.947	.500
Avg.	**.025**	.026	.143	.264	.405	.602	.742	.027	.026	**.025**	.982	.495

From equation (42), we can calculate the *AND* from min (*VCT*). Table 7 is from the number of jobs perspective and Table 8 is from the *CV* levels perspective. From Table 7 and Table 8, we can tell that ToB(0.5, …, 1.0) heuristics have the smallest average deviations at each number of jobs and individual *CV* levels, and the smallest grand one of 0.054 across all *CV* levels. However, the performance of the EP and ES heuristics is not very well.

Table 7: *AND* from min (*VCT*) from the number of jobs perspective.

N	ToB(α)						EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~1.0					
5	.933	.933	.815	.606	.314	**.072**	.933	.933	.933	.194	.642
6	.947	.947	.822	.451	.241	**.068**	.948	.948	.947	.281	.543
7	.951	.951	.738	.424	.213	**.047**	.951	.951	.951	.289	.624
8	.939	.939	.512	.334	.168	**.054**	.941	.941	.939	.356	.628
9	.945	.945	.487	.303	.152	**.043**	.946	.946	.945	.376	.690
10	.946	.921	.398	.252	.120	**.041**	.949	.947	.946	.421	.629

Table 8: *AND* from min (*VCT*) from the *CV* levels perspective.

CV	ToB(α)						EP	ES	SEPT	LEPT	FCFS
	0.0	0.1	0.2	0.3	0.4	0.5~1.0					
0.1	.978	.974	.640	.382	.162	**.004**	.978	.978	.978	.290	.633
0.2	.965	.961	.633	.382	.175	**.021**	.965	.965	.965	.302	.629
0.3	.950	.946	.633	.393	.196	**.048**	.950	.950	.950	.317	.626
0.4	.926	.922	.625	.404	.223	**.081**	.928	.928	.926	.334	.622
0.5	.898	.894	.614	.414	.250	**.117**	.902	.901	.898	.353	.621
Avg.	.943	.939	.629	.395	.201	**.054**	.945	.945	.943	.319	.626

Comparing the deviations in Table 6 which have the min (*TCT*) criteria and the deviations in Table 8 which have the min (*VCT*) criteria, ToB(0.0) heuristic has the smallest average deviations from min (*TCT*) and ToB(0.5, …, 1.0) heuristics have the smallest average deviations from min (*VCT*). It shows the inconsistencies between min

(*TCT*) and min (*VCT*) for stochastic processing times, in terms of a small deviation achieved by a heuristic on one KPI, but a large one on the other.

Pareto efficiency can be defined as follows (Li et al., 2019), given a set of H methods generating feasible solutions to min (y_k) problems for k = 1, …, *K*, a method *h* is Pareto efficient, if and only if there is no other $h' \in H$ such that $\forall k$ $y_{h',k} \leq y_{k,h}$ and $\exists k$ $y_{h',k} < y_{k,h}$. Accordingly, based on the grand averages in Table 6 and Table 8, we can determine that our ToB heuristic with 11 α is Pareto efficient and dominates the rest of the heuristics.

4.3 Multi-objective optimization

4.3.1 The fluctuations of trade-off balancing

Since our ToB heuristic dominates the rest of the heuristics, we eliminate the dominated heuristics which are the EP and ES heuristics, the SEPT rule, the LEPT rule, and the FCFS rule. Then we just plug the normalized deviations of our ToB heuristic for *TCT* and *VCT* into the equation (31) and (32) and set $[w_1 \quad w_2]$ as a 501 by 2 matrix, where $w_1 + w_2$= 1 and w_1 changing from 0 to 1 with an increment of 0.002 to get the expected return (E) and the risk (σ). The efficient portfolio frontiers at each *CV* level are shown in Figure 6. Accordingly, the value of the expected return (E) and the risk (σ) are in Table 9 for the different objectives of min (E) and min (σ), respectively. For a portfolio of normalized deviations, the smaller the expected value of E and the smaller the risk of σ, the better the trade-offs balancing. From Table 9, we can see that minimum E and minimum σ do not occur simultaneously at each *CV* level, that is, with the objective of min (E), the minimum E is 0.293 at *CV*=0.1 however with the objective of min (σ), the minimum σ is 0.066 at CV=0.1. We can find the same properties for the other CV levels. From Figure 6, we can see the fluctuations of trade-off balancing along with variation in processing times. Furthermore, it is shown that, as CV level increases from 0.1 to 0.5, the efficient portfolio frontier shifts from the left to the right, which means that, as processing time uncertainty increases, we need to take a larger risk of σ to achieve the same value of E, or at a given risk level of σ, we need to expect a larger value of E from the portfolio.

Table 9: Statistics from the MPT model at CV levels.

CV\Obj.	min(E)		min(σ)	
	E	σ	E	σ
0.1	**.293**	.400	.426	**.066**
0.2	.303	.397	.426	.075
0.3	.321	.395	.425	.085
0.4	.341	.393	.426	.095
0.5	.363	.389	.428	.106

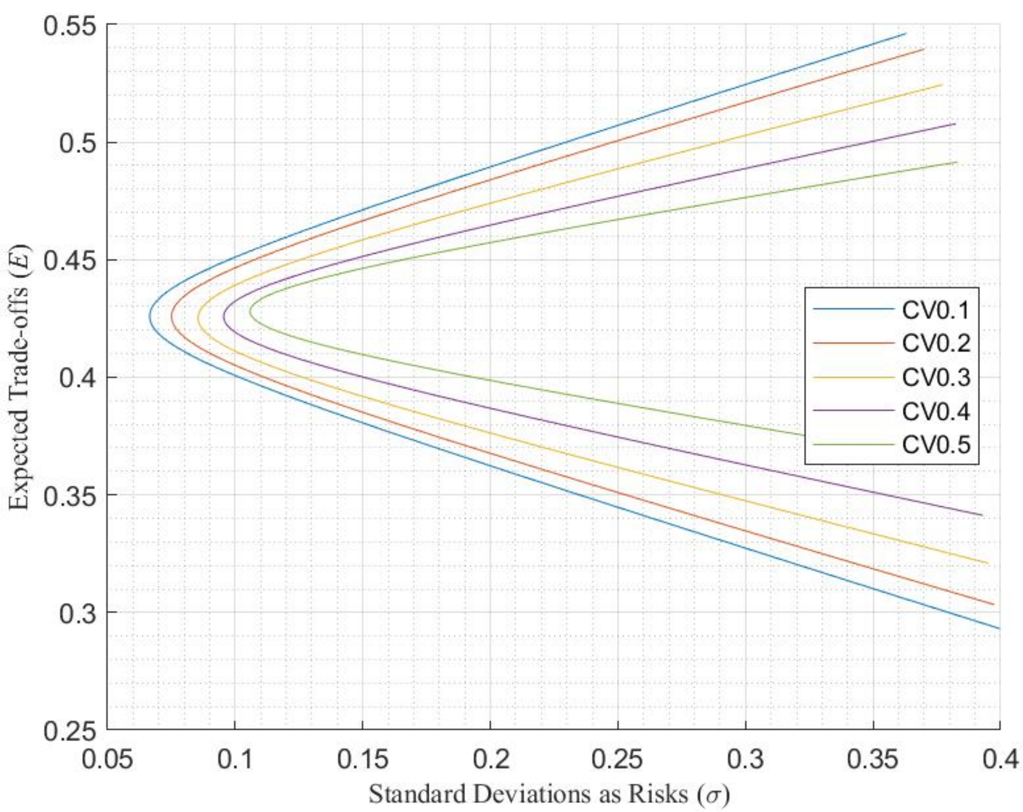


Figure 6: Efficient Portfolio Frontiers at CV Levels.

4.3.2 ToB heuristic facilitates multi-objective optimization

We can also use our ToB heuristic to facilitate the MPT model for multi-objective optimization. Our ToB heuristic generates 11 sequences for trade-off balancing with preference α = 0.0 : 0.1 : 1.0, respectively. We can consider these 11 sequences as portfolio assets to balance trade-offs between minimization of the first-order effect, i.e., minimizing the expected return (E) of the 11 sequences assets, and minimization of the second-order effect, i.e., minimizing the risk (σ). From Table 10, we can tell that, when we apply the MPT model to min(TCT) and min(VCT), the minimum E is 0.324 with the objective of min (E) and the minimum σ is 0.047 with the objective of min (σ). The value in Table 11 is the result that our ToB facilitates multi-objective optimization. The E is minimized from 0.324 in Table 10 to 0.213 in Table 11 and the σ is minimized a lot from 0.047 in Table 10 to 0.009 in Table 11, which means that, given the variation in processing times or fluctuation, the combination of good scheduling methods can achieve a better result on minimizing both first-order effect and second-order effect.

Table 10: Applying MPT to min (TCT) and min (VCT).

Objectives	min (E)	min (σ)
E	**.324**	.430
σ	.371	**.047**

Table 11: Applying MPT to 11 sequences generated by the ToB heuristic.

Objectives	min (E)	min (σ)
E	**.213**	.481
σ	.100	**.009**

4.4 Statistical process control (SPC) techniques

4.4.1 The X bar chart and the *R* chart

Given the inconsistency between TCT and VCT and the processing time uncertainties at 5 CV levels, we can use control charts to show fluctuations of process performance in terms of the $\bar{x}$ chart for sample means and the *R* chart for variation ranges according to our ToB(0.0, 0.4, 1.0) heuristics.

We use three process performances to plot control charts. In addition to two normalized deviations of TCT and VCT, the expected return for each instance and each sample is calculated by using equation (31), in which the weight between KPIs is set to $[w_1 \quad w_2] = [0.5 \quad 0.5]$. The three process performances are respectively averaged first across all 250 samples for 5 CV levels, and then across 50 instances for each of 6 job numbers. Statistics of control charts are provided in Table 12. From Table 12, we can see that, although ToB(0.0) and ToB(1.0) achieve the smallest grand mean of $\bar{x}$, R, lower control limit (LCL) and upper control limit (UCL) on TCT and VCT respectively, ToB(0.4) achieves the smallest expected return E of deviations from the MPT model.

Table 12: Statistics in control charts for TCT, VCT, and the expected returns in the MPT model according to ToB(0.0, 0.4, 1.0).

	$\bar{x}$			$LCL(\bar{x})$			$UCL(\bar{x})$		
	0.0	0.4	1.0	0.0	0.4	1.0	0.0	0.4	1.0
TCT	**.025**	.405	.743	**.015**	.383	.720	**.035**	.427	.765
VCT	.943	.201	**.054**	.919	.171	**.043**	.967	.232	**.065**
MPT	.484	**.303**	.398	.474	**.296**	.387	.494	**.310**	.409
	R			$LCL(R)$			$UCL(R)$		
	0.0	0.4	1.0	0.0	0.4	1.0	0.0	0.4	1.0
TCT	**.109**	.234	.238	**.062**	.132	.135	**.156**	.335	.342
VCT	.256	.321	**.119**	.145	.181	**.067**	.368	.461	**.171**
MPT	.106	**.075**	.115	.060	**.042**	.065	.152	**.107**	.166

The $\bar{x}$ charts and the R charts for normalized deviations for *TCT*, *VCT*, and the expected returns in the MPT model are plotted in Figure 7 to Figure 15, respectively. From these Figures, we can tell that ToB(0.0), ToB(0.4), and ToB(1.0) heuristics have some points out of control limits either in X-bar charts or in *R* charts for all three performance measures, which means that processing time uncertainties affect the inconsistency between min (*TCT*) and min (*VCT*).

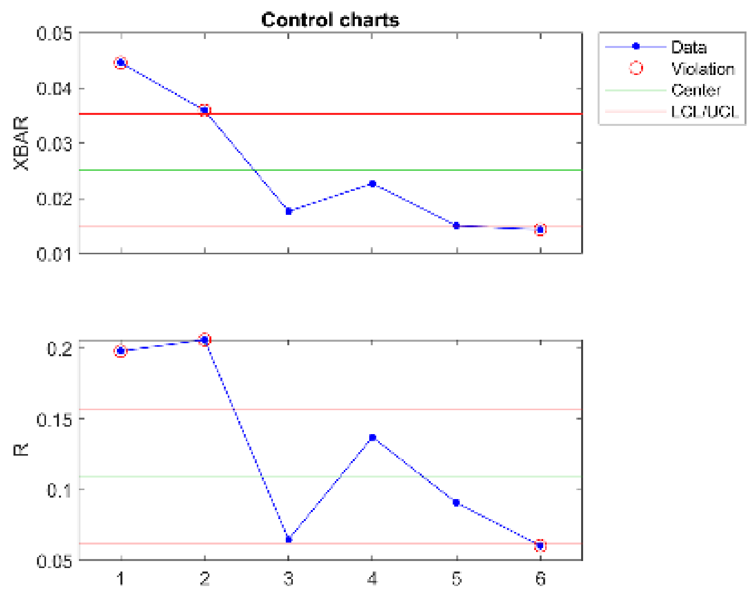


Figure 7: Control charts of *ND* for *TCT* generated by ToB(0.0).

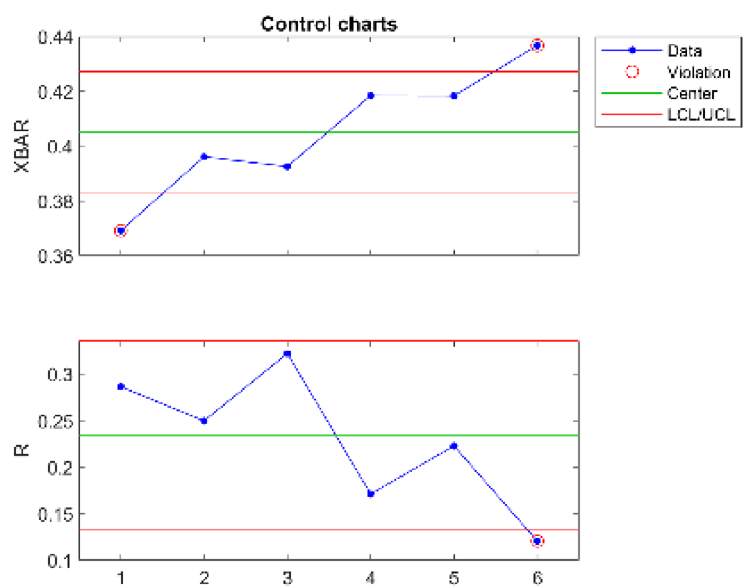


Figure 8: Control charts of *ND* for *TCT* generated by ToB(0.4).

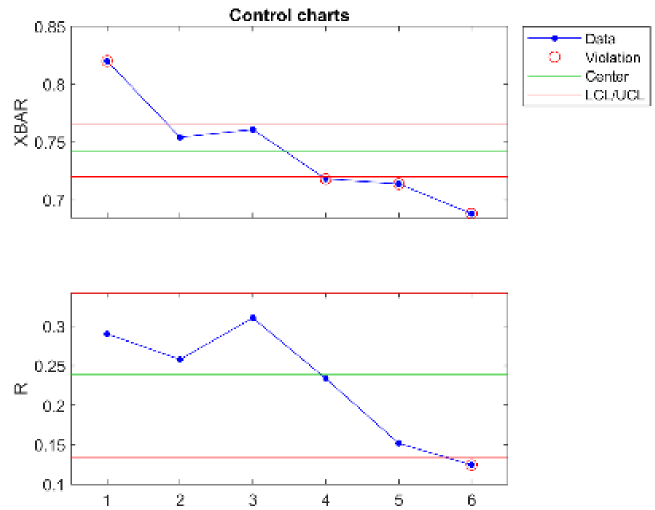


Figure 9: Control charts of *ND* for *TCT* generated by ToB(1.0).

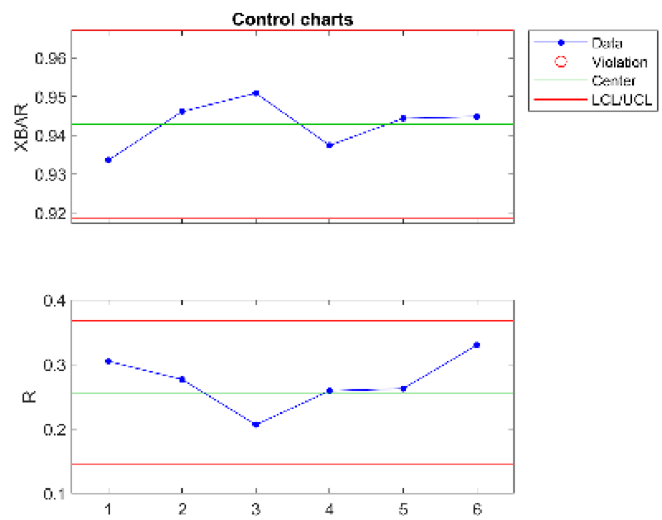


Figure 10: Control charts of *ND* for *VCT* generated by ToB(0.0).

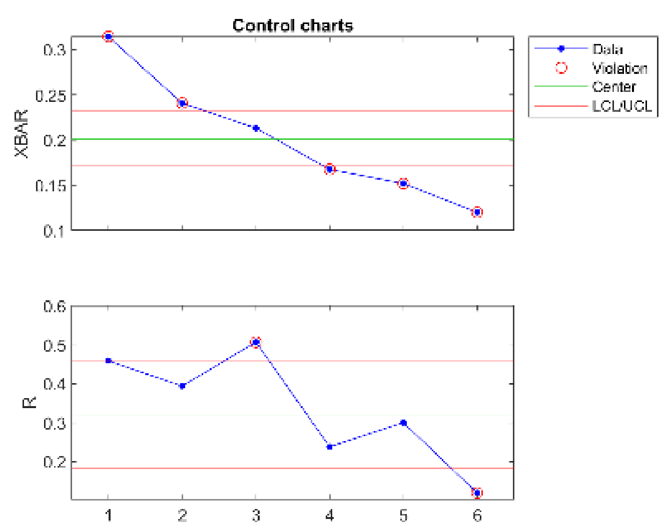


Figure 11: Control charts of *ND* for *VCT* generated by ToB(0.4).

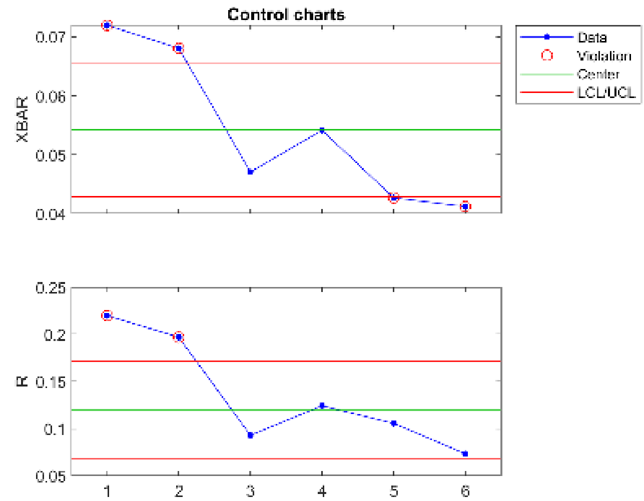


Figure 12: Control charts of *ND* for *VCT* generated by ToB(1.0).

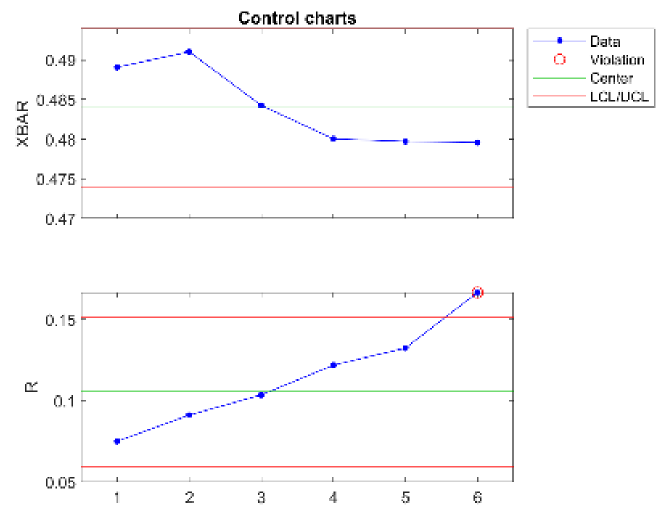


Figure 13: Control charts of *ND* for the expected returns in the MPT model generated by ToB(0.0).

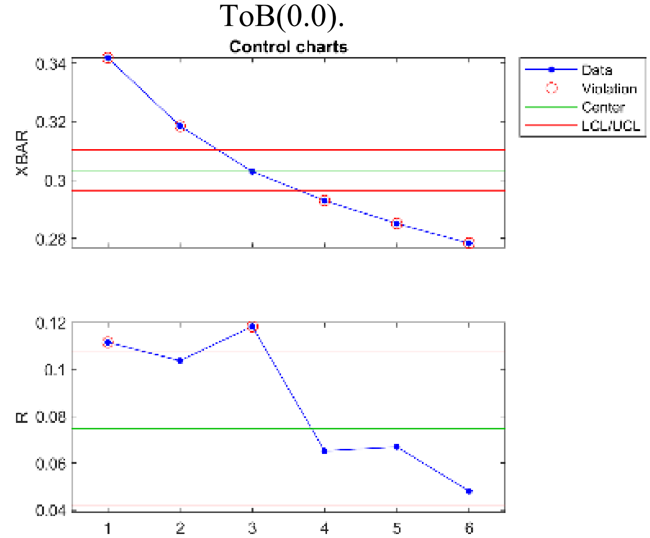


Figure 14: Control charts of *ND* for the expected returns in the MPT model generated by ToB(0.4).

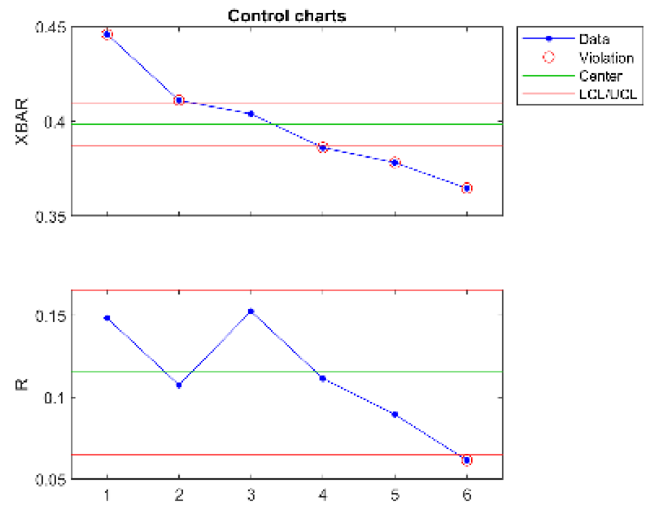


Figure 15: Control charts of *ND* for the expected returns in the MPT model generated by ToB(1.0).

4.4.2 Setting up specification limits

Besides control charts, two process capability indices of C_p and C_{pk} and probability to fit specification limits are also extensively used to verify if the process is under control. We can design reasonable specification limits to help decision-makers choose a heuristic when facing processing time uncertainty and trade-offs among KPIs.

We can design the lower specification limit (*LSL*) and upper specification limit (*USL*) by the performance of any heuristic. We just consider our ToB heuristic with 11 preferences since other heuristics are dominated by our ToB heuristic. The specification limits are usually independent from the process performance, being external from customers. Given 11 preferences for our ToB heuristic, we have 11 choices for setting up specification limits which are $[LSL_{h'}, USL_{h'}]$ for $h' \in \{ToB(0.0), \dots, ToB(1.0)\}$. Then we could check the probability of each h' to fit specification limits. But each heuristic achieves the highest probability to fit its own specification limits.

4.4.3 ToB heuristic facilitates SPC techniques

Through the analysis in Section 4.4.2, we carry out the following case studies to use our ToB heuristic facilitating SPC techniques. In our case study, all data were normalized finalizing normalized deviations vary between [0, 1], and for minimization problems, the smaller the value of the objective, the better the performance of a heuristic. So, the variation range of normalized deviations falls into [0, 0.5] is better than that fall into [0.5, 1]. We divide the range of [0, 0.5] into three equal intervals which are [0, 0.167], [0, 0.333], and [0, 0.500] and design the following three specification limits crossing all 11 sequences for both TCT and VCT at each CV level in Tables 13 and 14, respectively. The larger the probability to fit specification limits, C_p and C_{pk}, the better the process is under control, so the largest values will be highlighted in bold in the following tables. From Table 13, we can tell that ToB(0.0) heuristic achieves largest probabilities to fit specification limits [0, 0.167] and [0, 0.333] and largest C_{pk} values in the range of [0, 0.167] and [0, 0.333] at each CV level. However, ToB(0.4) heuristic achieves the largest probability to fit specification limits [0, 0.500] and the largest C_{pk} value in the range of [0, 0.500] at each CV level. The larger the probability to fit specification limits, the better the process is under control. According to equations (18), (19), and (20) C_{pk} measures the process centering. The larger the value of C_{pk}, the more centered the process is between its specification limits. Moreover, the negative values of C_{pk} mean that the process mean has drifted over either the LSL or the USL. As for C_p index, ToB(0.0) achieves largest C_p values from CV = 0.1 to CV = 0.4, but when CV level increase to 0.5, ToB(0.4) achieves largest C_p values in all three specification limits. When the specification limits enlarge, the probability of fit by ToB(0.4) increases. Equation (17) explains the reason why it is true in this case, where because (USL – LSL) is external and independent from the process performance. As long as $\hat{\sigma}$ is small, C_p is large. From Table 14, we find the same properties for probability to fit specification limits and similar properties for C_{pk} comparing ToB(0.4) with ToB(1.0). However, as for the C_p index, ToB(1.0) achieves the largest C_p values at all CV levels. From both Table 13 and Table 14, we can tell that as the CV level enlarges, the value of C_p increases correspondingly.

Table 13: Probability to fit specification limits, C_p and C_{pk} for TCT.

CV	[LSL, USL]	Prob.			C_p			C_{pk}		
		0.0	0.4	1.0	0.0	0.4	1.0	0.0	0.4	1.0
0.1	[.000, .167]	**.631**	.000	.000	**6.710**	.412	.361	**.112**	-1.114	-2.586
	[.000, .333]	**.631**	.192	.000	**13.421**	.824	.721	**.112**	-.290	-1.865
	[.000, .500]	.631	**.945**	.000	**20.131**	1.236	1.082	.112	**.534**	-1.144
0.2	[.000, .167]	**.642**	.000	.000	**1.363**	.419	.369	**.121**	-1.157	-2.614
	[.000, .333]	**.642**	.170	.000	**2.725**	.838	.738	**.121**	-.319	-1.875
	[.000, .500]	.642	**.941**	.000	**4.088**	1.258	1.108	.121	**.520**	-1.137
0.3	[.000, .167]	**.705**	.000	.000	**.939**	.449	.378	**.180**	-1.272	-2.638
	[.000, .333]	**.705**	.130	.000	**1.879**	0.897	.757	**.180**	-.375	-1.881
	[.000, .500]	.705	**.941**	.000	**2.818**	1.346	1.135	.180	**.522**	-1.124
0.4	[.000, .167]	**.779**	.000	.000	**.618**	.454	.389	**.259**	-1.336	-2.643
	[.000, .333]	**.781**	.099	.000	**1.236**	.908	.778	**.259**	-.428	-1.866
	[.000, .500]	.781	**.925**	.001	**1.853**	1.362	1.166	.259	**.479**	-1.088
0.5	[.000, .167]	**.829**	.000	.000	.478	**0.492**	.418	**.376**	-1.514	-2.739
	[.000, .333]	**.870**	.056	.000	.956	**.985**	0.837	**.376**	-.530	-1.903
	[.000, .500]	.870	**.914**	.001	1.434	**1.477**	1.255	.376	**.455**	-1.066

Table 14: Probability to fit specification limits, C_p and C_{pk} for VCT.

CV	$[LSL, USL]$	Prob.			C_p			C_{pk}		
		0.0	0.4	1.0	0.0	0.4	1.0	0.0	0.4	1.0
0.1	[.000, .167]	.000	.433	**.710**	.548	.238	**3.997**	-5.333	.013	**.184**
	[.000, .333]	.000	**.847**	.710	1.096	.477	**7.993**	-4.237	.464	**.184**
	[.000, .500]	.000	**.916**	.710	1.644	.715	**11.990**	-3.141	**.464**	.184
0.2	[.000, .167]	.000	.415	**.796**	.507	.252	**1.077**	-4.857	-.024	**.276**
	[.000, .333]	.000	**.869**	.796	1.014	.505	**2.155**	-3.842	**.480**	.276
	[.000, .500]	.000	**.942**	.796	1.522	.757	**3.232**	-2.828	**.529**	.276
0.3	[.000, .167]	.000	.359	**.907**	0.445	.284	**.767**	-4.179	-.101	**.443**
	[.000, .333]	.000	.897	**.908**	0.890	.567	**1.534**	-3.289	**.467**	.443
	[.000, .500]	.000	**.977**	.908	1.335	.851	**2.301**	-2.399	**.668**	.443
0.4	[.000, .167]	.000	.272	**.928**	.403	.279	**.601**	-3.674	-.190	**.583**
	[.000, .333]	.000	.853	**.960**	.807	.558	**1.202**	-2.867	.369	**.583**
	[.000, .500]	.000	**.985**	.960	1.210	.837	**1.803**	-2.060	**.748**	.583
0.5	[.000, .167]	.000	.179	**.821**	.379	.301	**.537**	-3.322	-.302	**.322**
	[.000, .333]	.000	.813	**.988**	.758	.602	**1.074**	-2.564	.300	**.752**
	[.000, .500]	.000	**.993**	.988	1.137	.902	**1.611**	-1.806	**.901**	.752

To better show our ToB(0.0, 0.4, 1.0) heuristics performance, we take probabilities of fit on the specification limits of [0.0, 0.5] at CV = 0.1 as an example to plot their process capability charts in Figure 16 to Figure 21. The probabilities to fit the specification limits are identical to those in Table 13 and Table 14, respectively.

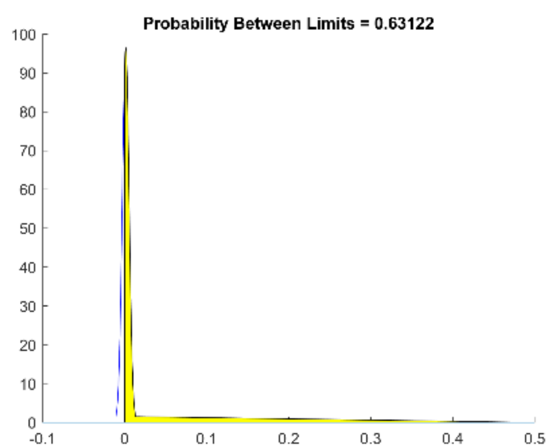


Figure 16: Probabilities of fit on [0.0, 0.5] of ToB(0.0) at CV = 0.1 for TCT.

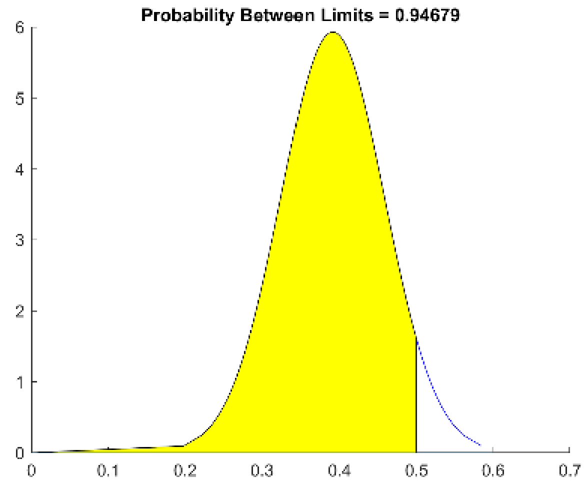


Figure 17: Probabilities of fit on [0.0, 0.5] of ToB(0.4) at $CV = 0.1$ for *TCT*.

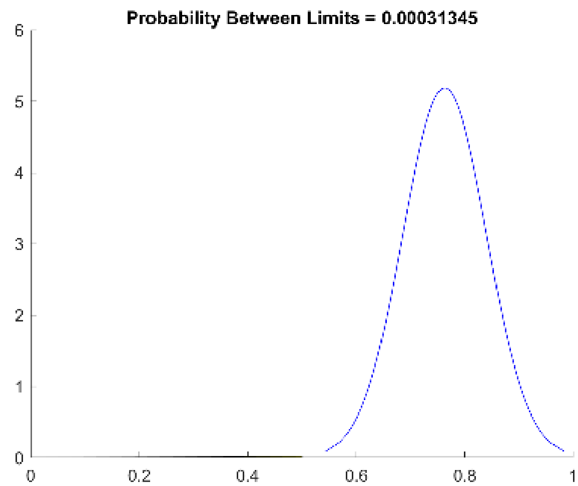


Figure 18: Probabilities of fit on [0.0, 0.5] of ToB(1.0) at $CV = 0.1$ for *TCT*.

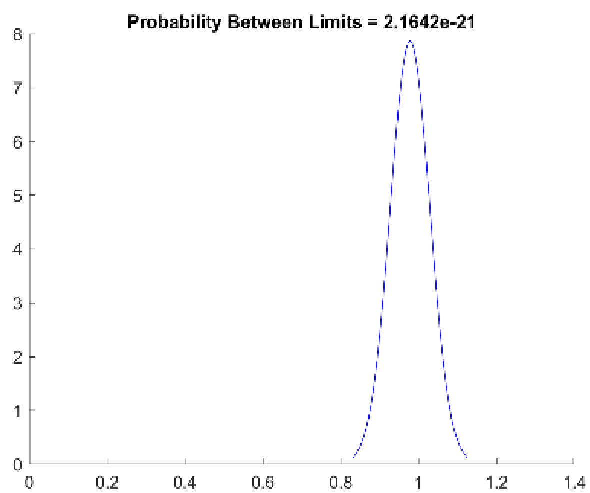


Figure 19: Probabilities of fit on [0.0, 0.5] of ToB(0.0) at $CV = 0.1$ for *VCT*.

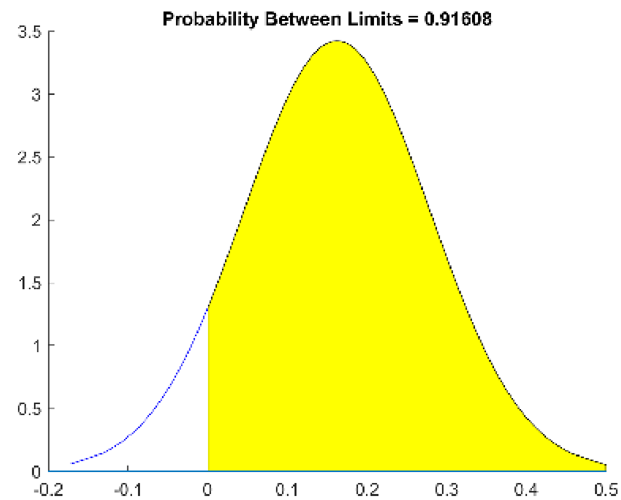


Figure 20: Probabilities of fit on [0.0, 0.5] of ToB(0.4) at $CV = 0.1$ for VCT.

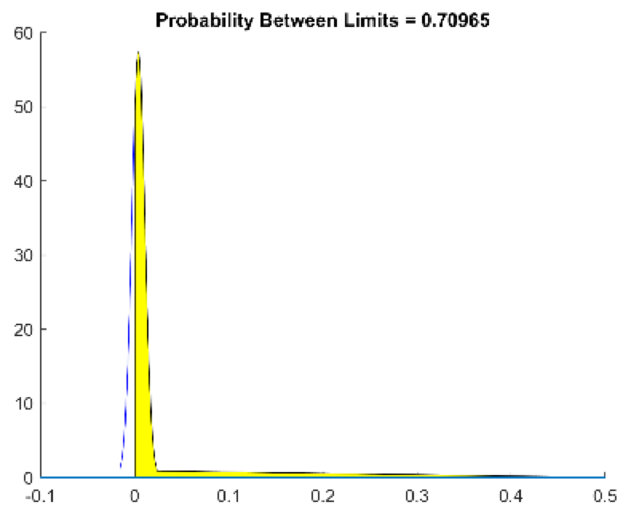


Figure 21: Probabilities of fit on [0.0, 0.5] of ToB(1.0) at $CV = 0.1$ for VCT.

Based on the result in Table 13 and Table 14 and the above analysis, we cannot choose a heuristic just based exclusively on C_p or C_{pk}, but on all three statistics. Due to the challenge of inconsistency among KPIs and the processing time uncertainty, we might have to relax the specification limits to keep process performance under control.

Overall, we can see that our ToB heuristic effectively addresses the above three challenges. First, our ToB heuristic dominates the other 5 heuristics, i.e., EP, ES, SEPT, LEPT, and FCFS methods for bi-objective optimization and balances the trade-offs between min (TCT) and min (VCT) and the trade-offs between the expected return and the risk through the results of our case studies. Besides, our ToB heuristic is useful for controlling stochastic production and gives insights for setting specification limits.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

Scheduling for one-stage production is important as a multi-stage production process can be modeled as one unit, which provides more insights into the whole process, especially when *NP*-complete or *NP*-hard problems are involved in multi-objective optimization for decision making. Because of the three challenges, i.e., (1) Challenge 1: inconsistencies among KPIs, (2) Challenge 2: inconsistencies between the expected return and the risk, and (3) Challenge 3: the variation in processing times, one-stage production scheduling is still challenging at both theory and application levels.

The following three problems arise from the three challenges in the practical application, respectively: (1) The scheduler may unconsciously make one KPI worse when optimizing the other objective. For example, if the scheduler in the medical system blindly pursues the improvement of hospital utilization, the patient flow time will decrease (Li et al., 2019). (2) When the decision-maker optimizes the expected return, such preference will bring about higher risk. (3) A great deal of uncertainties is in actual production, the most common manifestation is the variation of processing times, which makes it more difficult to address the first and the second aforementioned challenges.

In dealing with the three challenges to one-stage production scheduling, we propose a ToB heuristic with 11 preference α in sequencing. The concept of modern portfolio theory has been integrated into our heuristic development. We address the first challenge and the second challenge very well by using our ToB heuristic. For Challenge 1, we compared our ToB heuristic with five heuristics (ES, EP, SEPT, LEPT, and FCFS) based on 15000 samples (300 instances each of which has 50 samples). With the objective of min

(*TCT*), our ToB(0.0) heuristic outperformed the other five heuristics not only at individual *CV* level but also at grand *AND* across all *CV* levels with the value of only 0.025. With the objective of min (*VCT*), our ToB(0.5, …, 1.0) heuristic outperformed the other five heuristics distinctly. For Challenge 2, we excluded dominated heuristics and presented efficient portfolio frontiers at each *CV* level of our ToB heuristic. It was shown that our ToB heuristic was more flexible to reflect the effect of normalized deviations on portfolio returns and risks. With different *CV* levels, the minimum E and the minimum σ did not occur simultaneously, which gave us the insight that the fluctuation would affect the first- and second-order effects. It was also shown that, as the fluctuation increased, we needed to take a greater risk of σ to achieve the same value of E and vice versa. Additionally, our ToB heuristic facilitated multi-objective optimization very well when we applied the MPT model to 11 orders of our ToB heuristic, getting a smaller σ (0.002) than that (0.047) of applying the MPT model to KPIs. In dealing with Challenge 3, Daniels and Kouvelis (1995) proposed their scheme for stochastic production scheduling, which is to optimize the worst-case scenario, that is, to maximize the minimum deviation from the upper bound. Accordingly, they developed their EP and ES heuristics for min (*TCT*). Through the results of our case studies, in which variation in processing times is at five levels, we found that although the EP and ES heuristics provided good solutions to hedge against processing time uncertainty, their solutions were robust when $CV \leq 0.4$, our ToB (0.0) generated the smallest *AMND* of 0.425 at $CV = 0.5$, being more robust. And our ToB(0.0) heuristic outperformed the other heuristics not only at individual *CV* level but also at grand *AND* across all *CV* levels with the value of only 0.025, which means that our ToB heuristic outperformed the EP and ES heuristics not only on the worst-case scenario (*AMND*) but also on the expected averages (*AND*).

For application in industry, it is necessary to set up solid specification limits to control stochastic production, not only for addressing the process response of individual KPIs but also for addressing the three challenges. Since our ToB heuristic with 11 preferences provided solid solution space of each KPI as the preference of α spans all possible combinations of weights, it is useful to design the control limits of [LCL, UCL] for production scheduling, and to verify the specification limits of [LSL, USL] for customer service. What's more, undominated solution space is more accurate to reflect the process sensitivity to uncertainties. Results from our ToB heuristic facilitating SPC techniques illustrate that we need to take C_p, C_{pk}, and probability to fit the specification limits all together into account.

5.2 Future work

The next topic for our future research on production scheduling is adaptive production control. Variation in processing times is common in manufacturing. Consequently, the processing time is a random variable. From a formal point of view, a probabilistic - or random - variable is subject to a probability distribution and therefore unpredictable, which means it is impossible to know the value of a random variable at time t, no matter how accurately we have measured the past data up to time t–1. We plan to integrate the prediction of processing times into our ToB heuristic, as actual processing times unfold themselves in real-time. Accordingly, we can model the steady-state and the stability of a process and achieve adaptive production scheduling and control.

REFERENCES

Astrom, K. J. (1970). Introduction to Stochastic Control Theory. Dover Publications, Inc. Mineola, New York.

Aldowaisan, T., and Allahverdi, A. (2004). New heuristics for m-machine no-wait flowshop to minimize total completion time. *Omega*, 32(5), 345-352.

Bertolissi, E. (2000). Heuristic algorithm for scheduling in the no-wait flow-shop. *Journal of Materials Processing Technology*, 107(1-3), 459-465.

Bonney, M. C., and Gundry, S. W. (1976). Solutions to the Constrained Flowshop Sequencing Problem. *Operations Research Quarterly*, 27(4), 869-883.

Brucker, P. (2006). *Scheduling Algorithms*. springer.

Bagchi, U. (1989). Simultaneous Minimization of Mean and Variation of Flow Time and Waiting Time in Single Machine Systems. *Operations Research*, 37(1), 118-125.

Chen, C. L., Neppalli, R. V., and Aljaber, N. (1996). Genetic algorithms applied to the continuous flow shop problem. *Computers & Industrial Engineering*, 30(4), 919-929.

Campbell, H. G., Dudek, R. A., and Smith, M. L. (1970). A Heuristic Algorithm for the *n* Job, *m* Machine Sequencing Problem. *Management science*, 16(10), B-630-B-637.

Ciavotta, M., Minella, G., and Ruiz, R. (2013). Multi-objective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study. *European Journal of Operational Research*, 227(2), 301-313.

Chandrasekaran, S., Ponnambalam, S. G., Suresh, R. K., and Vijayakumar, N. (2007). Multi-objective particle swarm optimization algorithm for scheduling in flowshops to minimize makespan, total flowtime and completion time variance. *IEEE Congress on Evolutionary Computation*, 4012-4018.

Chiang, T. C., Cheng, H. C., and Fu, L. C. (2011). NNMA: An effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems. Expert Systems with Applications, 38(5), 5986-5999.

Dhingra, A., and Chandna, P. (2010). Multi-objective flow shop scheduling using hybrid simulated annealing. *Measuring Business Excellence*, 14(3), 30-41.

Duncan, J. A. (1959). *Quality Control and Industrial Statistics* (1st ed.). Richard D. Irwin, INC.

Daniels, R. L., and Kouvelis, P. (1995). Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science*, 41(2), 363-376.

Eilon, S., and Chowdhury, I. G. (1977). Minimising waiting time variance in the single machine problem. *Management Science*, 23(6), 567-575.

Framinan, J. M., Leisten, R., and Rajendran, C. (2003). Different initial sequences for the heuristic of Nawaz et al. (1983) to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. *International Journal of Production Research*, 41(1), 121-148.

Fernandez-Viagas, V., and Framinan, J. M. (2015). A new set of high-performing heuristics to minimise flowtime in permutation flowshops. *Computers & Operations Research*, 53, 68-80.

Gupta, J. N. D. (1972). Heuristic algorithms for multistage flowshop scheduling problem. *AIIE Transactions*, 4(1), 11-18.

Gupta, J. N. D., and Stafford, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3), 699-711.

Garey, M. R., Johnson, D. S., and Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2), 117-129.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 4, 287-326.

Gupta, S. K., and Kyparisis, J. (1987). Single machine scheduling research, *Omega*, 15(3), 207-227.

Hoogenveen, J. A., and Kawaguchi, T. (1999). Minimizing total completion time in a two-machine flowshop: Analysis of special cases. *Mathematics of Operations Research*, 24(4), 887-910.

Ho, J. C., and Chang, Y. L. (1991). A new heuristic for the n-job, M-machine flow-shop problem. *European Journal of Operational Research*, 52(2), 194-202.

Ishikawa, K. (1974). *Guide to quality control*. Asian Productivity Organization. Tokyo.

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

John, S. O. (2003). *Statistical Process Control*. Butterworth-Heinemann. Oxford, Boston.

Ke, L., Zhang, Q., and Battiti, R. (2014). Hybridization of decomposition and local search for multi-objective optimization. *IEEE Transactions on Cybernetics*, 44(10), 1808-1820.

Kanet, J. J. (1981). Minimizing variation of flow time in single machine systems. *Management Science*, 27(12), 1453-1459.

King, J. R., and Spachis, A. S. (1980). Heuristics for Flowshop Scheduling. *International Journal of Production Research*, 18, 343-357.

Kubiak, W. (1993). Completion time variance minimization on a single machine is difficult. *Operations Research Letters*, 14(1), 49-59.

Leavengood, S., and Reeb, J. E. (1999) Statistical Process Control, *Performance Excellence in the Wood Products Industry*, EM 8726.

Li, W., Mitchell, V. L., and Nault, B. R. (2014). Inconsistent objectives in operating room scheduling. *Institute of Industrial and Systems Engineers*, 727-736.

Li, W., Nault, B. R., and Ye, H. (2019). Trade-off balancing in scheduling for flow shop production and perioperative processes. *European Journal of Operational Research*, 273(3), 817-830.

Li, X., and Li, M. (2015). Multi-objective local search algorithm-based decomposition for multi-objective permutation flow shop scheduling problem. IEEE Transactions on Engineering Management, 62(4), 544-557.

Li, X., and Ma, S. (2016). Multi-Objective Memetic Search Algorithm for Multi-Objective Permutation Flow Shop Scheduling Problem. *IEEE Access*, 4, 2154-2165.

Merten, A. G., and Muller, M. E. (1972). Variance Minimization in Single Machine Sequencing Problems. *Management Science*, 18(9), 518-528.

Montgomery, D. C. (2009). *Introduction to Statistical Quality Control* (6th ed.). New Jersey: John Wiley & Sons.

Minella, G., Ruiz, R., and Ciavotta, M. (2011). Restarted Iterated Pareto Greedy algorithm for multi-objective flowshop scheduling problems. *Computers & Operations Research*, 38(11), 1521-1533.

Markowitz, H. M. (1952). *Portfolio Selection*. J Financ.

Miyazaki, S., Nishiyama, N., and Hashimoto, F. (1978). An adjacent pairwise approach to the mean flow-time scheduling problem. *Journal of Operations Research Society of Japan*, 21(2), 287-301.

Nawaz, M., Enscore, E. E., and Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1), 91-95.

Pinedo, M. L. (2016). *Scheduling: Theory, algorithms, and systems* (5th ed.). New York: Springer Science+Business Media, LLC.

Rajendran, C., and Chaudhuri, D. (1990). Heuristic algorithms for continuous flow-shop problem. *Naval Research Logistics*, 37(5), 695-705.

Rajendran, C., & Chaudhuri, D. (1992). An efficient heuristic approach to the scheduling of jobs in a flowshop. *European journal of operational research*, 61(3), 318-325.

Rajendran, C. (1993). Heuristic algorithm for scheduling in a flowshop to minimize total flowtime. *International Journal of Production Economics*, 29(1), 65-73.

Rajendran, C., and Ziegler, H. (1997). An effcient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs. *European Journal of Operational Research*, 103(1), 129-138.

Shabtay, D. and Steiner, G. (2007). A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, 155(13), 1643-1666.

Schrage, L. (1975). Minimizing the Time-in-System Variance for a Finite Jobset. *Management Science*, 21(5), 540-543.

Shewhart, W. A. (1931). *Economic control of quality of manufactured product* (1st ed.). Macmillan and Co., Limited. St. Martin's Street, London.

Taillard, E. (1993). Benchmarks of basic scheduling problems. *European Journal of Operation Research*, 64(2), 278-285.

Woo, H. S., and Yim, D. S. (1998). A heuristic algorithm for mean flowtime objective in flowshop scheduling. *Computers & Operations Research*, 25(3), 175-182.

VITA

NAME:

Jingjing You

EDUCATION:

Bachelor of Engineering	2011
Tianjin Medical University, Tianjin, China	

AWARDS:

North American Manufacturing Research Conference 49 Student Support Award	2021
Institute Scholarship Award (China)	2010
Institute Scholarship Award (China)	2009
University Scholarship Award (China)	2009
University Scholarship Award (China)	2008

PUBLICATIONS AND PRESENTATIONS

Li, W., Nault, B.R., **You, J.**, and Bilderback, B. Balancing trade-offs in one-stage production with processing time uncertainty. *49th North American Manufacturing Research Conference*.