8-3-2018

An Efficient Constructive Heuristic to Balance Trade-Offs Between Makespan and Flowtime in Permutation Flow Shop Scheduling

Feidi Dang
University of Kentucky, feidi.dang@outlook.com

Wei Li
University of Kentucky, wei.mike.li@uky.edu

Honghan Ye
University of Kentucky, honghan.ye@uky.edu

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 26 (2018) 40–48

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA

An efficient constructive heuristic to balance trade-offs between makespan and flowtime in permutation flow shop scheduling

Feidi Dang[a], Wei Li[a]*, Honghan Ye[a]

[a]*University of Kentucky, Lexington, KY, USA*

* Corresponding author. Tel.: +1-859-257-4842; fax: +1-859-257-3304.
E-mail address: wei.mike.li@uky.edu

Abstract

Balancing trade-offs between production cost and holding cost is critical for production and operations management. Utilization of a production line affects production cost, which relates to makespan, and work-in-process (WIP) inventories in a production line affect holding cost, which relate to flowtime. There are trade-offs between two objectives, to minimize makespan and to minimize flowtime. Without addressing trade-off balancing issues in flow shop scheduling, WIP inventories are still high in manufacturing, generating unnecessary holding cost. However, utilization is coupled with WIP inventories. Low WIP inventory levels might lower utilization and generate high production cost. Most existing constructive heuristics focus only on single-objective optimization. In the current literature, the NEH heuristic proposed by Nawaz, Enscore, and Ham (1983) is the best constructive heuristic to minimize makespan, and the LR heuristic proposed by Liu and Reeves (2001) is the best to minimize flowtime. In this paper, we propose a current and future deviation (CFD) heuristic to balance trade-offs between makespan and flowtime minimizations. Based on 5400 randomly generated instances, 120 instances in Taillard's benchmarks, and one-year historical records of operating room scheduling from University of Kentucky HealthCare (UKHC), our CFD heuristic outperforms the NEH and LR heuristics on trade-off balancing, and achieves the most stable performances from the perspective of statistical process control (SPC).

Peer-review under responsibility of the scientific committee of the 46th SME North American Manufacturing Research Conference.

Keywords: Permutation flow shop, trade-off balancing, constructive heuristic;

1. Introduction

Permutation flow shop production is widely used in automobile industry, such as assembly lines. The objective of production and operations management on flow shop production scheduling is to improve the

2351-9789
Peer-review under responsibility of the scientific committee of the 46th SME North American Manufacturing Research Conference.
10.1016/j.promfg.2018.07.005

efficiency of manufacturing system [1]. Minimization of makespan and minimization of flowtime are two fundamental criteria in flow shop scheduling, because many other performance measures are derived out from them, such as improving utilization of production lines, meeting due dates, reducing lateness or earliness, reducing work-in-process inventories, smoothing material flows in supply chains, etc.

Makespan is defined as the completion time at which the last job leaves the production line. Minimization of makespan suggests maximization of utilization, because utilization of one machine in a production line equals to the sum of processing times or workload divided by its makespan. Flowtime is defined as the total completion time of all jobs, and it affects WIP inventory levels. Production cost and holding cost directly relate to utilization of a production line and WIP inventory levels between machines, respectively [1, 2]. As both production cost and holding cost are important to production and operations management in manufacturing, production scheduling should minimize makespan and minimize flowtime simultaneously to achieve multi-objective optimization, especially for production planning in a long run.

Both minimization of makespan and minimization of flowtime are *NP*-complete for permutation flow shop production scheduling [3, 30]. Thus, it is difficult to obtain optimal solutions to a general *n*-job *m*-machine problem within acceptable computation time. Moreover, it has been proved that both minimizations are not consistent with each other [4], which means that minimizing one completion time does not necessarily minimize the other. Currently, few heuristics address such a relationship of inconsistency between minimizations of makespan and flowtime, and provide effective and efficient solutions to trade-off balancing in flow shop production scheduling.

Given the inconsistency between minimizations of makespan and flowtime, and to achieve stable production performance, we propose a current and future deviation (CFD) heuristic to balance trade-offs between makespan and flowtime minimizations in permutation flow shop scheduling. First of all, we derive out the lower and upper bounds of completion time for each job *j* on each machine *i*. Then, we calculate the deviations from lower bound to minimize the flow time and deviations from the upper bound to minimize the makespan. Consequently, in the initial sequence, we assign higher weights to current deviations generated by jobs in the head than those generated by jobs in the tail of the sequence. Furthermore, we adopt the insertion technique to improve the solution qualities.

The structure of the rest paper is organized as follows: in section 2, we provide the literature review about the existing heuristics for single objective and multi-objective. In section 3, the problem description is provided. The derivations of lower and upper bound and CFD heuristic are presented in section 4. Section 5 shows the results of computational experiment based on small-scale instances, classic Taillard's benchmark [5] and historical data from University of Kentucky HealthCare (UKHC). The conclusions and future work are discussed in section 6.

2. Literature Review

This section provides the literature review on permutation flow shop scheduling based on makespan minimization, $\min(C_{max})$, flowtime minimization, $\min(\sum C_j)$, multi-objective optimization. In general, there are two types of methods to generate the solutions for flow shop production scheduling: one is the exact methods and heuristics. For example, the branch and bound (B&B) method is a typical example of exact methods. However, it is extremely time consuming for exact methods to generate optimal solutions, and thus, it is impractical to use them even for medium-size problems. Therefore, constructive heuristics and/or meta-heuristics are preferred for production scheduling in industry. Literature review focuses on constructive heuristics in flow shop production scheduling, since the computation time of meta-heuristics is much longer than that of constructive heuristics, and both types of heuristics provide near-optimal solutions.

2.1 Review of makespan minimization objective

Minimization of makespan for permutation flow shop scheduling problem has been proved to be *NP*-complete for a *m*-machine flow shop [6]. From Johnson's algorithm [7], the optimal solution of makespan can be obtained with $O(n*log\ n)$ for two-machine flow shop. Campbell et al. proposed CDS heuristic [8], which *m* machines were regrouped as *m*-1 artificial two-machines flow shops. Then, apply the Johnson's algorithm to solve these *m*-1 two-machine flow shop problems, and the sequence with minimum makespan is selected as the final solution. In 1965, a heuristic is proposed by Palmer based on the concept

of 'slop index' [9], which the solution is generated by decreasing order of *SI*. Gupta [10] proposed a revised function of *SI*, and the author showed that the new proposed heuristic obtained better performance than Palmer's.

The famous NEH heuristic was proposed by Nawaz et al. in 1983 [11]. NEH heuristic has two different phases: initial sequence is generated by sorting jobs according to non-increasing order of total processing times on all machines. In second phase, select first two jobs from the initial sequence to create a partial sequence with minimum makespan value. Then, insert the next job one by one from initial sequence in orders into all possible locations of current partial sequence and select the partial sequence with minimum makespan. Repeat the insertion step until all jobs are removed from the initial sequence. In the work of Ruiz [12], they evaluated 25 existing heuristics and the results show that the NEH heuristic is the best heuristic for Taillard's benchmarks. Meanwhile, the frame of NEH heuristic has been applied in many existing heuristics for different objectives. Kalczynski and Kamburowski proved that NEH was the best constructive heuristic for permutation flow shop scheduling problem [13].

2.2 Review of flowtime minimization objective

Minimization of flowtime is also NP-complete for permutation flow shop scheduling [31], and has been studied for several decades. Ho and Chang [14] and Rajendran and Chauduri [15] proposed several different effective heuristics for flowtime objective.

In 1993, Rajendran proposed a heuristic to minimize the flowtime, named as Raj [16]. In this heuristic, the jobs are sequenced according to the ascending order of T_j, where $T_j = \sum_{i=1}^{m}(m-i+1)p_{j,i}$, where $p_{j,i}$ is the processing time of job j on machine i. Then select the first job as the partial sequence, and insert the rest job one by one into all possible location of the partial sequence. From the computational results, the Raj heuristic can obtain better solutions than heuristics proposed by Ho and Chang [14] and Rajendran and Chauduri [15].

WY heuristic, proposed by Woo and Yim [17], also applied the insertion strategy of NEH heuristic. The difference of WY heuristic is that the initial sequence is not required, which means the insertion phase has to be applied to each unscheduled job. Then the partial sequence with minimum flowtime is selected. According to the experiment result, the performance of WY is the best among CDS, NEH and Raj on mean flowtime objective.

In 2003, LF heuristic presented by Framinan [18] combined the insertion method of NEH and forward pair-wise exchange. The pair-wise exchange method was applied on the partial sequence that exchange any two jobs from insertion phase, and the new partial sequence is selected if a better performance is obtained. LF heuristic is better than WY on flowtime minimization objective.

In 2009, Laha and Sarin revised the LF heuristic, denote as LF-LS [19]. In this heuristic, the interchange method was modified, and the authors proved that the performance of LF heuristic is improved if the new exchange method was applied. However, for LF and LF-LS, the computational complexity is increased by one order because of the pair-wise step.

Liu and Reeves presented LR heuristic in their work [20]. An index function was developed, which considered the effect of idle time and the expect completion time of unscheduled jobs. The final solution was generated by sequencing jobs following ascending order of index function value. In their work, the author showed that LR heuristic outperformed existing heuristics, such as Ho and Chang [14] and WY [17]. From the literature, the LR is the best constructive heuristic to minimize flowtime with the computational complexity of $O(n^3m)$.

2.3 Review of multi-objective minimization

The heuristic proposed by Ho and Chang [14], they claimed that the performance of proposed heuristic is better than other existing heuristics on makespan, flowtime and total idle time minimization. Framinan et al. [21] developed a multi-objective heuristic to minimize the makespan and flowtime, and the NEH insertion method was applied. In this heuristic, a function $Y = w^* C_{max} (n/2) + (1\text{-}w)^* \sum C_j$ was developed, and the partial sequence with minimum Y is selected as current partial sequence. They compared the proposed heuristic with other existing heuristics, such as WY and R94 [22] and R95 [23]. The results show that the performance of the heuristic is better than others. However, in their work, Ho heuristic [24] can obtain better solutions than Framinan's heuristic when the flowtime minimization objective is given a large weight.

Furthermore, a lot of evolutionary algorithms were developed to solve the flow shop scheduling problem. For example, Varadharajan and Rajendran [25] applied the simulated annealing (SA) algorithm to minimize flowtime and makespan. Sayadi et al. [26] combine the firefly metaheuristic and local search method to solve the makespan minimization problem in permutation flow shop. However, the computation time of meta-heuristic is much longer than those of constructive heuristics.

For more details about trade-off balacning in flow sho shop scheduling and in manufacturing systems, we refer readers to [31, 32].

3. Problem description

In a permutation flow shop, n jobs must be processed on m machines, and follow the same order from the first machine to the last machine. Each machine can only process one job at the same time, pre-emption is not allowed, and setup times are included in processing times. In order to describe the problem, the following notation are used in this paper:

n	The number of jobs
m	The number of machines
$p_{j,i}$	The processing time of job j on machine i
$C_{j,i}$	The completion time of job j on machine i
$LB_{j,i}$	The lower bound of $C_{j,i}$.
$UB_{j,i}$	The upper bound of $C_{j,i}$.
$Dev^{UB}{}_{j,i}$	The deviation of $C_{j,i}$ from the $UB_{j,i}$
$Dev^{\mathrm{LB}}{}_{j,i}$	The deviation of $C_{j,i}$ from the $LB_{j,i}$

The calculation method of makespan and flowtime for permutation flow shop is discussed as follows. As the all jobs are prepared to be processed on first machine, there is no idle time on first machine and the completion time for each job on each machine can be generated by following equations:

$$C_{j,1} = \sum_{j=1}^{n} p_{j,1} \quad (1)$$

$$C_{1,i} = \sum_{i=1}^{m} p_{1,i} \quad (2)$$

$$C_{j,i} = max\{C_{j-1,i}, C_{j,i-1}\} + p_{j,i} \quad where\ j = 2 \dots n\ and\ i = 2 \dots m \quad (3)$$

Therefore, the makespan and flowtime can be calculated by:

$$C_{max} = C_{n,m} \quad (4)$$

$$\sum C_j = \sum_{j=1}^{n} C_{j,m} \quad (5)$$

4. CFD heuristic

The proposed CFD heuristic consists of two phases: the initial sequence is generated based on the deviations from lower bound and upper bound. In the second phase, we applied the insertion technique to further improve the solutions. In this section, we first introduce the calculation of lower and upper bounds, then the initial sequence generation is given, and at last, the procedure of the CFD heuristic is discussed.

4.1 Lower and upper bound generation

The sequence-independent lower and upper bounds for machine i are calculated based on the minimum and maximum idle time on machine i respectively. The minimum idle time (*minIT*) on machine i can be obtained by a fast flow from machine i-1 and a slow flow out of machine i. Moreover, the maximum idle time (*maxIT*) on machine i are generated by a slow flow from machine i-1 and a fast flow out of machine i. Therefore, the calculation method of minimum and maximum idle time is introduced as follows:

$$minIT_{j,1} = 0 \quad (6)$$

$$minIT_{j,i} = max\{LB_{j,i-1} - UB_{j-1,i}, 0\} \quad (7)$$

$$minIT_{j,i} = max\{LB_{j-1,i-1} + minIT_{j,i-1} + p^{SPT}_{j,i-1} - UB_{j-2,i} - minIT_{j-1,i} - p^{LPT}_{j,i-1}, 0\} \quad (8)$$

$$maxIT_{j,1} = 0 \quad (9)$$

$$maxIT_{j,i} = max\{UB_{j,i-1} - LB_{j-1,i}, 0\} \quad (10)$$

$$maxIT_{j,i} = max\{UB_{j-1,i-1} + maxIT_{j,i-1} + p^{LPT}_{j,i-1} - LB_{j-2,i} - max\,IT_{j-1,i} - p^{SPT}_{j,i-1}, 0\} \quad (11)$$

where $LB_{0,i} = LB_{j,0} = 0$ and $UB_{0,i} = UB_{j,0} = 0$. The $p^{LPT}_{j,i}$ and $p^{SPT}_{j,i}$ are the processing time of j^{th} job on machine i that follow the decreasing and increasing order of processing time of all jobs on machine i. In addition, according to the equation 6 to 11, the lower bound ($LB_{j,i}$) and upper bound ($UB_{j,i}$) can be computed by following equations:

$$LB_{j,i} = LB_{j-1,i} + minIT_{j,i} + p^{SPT}_{j,i} \quad (12)$$

$$UB_{j,i} = UB_{j-1,i} + maxIT_{j,i} + p_{j,i}^{LPT} \quad (13)$$

4.2 Initial sequence generation

In the CFD heuristic, the jobs are divided into two groups: scheduled job set (*S*) and unscheduled job set (*U*). Because the CFD heuristic aims to balance the trade-off between the makespan and flowtime, there are two different types of current and future deviation are developed: (1) For makespan objective, we minimize the deviation from upper bound, because it less likely generate idle time on previous machines; (2) For flowtime objective, we minimize the deviation from lower bound, which can generate small idle times on previous machines, depending on the value of completion time on previous machines. The steps of initial sequence generation are shown as follows:

Step 1:

Set location index k=1. Set $S = \emptyset$ and $U = \{J_1, J_2, \dots, J_n\}$.

Step 2:

Select the j^{th} job, denote as $J_{[j]}$ in U (j=1,…,n-k+1), and insert into k^{th} position of S. Then we calculate the average processing time ($AveP_i$) on each machine of the jobs in U except the $J_{[j]}$. We generated n-k artificial jobs with $AveP_i$ as the processing time of each artificial job on each machine. These artificial jobs are temporarily appended to S from $(k+1)^{th}$ to n^{th} in S.

Step 3:

Computed the completion times (C_{ji}) of $\{S\}$ by applying the equation (1) to (3). Then, the current and future deviations for each objective can be generated by following equations:

$$CuD_j^C = \sum_{i=1}^{m} (m-i+1) * (UB_{k,i} - C_{k,i}) \quad (14)$$

$$CuD_j^{\Sigma C} = \sum_{i=1}^{m} (m-i+1) * (C_{k,i} - LB_{k,i}) \quad (15)$$

$$FuD_j^C = \sum_{i=1}^{m} (m-i+1) \sum_{j=k+1}^{n} (UB_{j,i} - C_{j,i}) \quad (16)$$

$$FuD_j^{\Sigma C} = \sum_{i=1}^{m} (m-i+1) \sum_{j=k+1}^{n} (C_{j,i} - LB_{j,i}) \quad (17)$$

$$Dev_j^C = (n-k+1) * CuD_j^C + \frac{FuD_j^C}{(n-k)} \quad (18)$$

$$Dev_j^{\Sigma C} = (n-k+1) * CuD_j^{\Sigma C} + \frac{FuD_j^{\Sigma C}}{(n-k)} \quad (19)$$

The total deviation (TD) can be obtained by the following equation:

$$TD_j = \alpha Dev_j^C + (1-\alpha) Dev_j^{\Sigma C} \quad (20)$$

Where the α is the preference factor (α=0:0.1:1) for two objectives which is obtained from decision makers. Then the job $J_{[j]}$ with minimum value of total deviation (TD_j) will be selected and inserted to k^{th} location of S.

Step 4:

Remove the select job $J_{[j]}$ from the U. If k<n-1, set k=k+1 and go to step 2. If k=n-1, append the remaining job in U to S, and save the S as initial sequence $\{\pi\}$.

The reasons for the development of weighting factors (m-i+1 in equations 14-17, n-k+1 for current deviations and n-k for future deviations in equation 18-19) are explained as follows:

- From equations 14-17, the deviations generated on early machines have greater effects than those generated on later machines [27]. Therefore, the m-i+1 shows the decreasing effects as the machine number increases.
- Because the completion times of current job are fixed, we assign larger weight (n-k+1) on current deviations than future deviations [28, 29]. In addition, since the future deviation is generated by all unscheduled jobs, we divide future deviations by (n-k) to balance the effects between current and future deviations.

4.3 CFD heuristic

We use technique of insertion to further improve the initial solutions. Since our CFD heuristic is designed for balancing the trade-off between makespan and flowtime, we introduce the preference factor α into our insertion scheme. The steps of the CFD heuristic are presented as below:

Step 1:

Generate the initial sequence (π) using the initial sequence generation method from section 4.2.

Step 2:

Set k=2. Select the first two jobs from π to create a new k-jobs partial sequence $\{\phi\}$. Then exchange the position of these two jobs, and calculate the value of RIV in the following equations for two candidate partial sequences:

$$RIV = \alpha \cdot \left(\frac{C^{\{\phi\}} - LB_{k,m}}{UB_{k,m} - LB_{k,m}}\right) + (1-\alpha)\left(\frac{\Sigma C^{\{\phi\}} - \sum_{j=1}^{k} LB_{j,m}}{\sum_{j=1}^{k} UB_{j,m} - \sum_{j=1}^{k} LB_{j,m}}\right) \quad (21)$$

where $C^{\{\phi\}}$ and $\Sigma C^{\{\phi\}}$ are makespan and flowtime for $\{\phi\}$. The candidate partial sequence with minimum *RIV* is selected as current partial sequence $\{\phi\}$.

Step 3:

Set k=k+1, choose k^{th} job from initial sequence and insert to all k possible locations of $\{\phi\}$. Calculate the *RIV* value for k candidate sequences. Update the $\{\phi\}$ by the candidate sequence with minimum *RIV*.

Step 4:

If k<n, go to Step 3, otherwise output the current partial sequence $\{\phi\}$ as the final solution.

The computational complexity of our CFD heuristic is determined by the insertion phase in Step 3. Hence, the CFD heuristic has the same computational complexity as the NEH and LR heuristics, which is $O(n^3m)$.

5. Computational results

In the computational experiment, we compared our CFD heuristic with the NEH and LR heuristics on makespan (α=1) minimization, flowtime (α=0) minimization, and trade-off (α=0.5) minimization objectives based on random small-scale instances and Taillard's benchmark. Besides, we use the statistical process control to verify our CFD heuristic is better than the other two on historical OR data at UKHC.

5.1 Case Studies on Various Instances

We test our CFD heuristic on both small-scale and large-scale instances. The processing times for small-scale instances are randomly generated following the uniform distribution in [1, 99]. For small-scale instances, the number of jobs is 5, 6, 7, 8, 9, 10, and the number of machines is 3, 5, 7, 9, 11, 13, 15, 17, 19. Thus, there are 54 combinations. For each combination, 100 cases are randomly generated. Totally, we have 5400 instances for small-scale.

For large-scale instances, classic Taillard's benchmarks are used to test the performances of heuristics for flow shop scheduling, consisting of 120 instances in 12 combinations, where the number of jobs is 20, 50, 100, 200 or 500, and the number of machines is 5, 10 or 20. In each combination, there are 10 instances.

We have three criteria to evaluate the performances of heuristics in the following equations:

- Average relative percent deviation (ARPD) for makespan:

$$\text{ARPD}_{\text{C}} = \frac{1}{N}\left(\sum_{i=1}^{N}\left(\frac{C_i - MinC_i}{MinC_i}\right)\right) \times 100$$

- Average relative percent deviation (ARPD) for flowtime:

$$ARPD_{\Sigma C} = \frac{1}{N}\left(\sum_{i=1}^{N}\left(\frac{SC_i - MinSC_i}{MinSC_i}\right)\right) \times 100$$

- Trade-off value:

$$TO\ (\beta) = \frac{1}{N}\left(\beta \times \sum_{i=1}^{N}\left(\frac{C_i - MinC_i}{MinC_i}\right) + (1-\beta) \times \sum_{i=1}^{N}\left(\frac{SC_i - MinSC_i}{MinSC_i}\right)\right) \times 100$$

where the C_i and SC_i are makespan and flowtime for i^{th} instance. For small cases, $MinC_i$ and $MinSC_i$ are optimal solutions obtained by enumeration method. For Taillard's benchmark, $MinC_i$ and $MinSC_i$ are the best solutions for the i^{th} instance generated from three compared heuristics. N is the number of instances for each combination. N is 100 for small-scale instances but 10 for large-scale instances. β is the preference factor to evaluate the trade-off value, changing from 0 to 1 with the step of 0.1.

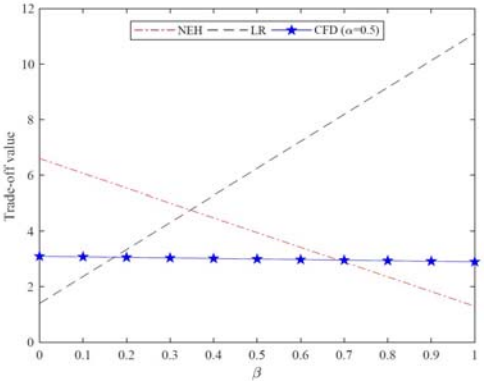
Table 1. ARPDs (%) of three heuristics for makespan and flowtime (S for small-scale and L for large-scale)

	CFD		NEH		LR	
ARPD	S	L	S	L	S	L
C_{max} (α=1)	**1.3**	3.5	1.3	**3.3**	11.1	12.5
ΣC_j (α=0)	**1.1**	1.2	6.6	8.2	1.4	**0.5**

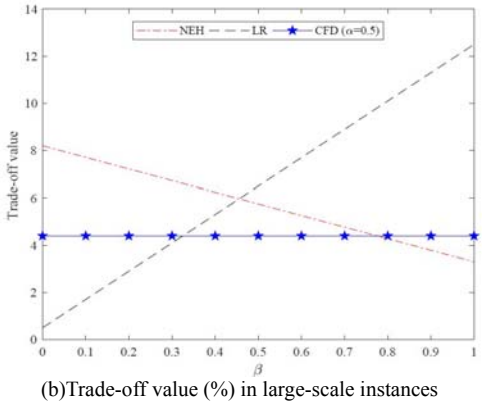
From the Table 1, for small-scale instances, we can see that the CFD heuristic has the smallest ARPD of 1.3% on makespan minimization, which is same as the NEH heuristic. For flowtime minimization, the CFD generates the smallest ARPD of 1.1%, better than the LR heuristic's 1.4%.

In large-scale instances, the CFD heuristic can obtain 3.5% of ARPD on makespan minimization objective, slightly larger than NEH heuristic of 3.3%. The LR has the worst ARPD of 12.5% on the

makespan minimization. For flowtime minimization objective, the LR has the smallest ARPD of 0.5%. The CFD generates the ARPD of 1.2% while the NEH heuristic obtains the worst ARPD of 8.2%.



(a) Trade-off value (%) for small-scale instances



(b)Trade-off value (%) in large-scale instances

Fig.1. Trade-off value of the CFD (0.5), NEH and LR heuristics

In order to balance the trade-off between makespan and flowtime, we set α=0.5. As shown in Fig. 1, we compared CFD (α=0.5) with NEH and LR heuristics.

From Fig. 1(a), in small-scale instances, as the β value changes, the CFD (α=0.5) heuristic has the most stable performance and the smallest value of average trade-off value, which is 3.03, while the NEH and LR generate 3.92 and 6.27, respectively. In addition, when the β changes from 0.2 to 0.6, the performances of the NEH and LR heuristics are dominated by the CFD (α=0.5) heuristic on trade-off minimization objective.

From Fig. 1(b), in large-scale instances, the CFD (α=0.5) heuristic has the most stable performance and generates the minimum average trade-off value which is 4.45, while the NEH and LR generate 5.78 and 6.50, respectively. Furthermore, when β changes from 0.4 to 0.7, the performance of CFD (α=0.5) heuristic can dominate the other two heuristics.

5.2 Statistical Process Control

To validate our CFD (α=0.5) heuristic for operating room (OR) scheduling across the perioperative process, we carry out case studies on historical OR data from University of Kentucky HealthCare, which consists of around 30,000 cases in a year from 2013 to 2014. Excluding the data from the weekend and holidays, we have more than 28,000 cases in 260 days for a year. Utilization of the perioperative process and patient flow time across the perioperative process are used to evaluate performances of OR scheduling methods. The relative performances of the NEH, LR, CFD (α=0.5) and UKHC are provided in Table 2.

Table 2. Utilization (%) and Patient Flow (mins)

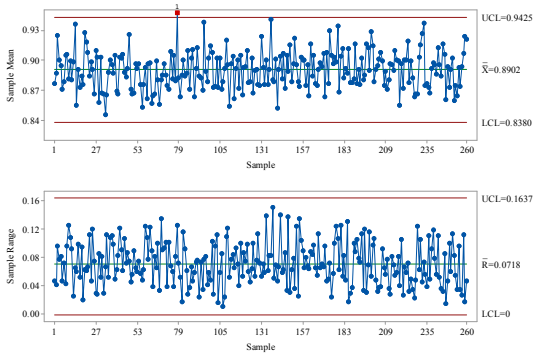
		NEH	LR	CFD (0.5)	UKHC
Util	Avg.	89.35	87.82	89.14	88.15
	StD	3.6	3.4	3.5	3.3
PF	Avg.	560.68	542.35	553.23	612.25
	StD	43.6	42.6	43.0	56.6

In Table 2, we can see that CFD (α=0.5) heuristic can achieve a perioperative process with the utilization of 89.14% higher than UKHC's 88.15%, with the patient flow time of 553.23 minutes lower than UKHC's 612.25 minutes.

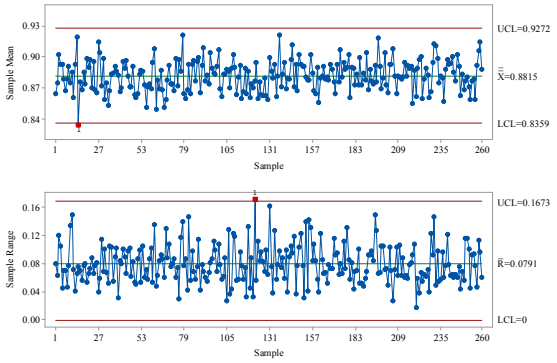
From the process utilization perspective in Fig. 2, we can see that a higher average utilization can be achieved by our CFD (α=0.5) heuristic. In the R-chart, the CFD (α=0.5) heuristic can generate small variation ranges without any points out of control limits. However, for the UKHC, there is a point above the upper control limit (UCL) in R-chart.

From the patient flow perspective in Fig. 3, we can see that our CFD (α=0.5) heuristic can achieve an average patient flow of 553.23 time units, which is smaller than that of 612.25 for UKHC. The improvement is (612.2-553.2)/553.2 = 10.66%, which indicates that potentially additional 2980 patients could be served in a year if our CFD (α=0.5) heuristic was used for OR scheduling. In the R-chart, the variation of patient flow achieved by our CFD (α=0.5) heuristic is smaller than that achieved by UKHC, in terms of a narrower range between upper and lower control limits for the CFD (α=0.5) heuristic than that for UKHC.

Such an improvement on the averages of performance measures for utilization and patient flow time indicate better trade-off balancing for OR scheduling, and small variation ranges suggest that the process performance is more stable, which release the burden of OR management.

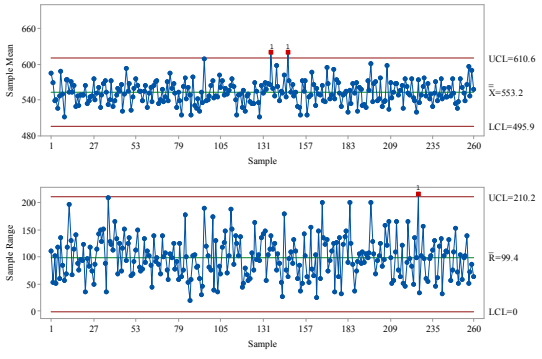


(a) Xbar-R chart of utilization values for CFD (0.5)

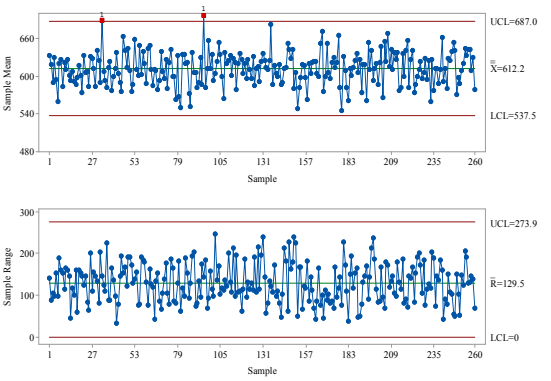


(b) Xbar-R chart of utilization values for UKHC

Fig.2. Xbar-R charts of utilization for CFD (0.5) and UKHC



(a) Xbar-R chart of Patient flow values for CFD (0.5)



(b) Xbar-R chart of Patient flow for UKHC

Fig.3. Xbar-R charts of patient flow for CFD (0.5) and UKHC

6. Conclusion

Permutation flow shop scheduling is widely applied in the industry. Generally, makespan and flowtime are related to utilization and work-in-process, respectively. Most existing heuristics focus on either makespan minimization or flowtime minimization. The minimization of makespan and flowtime has partially proved to be inconsistent. In order to balance the trade-off between makespan and flowtime, we proposed our CFD heuristic.

In the CFD heuristic, we first generate the lower and upper bounds. Then, we proposed an initial sequence generation method based on the deviations from lower or upper bounds. To further improve the solutions, we applied insertion method to the initial sequence. Through the computational experience, our CFD heuristic obtains the best performance of makespan, flowtime, and trade-off minimization objective on small-scale problems. For Taillard's benchmark, our CFD (α=0.5) heuristic has the best performance on trade-off balancing objective. Besides, using the statistical process control, our CFD (α=0.5) heuristic can achieve better performance than UKHC on the trade-off balancing.

In addition to trade-offs between completion times as performance measures based on averages, that is, the "first-order" effects, variation of performanc measures is also important for trade-off balancing to achieve sustainable operations management in manufacturing systems, based on the "second-order" effects, for example, completion time variance (CTV). Our future research will focus on balancing trade-offs

between the averages and their variance by using the CFD concept, especically about adaptive produciton control for stochastic problems.

Acknowledgements

This project was supported by Grant No. R03HS024633 from the Agency for Healthcare Research and Quality. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Agency for Healthcare Research and Quality. We also appreciate the support from UK HeathCare and Department of Mechanical Engineering at University of Kentucky.

References

[1] M. Pinedo, Scheduling. Springer, 2012.
[2] S. French, Sequencing and scheduling: an introduction to the mathematics of the job-shop. Ellis Horwood Ltd, Publisher, 1982.
[3] M. R. Garey, D.S. Johnson, R. Sethi, 1976. The complexity of flowshop and jobshop scheduling, Mathematics of Operations Research, 1 (2), 117-129.
[4] W. Li, V. L. Mitchell, B.R. Nault, 2004. Inconsistent Objectives in Operating Room Scheduling, in IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), 727-736.
[5] E. Taillard, 1993. Benchmarks for basic scheduling problems, European Journal of Operational Research, 64 (2), 278-285.
[6] G.K. Rand, Machine scheduling problems: Classification, complexity and computations, North-Holland, 1977.
[7] S. M. Johnson, 1954. Optimal two and three stage production schedules with setup times included, Naval Research Logistics (NRL), 1 (1), 61-68.
[8] H.G. Campbell, R. A. Dudek, M. L. Smith, 1970. A heuristic algorithm for the n job, m machine sequencing problem, Management Science, 16 (10), 630 -637.
[9] D. Palmer, 1965. Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum, Journal of the Operational Research Society, 16 (1), 101-107.
[10] J.N.D. Gupta, 1971. A functional heuristic algorithm for the flowshop scheduling problem, Journal of the Operational Research Society, 22(1), 39-47.
[11] M. Nawaz, E. E. Enscore, I. Ham, 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega, 11 (1), 91-95.
[12] R. Ruiz, C. Maroto, 2005. A comprehensive review and evaluation of permutation flowshop heuristics, European Journal of Operational Research, 165 (2), 479-494.
[13] P. J. Kalczynski, J. Kamburowski, 2007. On the NEH heuristic for minimizing the makespan in permutation flow shops, Omega, 35 (1), 53-60.
[14] J.C. Ho, Y.L. Chang, 1991. A new heuristic for the n-job, M-machine flow-shop problem, European Journal of Operational Research, 52 (2), 194-202.
[15] C. Rajendran, D. Chaudhuri, 1992. An efficient heuristic approach to the scheduling of jobs in a flowshop, European Journal of Operational Research, 61 (3), 318-325.
[16] C. Rajendran, 1993. Heuristic algorithm for scheduling in a flowshop to minimize total flowtime, International Journal of Production Economics, 29 (1), 65-73.
[17] H.S. Woo, D.S. Yim, 1998. A heuristic algorithm for mean flowtime objective in flowshop scheduling, Computers & Operations Research, 25 (3), 175-182.
[18] J.M. Framinan, R. Leisten, 2003. An efficient constructive heuristic for flowtime minimisation in permutation flow shops, Omega, 31 (4), 311-317.
[19] D. Laha, S. Sarin, 2009. A heuristic to minimize total flow time in permutation flow shop, Omega, 37 (3), 734-739.
[20] J. Liu, C.R. Reeves, 2001. Constructive and composite heuristic solutions to the P//∑ Ci scheduling problem, European Journal of Operational Research, 132 (2), 439-452.
[21] J.M., Framinan, R. Leisten, R. Ruiz-Usano, 2002. Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimisation. European Journal of Operational Research, 141(3), 559-569.
[22] C. Rajendran, 1994. A heuristic for scheduling in flowshop and flowline-based manufacturing cell with multi-criteria, The International Journal of Production Research, 32 (11), 2541-2558.
[23] C. Rajendran, 1995. Heuristics for scheduling in flowshop with multiple objectives, European Journal of Operational Research, 82 (3), 540-555.
[24] J.C. Ho, 1995. Flowshop sequencing with mean flowtime objective, European Journal of Operational Research, 81 (3), 571-578.
[25] T. Varadharajan, C. Rajendran, 2005. A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs, European Journal of Operational Research, 167 (3), 772-795.
[26] M.K. Sayadi, R. Ramezanian, N. Ghaffari-Nasab, 2010. A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems, International Journal of Industrial Engineering Computations, 1 (1), 1-10.
[27] W. Li, T. Freiheit, E. Miao, 2017. A lever concept integrated with simple rules for flow shop scheduling. International Journal of Production Research, 55 (11), 3110-3125.
[28] H. Ye, W. Li, A. Abedini, B.R. Nault, 2017. An effective and efficient heuristic for no-wait flow shop production to minimize total completion time, Computers & Industrial Engineering, 108, 57-69.
[29] H. Ye, W. Li, A. Abedini, 2017, An improved heuristic for no-wait flow shop to minimize makespan, Journal of Manufacturing Systems, 44, 273-279.
[30] J. Hoogeveen, T. Kawaguchi, 1999. Minimizing total completion time in a two-machine flowshop: analysis of special cases. Mathematics of Operations Research 24(4), 887-910.
[31] W.J. Hopp, M.L. Spearman, Factory Physics. McGraw-Hill: Boston, 2000.
[32] R. Y. Zhong, G.Q. Huang, Q.Y. Dai, T. Zhang, 2014. Mining SOTs and dispatching rules from RFID-enabled real-time shopfloor production data, Journal of Intelligent Manufacturing, 25 (4), 825-543.