Int J Adv Manuf Technol (2017) 88:1923–1936
DOI 10.1007/s00170-016-8879-7

ORIGINAL ARTICLE

Heuristics to minimize the completion time variance of jobs on a single machine and on identical parallel machines

Raju Rajkanth[1] · **Chandrasekharan Rajendran**[1] · **Hans Ziegler**[2]

Received: 5 July 2015 / Accepted: 5 May 2016 / Published online: 26 May 2016

Abstract This paper addresses the problem of scheduling n jobs on a single machine and on m identical parallel machines to minimize the completion time variance of jobs. This problem of scheduling jobs on parallel machines is motivated by a case study in an automobile ancillary unit. First, a heuristic to solve the single-machine scheduling problem is proposed. The parallel-machine scheduling problem is solved in two phases: job-allocation phase and job-sequencing phase. Two heuristics are proposed in the job-allocation phase, whereas in the job-scheduling phase, the single-machine scheduling approach is used. In this paper, both versions of parallel-machine scheduling problem (restricted and unrestricted) are considered. A good upper bound is obtained using a genetic algorithm, to evaluate the performance of the proposed heuristics for the parallel-machine scheduling problem. An extensive computation evaluation of the proposed heuristics is presented for both single-machine scheduling problem and the parallel-machine scheduling problem (especially considering the case study), along with the comparison of performances with the existing heuristics in the literature.

Keywords Single-machine scheduling · Parallel-machine scheduling · Completion time variance · Heuristics

1 Introduction

Given n jobs to be processed on a single machine, in the completion time variance (1||CTV) problem, the best sequence of the jobs is identified such that the CTV for this sequence is the minimum among all the possible sequences. The CTV problem is classified as a non-regular objective. A performance measure is called regular if it is non-decreasing in each of the job-completion times; otherwise it is called non-regular. Other non-regular performance measures include the mean squared deviation (MSD) of completion times from a given due date of jobs, the waiting time variance (WTV), and the flow time variance (FTV) problem. First, the 1||CTV problem is considered and then the case of an identical parallel machine is considered in this paper. The CTV minimization problem on identical parallel machines is denoted by Pm||CTV. The identical parallel-machine problem can further be classified into two: the restricted version and the unrestricted version. These two problems differ based on the presence of idle time before start time of first job on each machine. In the restricted problem all machines start at time zero and do not allow for the inserted time before jobs. This problem is denoted by Pm|Res|CTV. The unrestricted problem allows for the idle time to exist before the start time of the first job on a machine. This problem is denoted by Pm|Unres|CTV.

Our paper is motivated by a real-world case study in an automobile component manufacturer in India. The company follows the Just-In-Time approach to meet the buyer's requirement. The buyers have to be given a fair treatment in terms of order fulfillment by the manufacturer. Hence, the objective is to improve the customer-service rate by minimizing the

Electronic supplementary material The online version of this article (doi:10.1007/s00170-016-8879-7) contains supplementary material, which is available to authorized users.

✉ Chandrasekharan Rajendran
craj@iitm.ac.in

1 Department of Management Studies, Indian Institute of Technology Madras, Chennai 600 036, India

2 Department of Production and Logistics, Faculty for Business Administration and Economics, University of Passau, 94032 Passau, Germany

variance between order fulfillments of all the buyers in a given timeline. Hence the minimization of CTV is identified as the performance metric that can minimize variance and also bring uniformity in service across buyers. The major products manufactured by the manufacturing company (under study) are different types of dampers that are used to eliminate/minimize various kinds of vibrations that occur in internal combustion engines. Orders are unique in terms of both order quantity and product type. Each order from a buyer is treated as an unique job. The finished product has to undergo sequence of operations (e.g., turning, drilling, and milling, with CNC lathes present in the line). All the operations have been thoroughly analyzed and then the bottleneck operation and the corresponding work-center have been identified. This work-center has two identical parallel machines. Hence, keeping all these is mind, both literature and the corresponding theoretical research problem were explored. This research problem has led us to the problem of minimizing the CTV of jobs on parallel machines. In order to investigate this problem of scheduling on parallel machines, it is important to study the problem of scheduling on a single machine. Such a study of single-machine scheduling problem can give insight into scheduling on parallel machines. Therefore, in this paper the literature review and investigation concerning the scheduling of jobs on a single machine with the objective of minimizing the CTV of jobs is carried out. Subsequently, by using the findings of this investigation and by carrying out the literature review on the problem of scheduling on parallel machines, the problem of scheduling jobs on parallel machines with the objective of minimizing the CTV of jobs is ventured in this paper.

The measure of CTV with respect to the single-machine scheduling problem was first addressed by Merten and Muller [1] in the context of file organization problem in computing systems. Various properties of single machine CTV problem were proved by researchers in the past. Notably, the V-shape property (which means that the jobs before the job with shortest processing time should be scheduled in the non-increasing order of their processing times and the jobs after the smallest job should be scheduled in the non-decreasing order of their processing times) of optimal sequence was proved by Eilon and Chowdhury [2]. The position of the job with the largest process time in an optimal sequence was demonstrated by Schrage [3] and the author also made a conjecture about the position of next three largest jobs: the last, second, and third position respectively. Kanet [4] gave a counter example to disprove Schrage's conjecture about the scheduling position of the fourth largest job. Vani and Raghavachari [5] showed that Schrage's conjecture on the positions of the first three largest jobs is true for the 1||CTV problem. Further, Hall and Kubiak [6] verified Schrage's conjecture about the placement of the first three largest jobs. Kubiak [7] proved that the CTV minimization problem is NP-hard. Manna and Prasad [8] presented the bounds for the position of the smallest job in an optimal sequence for the 1||CTV problem. Many heuristics were proposed to derive a near-optimal schedule for the 1||CTV problem (see [2, 4, 5, 8–13]). Meta-heuristics such as genetic algorithm (see Srirangacharyulu and Srinivasan [13]), simulated annealing (Mittenthal, Raghavachari, and Rana [14]), and tabu search (Al-Turki, Fedjki, and Andijani [15]) were proposed to get good solutions for the 1||CTV problem. A few attempts are reported in the literature for solving the 1||CTV optimally using the exact methods such as the dynamic programming (De, Ghosh, and Wells [16]) and branch-and-bound technique (Viswanathkumar and Srinivasan [17]; Srinivasan and Srirangacharyulu [18]).

As for as the parallel-machine scheduling problem, Federgruen and Mosheiov [19] derived a lower bound for the identical parallel-machine variance minimization problem and proposed a heuristic, called alternating schedule heuristic. Chen, Li, and Sawhney [20] discussed various dominant properties for the Pm|Res|CTV problem, and presented a heuristic algorithm, which the authors referred to as balanced assignment verified spiral (BAVS), to solve the Pm|Res|CTV problem. Srirangacharyulu and Srinivasan [13] proposed heuristics for the single-machine CTV problem (SMH) and the parallel-machine CTV problem (called Multi-machine restricted heuristic (MMRH) and Multi-machine unrestricted heuristic (MMUH)). Li, Chen, and Sun [12] addressed the Pm|Unres|CTV problem and proved some dominant properties about an optimal solution to the problem. They also proposed a heuristic, which the authors referred to as WAVS for the unrestricted problem, and claimed that their heuristic solution is near-optimal for small-problem instances, and outperforms some existing algorithms for the large-sized problem instances. The allocation procedure of MMRH/MMUH is similar to that of the allocation procedure (referred to as balanced assignment in the heuristic BAVS) proposed by Chen, Li, and Sawhney [20]. Similarly the sequencing phase in BAVS is similar to that of WAVS. So, the BAVS heuristic is not considered in this paper, and only WAVS is considered in this study. In this paper, both restricted and unrestricted problems are addressed and hence we consider the heuristics proposed by Srirangacharyulu and Srinivasan [12] and Li, Chen, and Sun [13] in our work as benchmarks.

The single-machine and parallel-machine scheduling problems have been addressed more with respect to regular performance measures. Gokhale and Mathirajan [21] addressed a scheduling problem for minimizing total weighted flowtime, observed in automobile gear manufacturing. Specifically, the bottleneck operation of the pre-heat treatment stage of gear manufacturing process has been dealt with, in scheduling by the authors (for more information refer to Chen [22] and Lee, Lin, and Ying [23] for problems on multi-machine with regular measures of performance). A similar bottleneck problem is addressed in the case study presented in our paper, but with a non-regular performance measure.

Minimizing the CTV is an important measure when both earliness and lateness need to be penalized. Especially with the increasing interest in Just-In-Time (JIT) philosophy, the CTV problem has gained considerable attention in the scheduling area in recent times (Li, Chen, and Sun [12]). The CTV problem has applications in the manufacturing systems where it is essential to provide all jobs the same treatment, which could ensure fairness across different jobs.

There are good single-machine algorithms reported in the literature and these algorithms have been tested using good benchmark problem instances related to single-machine scheduling with CTV objective. Hence, the development of any new single-machine heuristic (such as ours) can be evaluated using the existing good heuristics and benchmark problem instances. If such a new heuristic performs better than the existing heuristics, then the extension of the idea to parallel-machine scheduling problem becomes logical and intuitively justifiable. Therefore, in this paper a new single-machine heuristic for minimizing the CTV has been developed. After proposing such a heuristic and showing its superiority over the existing heuristics, the extension of this single-machine heuristics to address the scheduling of jobs on the parallel machine with the objective of minimizing CTV has been ventured out in this paper.

This paper is organized as follows. Section 2 provides the 1||CTV problem definition and notations. Section 3 gives some existing results and lemmas for the 1||CTV problem. Section 4 presents the heuristic proposed in this paper to solve the 1||CTV problem and thereafter the computational analysis. Section 5 provides the Pm||CTV problem definition and notations, and presents some existing conjectures that are used in this paper to develop the proposed heuristics for the Pm||CTV problem. Section 5 also presents two proposed heuristics used to solve the Pm|Res|CTV and the Pm|Unres|CTV problems, followed by its computational evaluation (especially considering the case study). Finally, concluding remarks and the future directions are given in Section 6. A genetic algorithm is proposed for the sole purpose of obtaining a good upper bound (UB) on the CTV, so as to evaluate the goodness of solutions given by the proposed heuristics and the existing heuristics, and is presented in Appendix A. A numerical illustration of the proposed heuristic for the single-machine CTV problem is presented in Appendix B.

2 Single machine CTV problem

The assumptions made in solving the 1||CTV problem (also see Srirangacharyulu and Srinivasan [13]) are described as follows:

1. All jobs are available at time zero and the job-processing times are known in advance.
2. A machine can process only one job at a time.
3. No setup time exists between two consecutive jobs.
4. Preemption is not allowed.
5. The machine is available for processing at time zero.

Let

n	the total number of jobs to be scheduled
λ	a complete sequence
$C_j(\lambda)$	the completion time of job j in λ
$\overline{C}(\lambda)$	the mean completion time of n jobs in λ
J_j	job j
P_k	the process time of the job in position k in λ /*note: jobs $J_1, J_2, \ldots, J_n$ are numbered such that $P_1 \geq P_2 \geq P_3 \geq \ldots \geq P_n$, and hence the process times of jobs 1, 2, 3,…, n can be simply written as $P_1, P_2, P_3, \ldots, P_n$ corresponding to $J_1, J_2, J_3, \ldots, J_n$ */
CTV(λ)	the completion time variance of jobs, given the schedule λ

The CTV of n jobs on a single machine is given by

$$\mathrm{CTV}(\lambda) = \frac{1}{n}\sum_{j=1}^{n}\left(C_j(\lambda) - \overline{C}(\lambda)\right)^2, \tag{1}$$

where

$\overline{C}(\lambda) = \frac{1}{n}\sum_{j=1}^{n} C_j(\lambda)$, the mean completion time of jobs in sequence λ.

3 Preliminary results

Some important properties of the 1||CTV problem relevant to our study are presented in this section.

Property 1 (Eilon and Chowdhury [2]):

The optimal sequence that minimizes CTV is V-shaped.

Property 2 (Hall and Kubiak [6]):

In an optimal schedule, the largest job is scheduled in the first position, the second largest job is scheduled in the nth position and the third largest job is scheduled in the second position.

Property 3 (Merten and Muller [1]):

Every sequence and its dual have the same CTV. The dual of a schedule is obtained by reversing the order of last ($n-1$) jobs.

Property 4 (Srirangacharyulu and Srinivasan [13]):

In an optimal schedule of a 1||CTV problem, the jobs with the completion time less than the mean completion time are in the longest processing time (LPT) order and the jobs with start time greater than or equal to the mean completion time are in the shortest processing time (SPT) order.

Let the jobs be numbered such that $P_1 \geq P_2 \geq P_3 \geq \ldots \geq P_n$. Let λ be a given schedule and λ' be the schedule obtained by interchanging the two adjacent jobs in positions j and k in λ, where $k=j+1$.

Let $\Delta=\mathrm{CTV}(\lambda)-\mathrm{CTV}(\lambda')$.

Lemma 1 (Srirangacharyulu and Srinivasan [13]):

Let $C_j(\lambda)$ and $C_k(\lambda)$ be the completion time of jobs in position j and k respectively in λ such that $C_j(\lambda)$ and $C_k(\lambda)$ lie on either side of $\overline{C}(\lambda)$. Hence,

$$\Delta=\frac{(n-1)}{n}\delta^2-2\delta\left(\overline{C}(\lambda)-C_j(\lambda)\right), \tag{2}$$

where $\delta=(P_k-P_j)$, and $k>j$.

Lemma 2 (Srirangacharyulu and Srinivasan [13]):

Let λ be the given schedule and λ' be the schedule obtained by swapping the two jobs in positions j and k $(k>j)$ in λ. Hence, $\Delta'=\mathrm{CTV}(\lambda)-\mathrm{CTV}(\lambda')$, and is given by:

$$\Delta'=\frac{\delta}{n}\left\{2\left(C'_{j,k-1}(\lambda)-h\overline{C}(\lambda)\right)+h\delta-\frac{h^2\delta}{n}\right\},\ \text{and } h=(k-j), \tag{3}$$

where $C'_{j,k-1}(\lambda)=\sum_{i=j}^{k-1}C_i(\lambda)$, with $k>j$.

4 Proposed heuristic for the 1||CTV problem (SMH1)

In this section, a heuristic to solve the 1||CTV problem is proposed (referred to as SMH1 hereafter). For the purpose of easy understanding and presentation of the heuristic, the nomenclature of jobs J_1, J_2, …, and J_n, such that $P_1\geq P_2\geq P_3\geq\ldots\geq P_n$ is used.

4.1 Step-by-step procedure: SMH1 algorithm

Step 0 Let β be an arbitrary complete sequence following Property 1 and Property 2 (see Section 3) with $\mathrm{CTV}(\beta)$.

Step 1 Let π represent a partial sequence obtained by placing job J_1 in the first position of π, J_2 in the last position of π and job J_3 placed in the second position of π (based on Properties 1, 2, and 3 given in Section 2), i.e., $\pi=\{J_1-J_3-\ldots-J_2\}$.

Step 2 Do the following with respect to placement of job J_n in π in position $\left\lfloor\frac{n}{2}\right\rfloor$ and in at most three adjacent positions on either imminent side of position $\left\lfloor\frac{n}{2}\right\rfloor$; by placing J_n in these positions, at most seven possible sequences are generated.

Step 3 Consider one sequence at a time out of those generated in Step 2, do the following.

Step 3.1 Let $j=3$.

Step 3.2 Let $j=j+1$.

Step 3.3 Let π_1 and π_2 be the two partial sequences, derived from π by placing job J_j appropriately, be given as follows:

$$\pi_1=\{J_1-J_3-J_j-\ldots-J_n-\ldots-J_2\} and \pi_2=\{J_1-J_3-\ldots-J_n-\ldots-J_j-J_2\};$$

/*Note: when $j=4$, we have $\pi=\{J_1-J_3-\ldots-J_n-\ldots-J_2\}$, $\pi_1=\{J_1-J_3-J_4-\ldots-J_n-\ldots-J_2\}$ and $\pi_2=\{J_1-J_3-\ldots-J_n-\ldots-J_4-J_2\}$.*/

Step 3.4 Let σ_1 and σ_2 be two feasible sequences generated by placing jobs in the unscheduled positions of π_1 and π_2 as follows respectively. As for σ_1, a feasible complete sequence is generated by placing the unscheduled jobs with respect to π_1, taken one at a time, alternately in the unscheduled positions to the right extreme side and the left extreme side of J_n. This is done so in order to have a V-shaped property of the final full sequences. As for σ_2, a feasible complete sequence is generated by placing the unscheduled jobs with respect to π_2 (taken one at a time), alternately in the unscheduled positions to the left extreme side and the right extreme side of the partial sequence. For example, a ten-job problem with J_{10} placed in position six, we have σ_1 and σ_2 obtained from π_1 and π_2 respectively given as follows:

$$\sigma_1=\{J_1-J_3-J_4-J_6-J_8-J_{\mathbf{10}}-J_9-J_7-J_5-J_2\}$$

and

$$\sigma_2=\{J_1-J_3-J_5-J_7-J_9-J_{\mathbf{10}}-J_8-J_6-J_4-J_2\}$$

Step 3.5 Let $\mathrm{CTV}(\sigma_1)$ and $\mathrm{CTV}(\sigma_2)$ be the completion time variances of the two complete sequences σ_1 and σ_2.
If $\mathrm{CTV}(\sigma_1)<\mathrm{CTV}(\sigma_2)$
then
set $\pi=\pi_1$ and
set $\beta'\leftarrow\sigma_1$
else
set $\pi=\pi_2$ and
set $\beta'\leftarrow\sigma_2$

Step 3.6 If $\mathrm{CTV}(\beta')<\mathrm{CTV}(\beta)$,
then
set $\beta\leftarrow\beta'$ and
$\mathrm{CTV}(\beta)=\mathrm{CTV}(\beta')$.
/*e.g., a possible updated π appears as follows: $\{J_1-J_3-J_4-\ldots-J_n-\ldots-J_2\}$ with jobs

J_1, J_2, J_3, J_4, and J_n fixed in their respective positions.*/

Step 3.7 Go to Step 3.2 until $j = n - 2$. Place job $n - 1$ in the leftover position of β.

Step 4 Swap complimentary jobs k and j (see Kanet [4], for details) from sequence β to obtain β''.

/*Note: only complimentary pairs (complimentary pairs: two jobs (on on either side of J_n) when swapped, should result in a sequence, following the V-shaped property of the CTV problem) are chosen for swapping.*/

Step 5 Using Δ' from Lemma 2, if $\Delta' \leq 0$, then set $\beta = \beta''$ and CTV(β) = CTV(β'').

/*Note: if the mean completion time of β'' lies in-between the completion times of the complimentary-pair jobs j and k, use Δ from Lemma 1, and if $\Delta \leq 0$, then update β as β'' and evaluate the CTV accordingly.*/

Step 6 By repeating Steps 3, 4, and 5 for every sequence generated in Step 2, the final β and CTV(β) are obtained, yielding the solution of the proposed heuristic algorithm, SMH1.

A numerical illustration for the SMH1 is given in Appendix B.

4.2 Computational study for 1||CTV problem

The proposed heuristic is implemented using Visual C++ and executed on a personal computer with 4 GB RAM memory and with processor Intel(R) Core(TM)2 Duo CPU with 2.99 GHz. The performance of the proposed SMH1 is compared with some of the existing heuristics, namely EC1.1 and EC1.2 (Eilon and Chowdhury [2]), JJK (Kanet [4]), MP (Manna and Prasad [8]), SMH (Srirangacharyulu and Srinivasan [13] also used by them in their parallel-machine heuristic in the same paper) and VS (Ye, Li, Farley, and Xu [11]; also used by Li, Chen, and Sun [12] in their parallel-machine CTV problem) by using the benchmark problem instances presented by Srirangacharyulu and Srinivasan [13]. The results of performance evaluation comparing the heuristics for the single machine CTV problem are given in Table 1. The pseudo-polynomial algorithm presented by De, Gosh, and Wells [16] is used to obtain the optimal solution for these nine problem instances. The proposed heuristic SMH1 gives optimal solutions in five problem instances and with minimal deviation from the optimal solution for the remaining problem instances. The proposed heuristic (SMH1) in this paper has comparable CPU times with those of the existing heuristics, such as SMH and VS. Moreover, the CPU times for executing every one of the existing heuristics and our heuristic are very negligible (less than 1 s on the computer with afore mentioned specifications). Table 2 presents the comparison study of SMH1, SMH, and VS for some randomly generated problem instances (with number of jobs greater than 25). The branch-and-bound algorithm presented by Srinivasan and Srirangacharyulu [18] is used to obtain the optimal solution to these problem instances. The results show that the proposed heuristic performs better, when compared with the existing heuristics. Therefore, the proposed heuristic SMH1 is used hereafter in the present study, in respect of the parallel-machine scheduling problems.

5 Parallel-machine scheduling problem

In this paper, both the restricted and unrestricted versions of the multi-machine (identical parallel machine) CTV problem are considered. In the unrestricted case, the idle time is allowed before the start of each machine unlike the restricted case (see Srirangacharyulu and Srinivasan [13], for details).

5.1 Notations and assumptions

Assumptions in the problem formulations are made as given below (as given by [13]).

All jobs are available at time zero; job-processing times are known in advance; each machine can process only one job at a time; no setup time exists; preemption is not allowed; and all machines are available for processing at time zero.

Let

N	number of jobs to be scheduled
M	number of machines
Ω_i	set of jobs assigned to machine i; note: $\lvert\Omega_i\rvert = n_i^{'}$ and hence $\sum_{i=1}^{m} n_i^{'} = n$
λ_i	set of jobs allotted to machine i
λ^o	an optimal schedule
$ID_i(\lambda_i)$	inserted idle time before processing of jobs on machine i in λ_i /* holds in the case of unrestricted version */
$C_{ij}(\lambda_i)$	completion time of job j in λ_i, on machine i, $j = 1, 2, \ldots, n_i^{'}$
$\overline{\overline{C}}$	mean completion time of all jobs
$\overline{C}_i(\lambda_i)$	mean completion time of the jobs corresponding to those assigned to machine i in λ_i
$P_k(\lambda_i)$	process time of job k in the set of jobs allotted machine i

The problem is to find an optimum schedule to minimize the variance of job-completion times (for the restricted case), given by:

$$\mathrm{CTV}_r = \frac{1}{n}\sum_{i=1}^{m}\sum_{j\in\lambda_i}\left(C_{ij}(\lambda_i) - \overline{\overline{C}}\right)^2 \tag{4}$$

Table 1 Comparison of proposed heuristic SMH1 with the existing methods in the literature for the 1||CTV problem instances given by Srirangacharyulu and Srinivasan [13]

Problem	EC1.1	EC1.2	JJK	MP	SMH	VS	SMH1	Optimum
P(1)15	38,987.63	38,985.42	38,922.78	38,932.49	38,923.17	38,922.78	38,923.13	38,922.65[a]
P(2)15	20,153.71	20,120.11	20,102.51	20,103.36	20,102.51	20,102.51	20,102.38[a]	20,102.38[a]
P(3)15	29,238.49	29,218.86	29,217.56	29,217.56	29,217.56	29,217.56	29,217.09[a]	29,217.09[a]
P(4)15	32,604.78	32,595.29	32,551.71	32,553.66	32,551.71	32,551.71	32,551.32[a]	32,551.32[a]
P(5)20	64,399.43	64,399.42	64,343.54	64,343.54	64,344.33	64,343.55	64,341.63[a]	64,341.63[a]
P(6)20	51,831.66	51,828.79	51,739.55	51,739.55	51,739.24	51,739.55	51,737.55	51,736.99[a]
P(7)25	107,618.56	107,617.34	107,559.77	107,561.11	107,560.09	107,559.76	107,559.71	107,559.44[a]
P(8)25	67,388.08	67,388.09	67,359.22	67,359.27	67,359.23	67,359.23	67,358.88[a]	67,358.88[a]
P(9)25	91,102.48	91,080.87	91,019.21	91,019.60	91,020.15	91,019.20	91,019.78	91,018.42[a]

When we have executed the SMH (reported by [13]), we obtain the CTV as given in the table. Heuristics solution yielded by EC1.1, EC1.2, JJK, and MP are given in the table, as reported by [13]. Legends: EC1.1 and EC1.2 ([2]), JJK ([4]), MP ([8]), SMH ([13]), and VS ([11] also used by [12])

[a] Optimal solution

where

$$\overline{\overline{C}} = \frac{1}{n}\sum_{i=1}^{m}\sum_{j\in\lambda_i} C_{ij}(\lambda_i)$$

$$C_{ij}(\lambda_i) = \sum_{k=1}^{j} P_k(\lambda_i) \quad \forall j \le n_i^{'} \text{ and } i \le m$$

The problem is to find an optimum schedule to minimize the variance of job-completion times (for the unrestricted case), given by:

$$\mathrm{CTV}_u = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i^{'}}\left(C_{ij}(\lambda_i) - \overline{C}_i(\lambda_i)\right)^2 \tag{5}$$

where

$$\overline{C}_i(\lambda_i) = \frac{1}{n_i^{'}}\sum_{j\in\lambda_i} C_{ij}(\lambda_i), \text{ and}$$

$$C_{ij}(\lambda_i) = ID_i(\lambda_i) + \sum_{k=1}^{j} P_k(\lambda_i), \; j \le n_i^{'}$$

Table 2 Comparison of proposed heuristic SMH1 with SMH and VS for the randomly generated 1||CTV problem instances

Problem	SMH	VS	SMH1	Optimum
P(10)30	165,404.50	165,404.38	165,404.20	165,404.18[a]
P(11)30	123,921.51	123,921.17	123,920.51[a]	123,920.51[a]
P(12)35	208,382.33	208,382.14	208,382.19	208,382.05[a]
P(13)35	218,745.22	218,744.85	218,744.82	218,744.47[a]
P(14)40	224,741.84	224,741.99	224,741.55	224,741.18[a]
P(15)40	299,287.92	299,287.42	299,287.12	299,286.96[a]
P(16)45	277,054.72	277,053.24	277,052.75	277,052.47[a]
P(17)45	314,287.44	314,287.43	314,287.43	314,287.16[a]
P(18)50	495,310.01	495,309.94	495,309.58	495,309.56[a]
P(19)50	387,939.88	387,939.88	387,939.88	387,939.63[a]

Legends: SMH ([13]) and VS ([11]; also used by [12])

[a] Optimal solution

5.2 Preliminary results

Some conjectures are presented here on the basis of the review of literature. Most of these conjectures hold true for the unrestricted version of Pm||CTV (see Li, Chen, and Sun [12] and Srirangacharyulu and Srinivasan [13] for these conjectures). These are used in the proposed heuristics in this paper.

Conjecture 1: In an optimal schedule, the m longest jobs, namely, jobs $J_1, J_2, \ldots, J_m$ are each scheduled first on each of the m machines for the Pm|Unres|CTV problem.

Conjecture 2: In an optimal schedule, the job sequence on each of the m machine is V-shaped for both versions of the parallel-machine CTV problem.

Conjecture 3: Under λ^o, the schedule on each machine is optimal for both versions of the parallel-machine problem.

Conjecture 4: Under λ^o, the mean completion time on each machine is very close to each other in the restricted case and the same in the unrestricted case.

5.3 Proposed heuristic methods for the Pm||CTV problem

In this section, two heuristics are proposed for solving the parallel-machine scheduling problem for both restricted and unrestricted cases. The proposed pattern-based heuristics, namely, Secant Curve Heuristic (SCH) and Frog Curve Heuristic (FCH), are used to allocate the jobs to the machines. Once the assignment is done, the heuristic (SMH1) proposed for 1||CTV problem is used to schedule jobs allotted to each machine. The CTV of all jobs are evaluated without inserting idle times before the start of first job on each machine for the restricted case, and with the inserted idle times on machines before the first job for the unrestricted case. The allocation pattern in the SCH resembles a series of cup and an inverted cup alternatively, like the secant curve and hence the name, Secant Curve Heuristic. On the other hand, the FCH depicts the view of a frog and hence the name, Frog Curve Heuristic (FCH). The rationale behind the proposed heuristics is load balancing across all the machines in the first stage, followed by scheduling of jobs on each machine for the minimum CTV. Both these methods ensure that all machines get almost equal number of jobs to process. The allocation indirectly balances the load on each machine (because the jobs are arranged in the LPT order before the allocation cycle). Both the proposed heuristics (since SMH1 is used for phase 2 of SCH and FCH) are presented in the following section; hereafter SCH is referred to as SCH + SMH1 and FCH as FCH + SMH1 in this paper. Numerical illustrations of the SCH + SMH1 and FCH + SMH1 heuristics are given in Appendix C.

5.3.1 Step-by-step procedure: secant curve heuristic for the Pm||CTV problem, called SCH+ SMH1 (refer to Fig. 1)

Step 1 Arrange the jobs in the LPT order; call them J_1, J_2, …, J_n.

Step 2 In the first round, jobs J_1 to J_m are each assigned to machines 1 to m respectively; afterwards, jobs J_{m+1} to J_{2m} are each assigned to machines m to 1 respectively (this assignment pattern is illustrated in Fig. 1).

Step 3 In the second round, job J_{2m+1} to J_{3m} are each assigned to machines m to 1 respectively, and afterwards jobs J_{3m+1} to J_{4m} are each assigned to machines 1 to m.

Step 4 Repeat the pattern in Step 2 and Step 3 until all jobs are assigned to machines.

Step 5 Once the job-machine allocation is done, the jobs on each machine are scheduled using the proposed heuristic (SMH1) for the 1||CTV problem given in Section 4.1.

Step 6 Using the job-completion times, CTV is calculated, depending on the restricted and unrestricted versions of the parallel-machine problem.

5.3.2 Step-by-step procedure: frog curve heuristic for the Pm||CTV problem, called FCH+ SMH1 (refer to Fig. 2)

Step 1 Arrange the jobs in LPT order; call them J_1, J_2, …, J_n.

Step 2 Assign J_1 to J_m each in the first round, to machines 1 to m respectively, and J_{m+1} to J_{2m} each in the second round to machines 1 to m respectively in the same order.

Step 3 Assign J_{2m+1} to J_{3m} each in the first round to machines m to 1 respectively, and J_{3m+1} to J_{4m} each in the second round to machines m to 1 in the same order respectively.

Step 4 Repeat the pattern in Step 2 and Step 3 until all jobs are assigned to machines.

Step 5 This way of assignment resembles a frog (by shape) for each assigning cycle (the assignment pattern is illustrated in Fig. 2).

Step 6 Once the job-machine allocation is done, the jobs on each machine are scheduled using the proposed heuristic SMH1 in Section 4.1.

Step 7 Using job-completion times, CTV is calculated for all jobs, depending on the restricted and unrestricted versions of the parallel-machine problem.

5.4 Computational study for Pm||CTV problem

As indicated earlier, the present paper is motivated by a real-world case study. The manufacturing process of the dampers consists of two lines, Hub and Pulley. The semi-finished products from these two lines are later assembled to get the final finished product; refer to Sivasankaran and Shahabudeen [24] for detailed review of literature on balancing assembly line for parallel-machine scheduling problem. The sequences of operations involved in these two lines are in general: turning, drilling, CNC-Lathe, and broaching. The bottleneck operation is identified at the CNC-Lathe. Moreover, the CNC-Lathe workstation had two identical CNC-Lathe. Hence, the problem is solved using the Pm||CTV heuristics (considering both versions of the Pm||CTV problem), presented in this paper.

The proposed heuristics are compared with the existing heuristics proposed by Srirangacharyulu and Srinivasan [13] (called MMRH in the case of restricted version of the parallel–machine scheduling problem, and MMUH in the case of the unrestricted version of the parallel-machine scheduling problem – essentially both heuristics are the same except for the difference in the computation of the CTV) and Li, Chen, and Sun [12] (called WAVS in the case of unrestricted version of the parallel-machine scheduling problem). The existing heuristic by Srirangacharyulu and Srinivasan [13] employs a two-phase procedure: allocation phase involving the allocation of jobs to different machines followed by the sequencing phase involving the determination of sequence of jobs on a given machine. In the earlier section, the superior performance of our proposed sequencing heuristic (i.e., SMH1) in comparison to the sequencing heuristic by Srirangacharyulu and Srinivasan [13] (called SMH) has been proved. In view of this observation and in order to enhance the performance of MMRH/MMUH, the job-allocation phase is retained in the MMRH/MMUH, but SMH is replaced with SMH1 in the job-sequencing phase. This improved version of MMRH/MMUH is called MMRH1/MMUH1 in this paper. By the same logic, in view of the demonstration of the

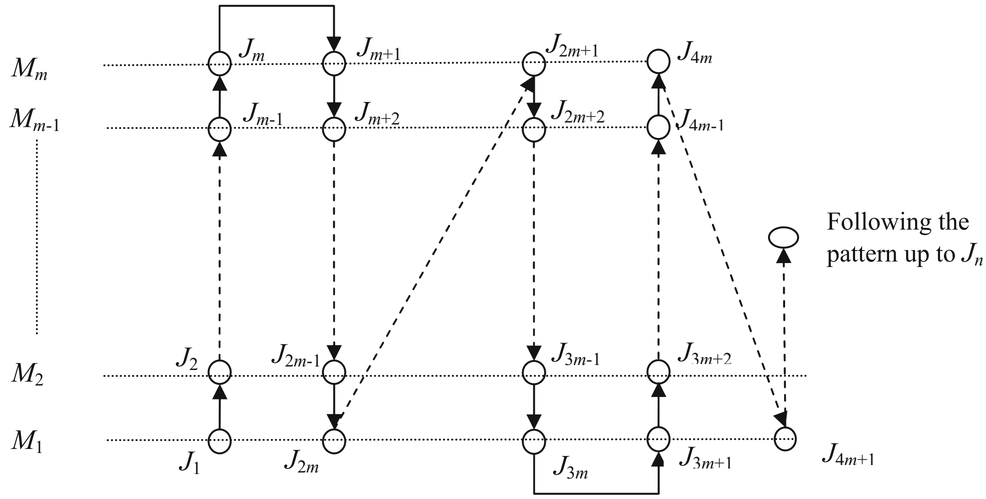

Fig. 1 Assignment pattern of SCH + SMH1 (the *arrow* shows the direction of sequential assignment of jobs (*Jn* refers to Job *n*) to machines (*Mm* refers to Machine *m*))

superior performance of SMH1 in comparison with the performance of the single-machine sequencing heuristic (called VS—also proposed by Ye et al. [11]) by Li, Chen, and Sun [12] (see Table 1 for details), the job-allocation phase is retained in the WAVS, but the VS method is replaced with SMH1 for the second phase. This improved version of WAVS is called WAVS1 in this paper.

The performance of all four methods is tested, by solving the real-world case involving 100 orders of four different types of damper. The order quantities are drawn from a uniform distribution with the order quantity, $OQ \in (100, 1000)$. The processing time at the bottleneck location is about 400 s (approx. 7 min). The best sequence in which these orders are taken up for production is derived using all four heuristics presented in the previous section. The SCH + SMH1 is found to give the best sequence for this case study. Moreover, 100 problem instances with different order quantities were simulated further to compare the methods. It is observed that all four heuristics are capable of giving the best sequence in more than one problem instance, though the MMRH + SMH1 appears to perform consistently better in most of the cases. The SCH + SMH1 and FCH + SMH1 give the best sequence in about 29 instances, whereas, the WAVS + SM1 give the best sequence in about 19 problem instances. The findings from the case study have motivated us to further explore the performance of the proposed heuristics across various sizes of the problem, i.e., varying the number of orders/jobs, number of machines, and processing time.

The performance of all four methods is tested by conducting experiments on two types of problem instances, namely: Type-1 and Type-2. The problem instances are classified in to Type-1 and Type-2 problems, based on the number of jobs and the number of machines. The problem instances are classified into two types to explore the sensitiveness of the proposed methods with respect to the problem size. All the experiments are carried out separately for both restricted and unrestricted versions of CTV problems. For problem instances of Type-1, we consider five different numbers of jobs: $n \in \{20, 40, 60, 80, \text{and } 100\}$. Furthermore, for each job size, we consider different number of machines: $m \in \{2, 4, 6, \text{and } 8\}$. For each combination of m and n, we generate 10 problem instances with processing times drawn from a uniform

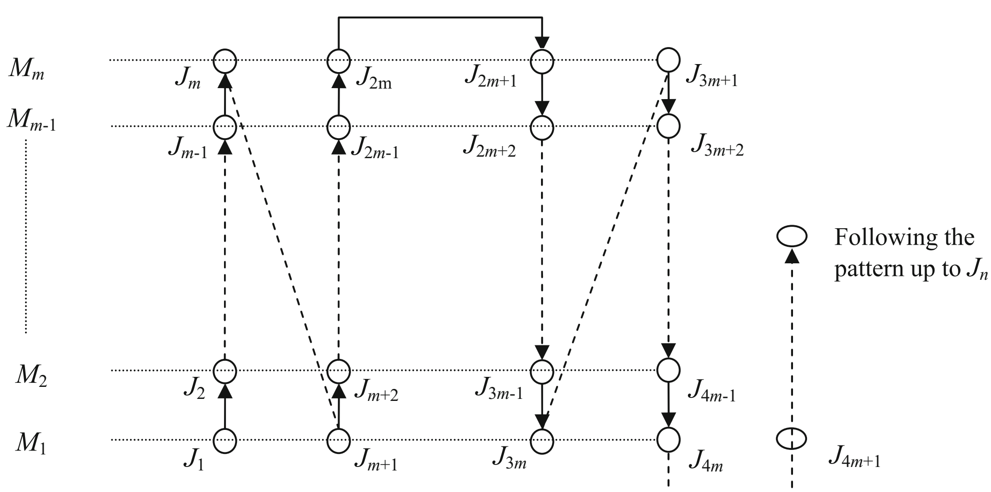

Fig. 2 Assignment pattern of FCH + SMH1 (the *arrow* shows the direction of sequential assignment of jobs (*Jn* refers to Job *n*) to machine (*Mm* refers to Machine *m*))

distribution with the job-processing times $P \in \{(1, 100), (1, 200), (1, 300), (1, 400), \text{and } (1, 500)\}$, respectively. Therefore, a total of 1000 problem instances are used to evaluate the heuristics under consideration using Type-1 problem instances. For problem instances of Type-2, we consider four different numbers of jobs: $n \in \{200, 300, 400, \text{and } 500\}$. For each problem instance, we consider different number of machines: $m \in \{10, 20, 30, 40, \text{and } 50\}$. For each combination of m and n, we generate 10 problem instances with processing times drawn from a uniform distribution with the job-processing times: $P \in \{(1, 100), (1, 200), (1, 300), (1, 400), \text{and } (1, 500)\}$. Altogether, a total of 1000 problem instances are used to evaluate the heuristics considering the Type-2 problem instances. Type-2 problem instances correspond to the large-sized problems, in comparison to Type-1 problem instances. Thus, these types of problem instances represent a variety of problem settings to evaluate the performance of heuristics under study. All parallel-machine heuristics under study require comparable and negligible computational effort (less than or equal to 1 s) to obtain a solution for all the problem instances. Hence, the computational time is not explicitly given in the tables.

5.4.1 Performance measures used for Pm||CTV problem

The two different performance measures used to evaluate the proposed methods are presented in this section. Since the CTV problem is NP-hard, the performance of all four heuristics is compared with a good upper bound (UB) obtained using the proposed hybrid multi-machine genetic algorithm (HMMGA) (see Appendix A for details). Note that the purpose of proposing HMMGA is solely for obtaining a good upper bound on CTV (i.e., UB). The main purpose of the present work is to present deterministic heuristics for both the versions of the parallel-machine scheduling problems. The performance measures used to compare all the four heuristic methods are discussed below,

(a) Mean competitive ratio (MCR):

The mean competitive ratio (presented by [12, 20]) is used to measure the performance of each of the four heuristics. The method with minimum competitive ratio is superior over the other methods on an average. The mean competitive ratio (MCR) is calculated as below:

$$\text{Mean competitive ratio (MCR)} = ((\mathrm{CTV}_i - \mathrm{CTV}_{\mathrm{UB}})/\mathrm{CTV}_{\mathrm{UB}} \times 100)/N, \quad (6)$$

where

N = Number of problem instances, and CTV_i is the CTV yielded by the ith heuristic under evaluation ($i=1$: SCH + SMH1, $i=2$: FCH + SMH1, $i=3$: WAVS1, $i=4$: MMRH1/MMUH1).

(b) Number of times the best known solution (NTBKS):

The number of problem instances where a heuristic provides the solution same as the best UB, given by HMMGA.

5.4.2 Type-1 problem (Pm|Res|CTV: Type-1 and Pm|Unres|CTV: Type-1)

The performance evaluation with respect to Type-1 problem instances for both versions of Pm||CTV problem (1000 problem instances each) is given in Table 3.

Table 3 presents the comparison of all the four heuristics with the UB obtained by using HMMGA for the Pm|Res|CTV: Type-1 problems. With respect to both measures, the MMRH1 performs better than the other heuristics for Type-1 problems of the restricted version. SCH + SMH1 fares second best for this type of problems based on both measures. Table 3 also presents the comparison all the four heuristics with the UB obtained by using HMMGA for Pm|Unres|CTV: Type-1 problems. Out of 1000 problems, with respect to the MCR performance measure, FCH + SMH1 perform better than the rest of the heuristics followed closely by WAVS1, which is the second best, based on the MCR measure of performance. MMUH1 performs better based on the NTBKS measure and followed by SCH + SMH1 as the second best for Type-1 problems of the unrestricted version. MMRH1 performs better for Type-1 problems (since the allocation phase in MMRH1 is based on balancing the load across each machines, this approach tends to work well for small-sized

Table 3 Comparison of the heuristics for Type-1 problems, using the HMMGA solution as the reference UB

Heuristic Method	Pm|Res|CTV: Type-1		Pm|Unres|CTV: Type-1	
	MCR	NTBKS	MCR	NTBKS
SCH + SMH1	1.4277[b]	114[b]	1.3014[d]	147[b]
FCH + SMH1	2.4143[d]	28[d]	0.3312[a]	11[d]
WAVS1	1.8062[c]	46[c]	0.3881[b]	126[c]
MMRH1/MMUH1	0.8257[a]	196[a]	0.8032[c]	181[a]

[a] Best performing heuristic with respect to given measure of performance is ranked from 1 to 4

[b] MCR: a lower value indicates better performance; NTBKS: a higher value indicates a better performance

[c] WAVS1 is the combination of allocation rule presented by [12] and the SMH1 heuristic propsoed in this paper

[d] MMRH1 is the combination of allocation rule presented by [13] and the SMH1 heuristic proposed in this paper as a sequencing rule, with respect to Pm|Res|CTV

[e] MMUH1 is the combination of allocation rule presented by [13] and the SMH1 heuristic proposed in this paper as a sequencing rule, with respect to Pm|Unres|CTV

problems); whereas, the other three heuristics attempt to assign an almost equal number of jobs to all machines.

5.4.3 Type-2 problem (Pm|Res|CTV: Type-2 and Pm|Unres|CTV: Type-2)

The performance evaluation with respect to Type-2 problem instances for both versions of Pm||CTV problem (1000 problem instances each) is given in Table 4.

Table 4 presents the comparison of all the four heuristics with the UB for the Pm|Res|CTV: Type-2 problems. The SCH + SMH1 performs better than all the other heuristics with respect to MCR and NTBKS. In fact, SCH + SMH1 gives the best solution for 503 problems out of 1000 problems, which is about 50 percent. The MCR values clearly indicate that SCH + SMH1 with 0.0648 is far better when compared with the other methods. FCH + SMH1 appears to performs better than MMRH1 and WAVS1 with respect to the MCR performance measure for the Pm|Res|CTV: Type-2 problems. MMRH1 gives 207 times the best solution, and is ranked the second best among the heuristics for the restricted version of Type-2 problems. Table 4 also compares the heuristics for Pm|Unres|CTV: Type-2 problems based on the final solution obtained by the heuristic methods. Out of 1000 problems, SCH + SMH1 gives the best solution for 466 problem instances (i.e., same as UB obtained by using HMMGA: see Appendix A). In fact, with respect to MCR, SCH + SMH1 clearly outperforms the rest of the heuristics and shows its superiority over the existing methods. MMUH1 performs fairly well and is second best based on the MCR performance measure and WAVS1 is the second best when the NTBKS measure is considered for the unrestricted version of Type-2 problems.

Table 4 Comparison of the heuristics for Type-2 problems, using the HMMGA solution as the reference UB

Heuristic Method	Pm|Res|CTV: Type-2		Pm|Unres|CTV: Type-2	
	MCR	NTBKS	MCR	NTBKS
SCH + SMH1	0.0648[a]	503[a]	0.0219[a]	466[a]
FCH + SMH1	0.0946[b]	87[c]	0.1159[d]	0[d]
WAVS1	0.3557[d]	73[d]	0.0606[c]	242[b]
MMRH1/MMUH1	0.1913[c]	207[b]	0.0571[b]	157[c]

[a] Best performing heuristic with respect to given measure of performance is ranked from 1 to 4

[b] MCR: a lower value indicates a better performance; NTBKS: a higher value indicates a better performance

[c] WAVS1 is the combination of allocation rule presented by [12] and the SMH1 heuristic proposed in this paper

[d] MMRH1 is the combination of allocation rule presented by [13] and the SMH1 heuristic proposed in this paper as a sequencing rule, with respect to Pm|Res|CTV

[e] MMUH1 is the combination of allocation rule presented by [13] and the SMH1 heuristic proposed in this paper as a sequencing rule, with respect to Pm|Unres|CTV

6 Conclusion

In this paper a new heuristic, called SMH1, is proposed to solve the single-machine CTV problem. Its performance is compared with the existing heuristics, and the proposed heuristic is shown to perform better than the existing ones. For the parallel-machine CTV problem, two new allocation procedures are proposed, namely, SCH + SMH1, and FCH + SMH1. The SMH1 is used for the sequencing phase in both the proposed heuristics for the Pm|Res|CTV and Pm|Unres|CTV problems. Another contribution of this paper is the proposal of a genetic algorithm (called Hybrid Multi-Machine Genetic Algorithm) that is used to obtain a good UB on CTV in both versions: restricted and unrestricted of the Pm||CTV problem. The computational testing has been done for both restricted and unrestricted versions using 2000 problems respectively. The MMRH1/MMUH1 heuristic (namely, the heuristic employing the allocation rule of Srirangacharyulu and Srinivasan [13] and the sequencing rule, SMH1 proposed in this paper) performs fairly good for Type-1 problem instances for the restricted version of the parallel-machine scheduling problem, followed, by SCH + SMH1 as the second best among the other heuristics. Similarly, for the unrestricted version FCH + SMH1 performs better based on the MCR measure. On the other hand, SCH + SMH1 outperforms the rest of the heuristics for Type-2 problem instances (i.e., large-sized problems) for both versions of the Pm||CTV problem based on the two performance measures (i.e., MCR and NTBKS).

This paper considers the identical case of parallel-machine CTV problem. The proposed heuristics for identical parallel machines can be modified and adapted to suit other non-regular measures like mean squared deviation (MSD) and flow time variance (FTV), and also to solve the non-identical parallel-machine problem in future research attempts.

7 Appendix A: hybrid multi-machine genetic algorithm (HMMGA) for Pm||CTV problem

A genetic algorithm is proposed for the sake of obtaining a good upper bound (UB) on the CTV, so as to evaluate the goodness of solutions given by the proposed heuristics and the existing heuristics, which are deterministic and computationally quick heuristic algorithm. The proposed HMMGA is used to solve the allocation phase of Pm||CTV problem. The sequencing phase is solved using SMH1. The main difference between the GA proposed by [13, 25] and the one presented in this work lies in the basic structure of the chromosome. The chromosome used by Srirangacharyulu and Srinivasan [13] has the sequence on each machine and also the number of jobs allotted to each machine. This way of representation uses more memory than our proposed GA because our chromosome consists of m genes less than their chromosome. All the offspring are subjected to repair mechanism throughout the GA procedure of Srirangacharyulu and Srinivasan [13], whereas in our GA, every generated offspring is always feasible (see Step 4 and Step 5, given below for details).

The four heuristics presented in Section 5.4 are used in the initial population of HMMGA. The UB from the HMMGA is used as the reference for evaluating the goodness of the proposed heuristics.

Step 1 Initialization:

In this paper, 16 parent chromosomes are randomly generated and four parent chromosomes are obtained from WAVS, SCH, FCH, and MMRH/MMUH heuristics respectively in the initial population.

Step 2 Representation:

In the HMMGA, the number of genes constituting each chromosome is equal to the number of jobs to be scheduled. The value of the ith gene of the chromosome represents the machine on which the job i is processed. For example, for a 14-job, 3-machine problem, a chromosome based on SCH would be as follows: Chromosome 1: {1, 2, 3, 3, 2, 1, 1, 2, 3, 3, 2, 1, 1, 2}.

Step 3 Fitness function:

Each string gives only the job-machine assignment and not the final schedule on each of the machines. The SMH1 is used to schedule the jobs on each machine. The completion time of job i on machine j (C_{ij}) is evaluated based on this schedule. The CTV of all jobs are evaluated, depending on the version of the parallel-machine CTV problem. The fitness of chromosome is given by $1/(1+Z)$, where Z is the CTV of the corresponding chromosome.

Step 4 Crossover operation:

The roulette wheel procedure is used to select chromosomes for the crossover operation. The single-point crossover operator is used with Variable Crossover Rate (VCR), i.e., depending on the current generation, the VCR varies. VCR is varied so that more parent chromosomes participate in the crossover operation in comparison to the offspring, as the generation progresses. The crossover is repeated for a number of times such that the number of new offspring produced is equal to the population size. Let l_c and k_c denote the step size and the crossover index respectively. The levels of crossover and the current generation with the VCR are related as follows:

$$\mathrm{VCR} = (1-(1+k_c)\times l_c) \quad \text{(A1.1)}$$

where $0<\mathrm{VCR}<1$, $l_c=0.05$, and

$$k_c = \left\lfloor \frac{\text{current generation}}{\text{total number of generations}} \times 4 \right\rfloor \quad \text{((A1.2))}$$

/*For example, if the current generation is 120, and with the total number of generations = 200, VCR is given as follows:

$$k_c = \left\lfloor \frac{120}{200} \times 4 \right\rfloor = 2;$$
$$\mathrm{VCR} = 1-(1+2)\times 0.05 = 0.85$$

It is evident that $0.75 \leq \mathrm{VCR} \leq 0.95$ holds for any given generation.*/

Note: since the single-point crossover operator is used, the new offspring contains only feasible job-machine combination.

For example, let Parent 1: {1, 2, 3, 3, 2, 1, 1, 2, 3, 3, 2, 1, 1, 2} and Parent 2: {1, 2, 3, 3, 2, 1, 3, 2, 1, 1, 2, 3, 3, 2} represent two parent chromosomes and Child 1 and Child 2 represent the offspring generated using the crossover operation. Let the randomly generated crossover point be 8, then:

Child 1: {1, 2, 3, 3, 2, 1, 1, 2, 1, 1, 2, 3, 3, 2} and Child 2: {1, 2, 3, 3, 2, 1, 3, 2, 3, 3, 2, 1, 1, 2}

It is observed that both offspring are feasible, and hence they do not need a repair mechanism.

Step 5 Mutation:

The gene-wise mutation is used in our HMMGA. The variable mutation rate (VMR) is used instead of regular mutation, i.e., based on the current generation, the VMR varies. Excited mutation (which promotes a higher amount of diversity in the population) is used, and this helps the heuristic to possibly come out of the local optimum. The excited mutation is invoked only if the on-hand best solution does not improve over the last fifty successive generations during the GA cycle. Excited mutation rate (EMR) is used to select chromosomes for excited mutation.

Let l_m and k_m denote the step size and the mutation index respectively. VMR is given as follows:

$$\mathrm{VMR} = l_m \times (1+k_m), \quad \text{(A1.3)}$$

where

$$\begin{aligned} 0 &< \mathrm{VMR} < 1 \\ l_m &= 0.05, \text{ and} \\ k_m &= \left\lfloor \frac{\text{current generation}}{\text{total number of generations}} \times 4 \right\rfloor \end{aligned} \quad \text{(A1.4)}$$

/*For the example given in Step 4, VMR is calculated as follows:

We have $l_m = 0.05$; $k_m = 2$; and hence, VMR = 0.15.

It is evident that 0.05 VMR 0.25 holds for any given generation.*/

/*Note: For a 14-job, 3-machine problem, consider the following sequence (based on SCH heuristic):

Chromosome: {1, 2, 3, **3**, 2, 1, 1, 2, 3, 3, 2, 1, 1, 2}.

Let the excited mutation rate (EMR) be 0.5, and consider the case when the on-hand best solution has not improved for the last fifty successive generations (say from, 51st generation to the 100th generation). All the chromosomes in the current population are subjected to the excited mutation, with EMR of 0.5. One gene is chosen at random from the selected chromosomes for mutation, i.e., let the 4th gene be the chosen one for mutation. The other values that gene 4 can take are machine 1 and machine 2. If, Gene 4 (J_4) is reassigned to machine 1 randomly, then the mutated chromosome is given as follows:

Mutated chromosome: {1, 2, 3, **1**, 2, 1, 1, 2, 3, 3, 2, 1, 1, 2}.

This procedure is repeated for all the remaining chromosomes in the population. */

Note: as long as each gene of the mutated chromosome is an integer between 1 and m, the feasibility of the chromosome will be maintained and this is ensured throughout the mutation operation.

Step 6 Elitism:

To ensure that the best chromosomes of a population are not lost during the selection strategy for the next generation, the best 10 chromosomes (out of the parent chromosomes and offspring) of the population are transferred to the next generation. This is done to ensure that the best chromosome is not lost during the transition from one generation to the next. This is referred to as elitism.

Step 7 Reproduction:

The size of the population is brought back to the population size at the end of every generation. For example, in our HMMGA, 10 chromosomes are selected based on elitism and out of the remaining chromosomes (consisting of the leftover parent chromosomes and offspring), 10 chromosomes are selected randomly for the next generation.

Step 8 Termination:

The HMMGA is terminated when either of the two conditions is satisfied:

- the process has reached the pre-defined limit with respect to the maximum number of generations (i.e., 1000);
- the best chromosome of the population has not shown an improvement for a number of successive generations, which is greater than or equal to the stagnation threshold (i.e., 500).

Upon termination, the best chromosome of the current population is presented as the heuristic solution of the problem, called UB to evaluate the performance of heuristic algorithms in the main work.

Parameters:

Parameters used in the HMMGA: population size = 20; number of generations = 1000; and excited mutation rate = 0.5 (all these are obtained after pilot runs). Details are not reported to save space and also because the primary purpose of the GA is to obtain a good UB on CTV to evaluate the deterministic heuristic solutions.

8 Appendix B: a numerical illustration of the proposed heuristic SMH1

The implementation of the SMH1 heuristic is explained with the help of a numerical example.

Consider the 10-job problem in Eilon and Chowdhury [2] with processing times of 19, 18, 16, 13, 10, 9, 8, 5, 2, and 1, respectively. Let the jobs be called $J_1, J_2, \ldots$, and J_{10}, such that $P_1 \geq P_2 \geq \ldots \geq P_n$.

Let an initial solution (for the purpose of obtaining the starting upper bound) be obtained from an arbitrary sequence given as follows: $\beta = \{1-3-4-8-9-10-7-6-5-2\}$ and CTV (β = 491.81) (see Step 0 in the Section 4.1). We have $\pi = \{1-3-\ldots-2\}$.

There are seven unscheduled positions in π. Do the following by placing job J_{10} (J_n) in positions $\left(\lfloor\frac{n}{2}\rfloor-2\right), \left(\lfloor\frac{n}{2}\rfloor-1\right), \left(\lfloor\frac{n}{2}\rfloor\right), \left(\lfloor\frac{n}{2}\rfloor+1\right), \left(\lfloor\frac{n}{2}\rfloor+2\right)$, and $\left(\lfloor\frac{n}{2}\rfloor+3\right)$, seven possible sequences are generated, and the steps presented below are repeated for each of these positions.

/*Note: When job J_{10} is placed in positions $\left(\lfloor\frac{n}{2}\rfloor-2\right)$, a complete feasible sequence is obtained, and hence Step 3 and Step 4 given in Section 4.1 are ignored*/

For the illustrative purpose, let us consider the following sequence with job J_{10} placed in position 6.

Let $j = 4$, i.e., job J_4 is placed in the first and the last unscheduled positions of π respectively to obtain π_1 and π_2,

π_1 and π_2 be the two partial sequences derived from π, and given as follows:

$\pi_1 = \{1-3-4---10----2\}$ and $\pi_2 = \{1-3----10---4-2\}$.

Let σ_1 and σ_2 be two feasible sequences generated by placing jobs in the unscheduled positions of π_1 and π_2 as follows respectively. As for σ_1, a feasible complete sequence is generated by placing the unscheduled jobs with respect to π_1, taken one at a time, alternately in the unscheduled positions to the right extreme side and left extreme side of J_{10}. This is done so in order to have a V-shaped property of the final full sequences. As for σ_2, a feasible complete sequence is generated by placing the unscheduled jobs with respect to π_2 (taken one at a time), alternately in the unscheduled positions to the left extreme side and right extreme side of the partial sequence. For example, with job J_{10} placed in position six in both π_1 and $\pi_{2,}$ and with the consideration of job J_4, we have σ_1 and σ_2 obtained from π_1 and π_2 respectively given as follows:

$\sigma_1=\{1-3-4-6-8-10-9-7-5-2\}$ and $\sigma_2=\{1-3-5-7-9-10-8-6-4-2\}$.

We have CTV(σ_1) = 487.24 and CTV(σ_2) = 488.36 as the completion time variances of the two complete sequences σ_1 and σ_2 (see Step 3.5 in Section 4.1).

Since CTV(σ_1) < CTV(σ_2)$\pi=\pi_1$, and hence$\pi=\{1-3-4---10----2\}$; set $\beta' \leftarrow \sigma_1$ and hence CTV(β') = CTV(σ_1) = 487.24 (see Step 3.5 in SMH1 heuristic).

Since CTV(β') < CTV(β),

$\beta=\beta'$ and

$\beta=\{1-3-\mathbf{4}-6-8-10-9-7-5-2\}$ and hence CTV (β) = 487.24 (see Step 3.6 in SMH1 heuristic).

Set $j=j+1$; if $j<(n-1)$, repeat the above procedure for every j.

For Job J_5: ($j=5$):

$\sigma_1=\{1-3-4-5-7-10-9-8-6-2\}$ and $\sigma_2=\{1-3-4-6-8-10-9-7-5-2\}$.

CTV (σ_1) = 498.6, CTV σ_2 = 487.24, and$\pi=\{1-3-4---10---\mathbf{5}-2\}$.

For Job $J_{6,}$ we have the following:$\sigma_1=\{1-3-4-6-8-10-9-7-5-2\}$ and $\sigma_2=\{1-3-4-7-9-10-8-6-5-2\}$.CTV ($\sigma_1$) = 487.24, CTV ($\sigma_2$) = 486.44,$\pi=\{1-3-4---10--\mathbf{6}-5-2\}$,

Since CTV(σ_2) < CTV(σ_1)

$\beta' \leftarrow \sigma_2$ and hence CTV(β') = CTV(σ_2) = 486.44.

Since CTV(β') < CTV(β),$\beta=\beta'$ and$\beta=\{1-3-4-7-9-10-8-\mathbf{6}-5-2\}$ and hence CTV (β) = 486.44.

For Job J_7, we have:$\sigma_1=\{1-3-4-7-9-10-8-6-5-2\}$ and $\sigma_2=\{1-3-4-8-9-10-7-6-5-2\}$.CTV($\sigma_1$ = 486.44), CTV((σ_2) = 491.81), and$\pi=\{1-3-4-\mathbf{7}--10--6-5-2\}$.

For Job J_8 (i.e., we proceed up to $j<n-1$):$\sigma_1=\{1-3-4-7-8-10-9-6-5-2\}$ and $\sigma_2=\{1-3-4-7-9-10-8-6-5-2\}$.CTV ($\sigma_1$) = 486.56, CTV ($\sigma_2$) = 486.44, and$\pi=\{1-3-4-7--10-\mathbf{8}-6-5-2\}$.Therefore, $\beta=\{1-3-4-7-9-10-8-6-5-2\}$ and CTV (β) = 486.44.

Next, this solution is checked for further improvement by using Lemma 1 or Lemma 2, i.e., by swapping the complimentary pairs. For example, since job J_9 and J_{10} are complimentary pairs, swapping the results in the following sequence:$\beta''=\{1-3-4-7-10-9-8-6-5-2\}$ and CTV (β'') = 486.86 (see Step 4 in SMH1 heuristic).

Since CTV(β'') > CTV(β), the sequence β is not updated.

By swapping the complimentary pairs J_6 and J_7, the following sequence for β'' is obtained.

$\beta''=\{1-3-4-6-9-10-8-7-5-2\}$ and CTV β'' = 486.4.

Since CTV(β'') < CTV(β), we have

CTV(β) = CTV(β'') = 486.4.

In the same manner, other feasible complimentary pairs are checked for possible improvement. The above steps are repeated for every sequence generated with respect to seven possible positions of J_{10} and the final β and CTV(β), yielding the solution of the proposed heuristic, SMH1 is obtained.

In this problem, the best sequence is obtained when job J_{10} is placed in position 6 and with application of the swapping technique presented in Lemma 2, the best solution with CTV, 486.4 with $\beta=\{1-3-4-6-9-10-8-7-5-2\}$ is obtained.

9 Appendix C: an illustrative example for SCH + SMH1 and FCH + SMH1

Consider a 14-job, 3-machine problem. For the purpose of easy understanding and presentation, the nomenclature of jobs J_1, J_2, …, and J_{14}, such that $P_1 \geq P_2 \geq P_3 \geq ... \geq P_{14}$ is used. Table 5 presents the allocation pattern of the jobs on the three machines using the SCH + SMH1 heuristic and Table 6 presents the allocation of the jobs on three machines using the FCH + SMH1 heuristic. The allocation pattern is the same for both versions of the parallel-machine problem. The SMH1 is used to solve the sequencing phase of the Pm||CTV problem.

Table 5 Allocation of jobs to machines using SCH + SMH1

Machine	Jobs				
M3	J_3	J_4	J_7	J_{12}	
M2	J_2	J_5	J_8	J_{11}	J_{14}
M1	J_1	J_6	J_9	J_{10}	J_{13}

Table 6 Allocation of jobs to machines using FCH + SMH1

Machine	Jobs				
M3	J_3	J_6	J_7	J_{10}	
M2	J_2	J_5	J_8	J_{11}	J_{14}
M1	J_1	J_4	J_9	J_{12}	J_{13}

Acknowledgments We thank DAAD for the support to carry out a part of the research work at the University of Passau, Germany. We are thankful to the organization where the case study had been studied. We are thankful to both of the reviewers for the constructive suggestion and comments that helped a lot to improve the paper.

References

1. Merten AG, Muller ME (1972) Variance minimization in single machine sequencing problems. Manag Sci 18(9):518–528
2. Eilon S, Chowdhury IG (1977) Minimizing waiting time variance in the single machine problem. Manag Sci 23(6):567–575
3. Schrage L (1975) Minimizing the time-in-system variance for a finite jobset. Manag Sci 21(5):540–543
4. Kanet JJ (1981) Minimizing variation of flow time in single machine systems. Manag Sci 27(12):1453–1459
5. Vani V, Raghavachari M (1987) Deterministic and random single machine sequencing with variance minimization. Oper Res 35(1): 111–120
6. Hall NG, Kubiak W (1991) Proof of a conjecture of Schrage about the completion time variance problem. Oper Res Lett 10(8):467–472
7. Kubiak W (1993) Completion time variance minimization on a single machine is difficult. Oper Res Lett 14:49–59
8. Manna DK, Prasad VR (1999) Bounds for the position of the smallest job in completion time variance minimization. Eur J Oper Res 114(2):411–419
9. Prasad VR, Manna DK (1997) Minimization of expected variance of completion times on single machine for stochastic jobs. Nav Res Logist 44:97–108
10. Li X, Chen Y, Sun Y, Sawhney R (2007) Minimizing class-based completion time variance on a single machine. Int J Oper Res 4:1–7
11. Ye N, Li X, Farley T, Xu X (2007) Job scheduling methods for reducing waiting time variance. Comput Oper Res 34:3069–3083
12. Li X, Chen Y, Sun Y (2010) Minimizing job completion time variance for service stability on identical parallel machines. Comput Ind Eng 58:729–738
13. Srirangacharyulu B, Srinivasan G (2010) Completion time variance minimization in single machine and multi-machine systems. Comput Oper Res 37:62–71
14. Mittenthal J, Raghavachari M, Rana AI (1993) A hybrid simulated annealing approach for single machine scheduling problems with non-regular penalty functions. Comput Oper Res 20(2):103–111
15. Al-Turki U, Fedjki C, Andijani A (2001) Tabu search for a class of single-machine scheduling problems. Comput Oper Res 28:1223–1230
16. De P, Ghosh JB, Wells CE (1992) On the minimization of completion time variance with a bicriteria extension. Oper Res 40(6):1148–1155
17. Vishwanathkumar G, Srinivasan G (2003) A branch and bound algorithm to minimize completion time variance on a single processor. Comput Oper Res 30:1135–1150
18. Srinivasan G, Srirangacharyulu B (2012) Minimizing variance of job completion times in a single machine. Int J Oper Res 13:110–127
19. Federgruen A, Mosheiov G (1996) Heuristics for multi machine scheduling problems with earliness and tardiness costs. Manag Sci 42(11):1544–1555
20. Chen Y, Li X, Sawhney R (2009) Restricted job completion time variance minimization on identical parallel machines. Eur J Ind Eng 3:261–276
21. Gokhale R, Mathirajan M (2012) Scheduling identical parallel machines with machine eligibility restrictions to minimize total weighted flowtime in automobile gear manufacturing. Int J Adv Manuf Technol 60(9-12):1099–1110
22. Chen JF (2009) Scheduling on unrelated parallel machines with sequence-and machine-dependent setup times and due-date constraints. Int J Adv Manuf Technol 44(11):1204–1212
23. Lee ZJ, Lin SW, Ying KC (2010) Scheduling jobs on dynamic parallel machines with sequence-dependent setup times. Int J Adv Manuf Technol 47(5):773–781
24. Sivasankaran P, Shahabudeen P (2014) Literature review of assembly line balancing problems. Int J Adv Manuf Technol 73(9-12): 1665–1694
25. Chaudhry IA, Drake PR (2009) Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms. Int J Adv Manuf Technol 42(5-6):581–594