

Minimization of completion time variance in flowshops using genetic algorithms

Imran Ali Chaudhry¹, Isam A-Q Elbadawi¹, Amer Farhan Rafique², Attia Boudjemline¹, Mohamed Boujelbene¹, Muhammed Usman³, Mohamed Aichouni¹

1 Department of Industrial Engineering, College of Engineering, University of Ha'il, Saudi Arabia

2 Department of Aerospace Engineering, College of Engineering, King Abdulaziz University, Saudi Arabia

3 College of Engineering, University College Dublin, Dublin, Ireland

Abstract

The majority of the flowshop scheduling literature focuses on regular performance measures like makespan, flowtime etc. In this paper a flowshop scheduling problem is addressed where the objective is to minimize completion time variance (CTV). CTV is a non-regular performance measure that is closely related to just-in-time philosophy. A Microsoft Excel spreadsheet-based genetic algorithm (GA) is proposed to solve the problem. The proposed GA methodology is domain-independent and general purpose. The flowshop model is developed in the spreadsheet environment using the built-in formulae and function. Addition of jobs and machines can be catered for without the change in the basic GA routine and minimal change to the spreadsheet model. The proposed methodology offers an easy to handle framework whereby the practitioners can implement a heuristic-based optimization tool with the need for advanced programming tools. The performance of the proposed methodology is compared to previous studies for benchmark problems taken from the literature. Simulation experiments demonstrate that the proposed methodology solves the benchmark problems efficiently and effectively with a reasonable accuracy. The solutions are comparable to previous studies both in terms of computational time and solution quality.

 **OPEN ACCESS**

Published: 02/06/2022

Accepted: 22/05/2022

DOI:
10.23967/j.rimni.2022.05.002

Keywords:
Completion time variance (CTV)
Flow shop
genetic algorithms
scheduling
spreadsheet

1. Introduction

Manufacturing plays a vital role in developing the economy as well as the human resources. In today's competitive world efficient use of materials and resources has become imperative to generate improved revenues. Scheduling research primarily focuses on regular performance measures where the objective function increases as a function of job completion times. Commonly used regular performance measures are: makespan, mean flowtime, tardiness, etc. For regular performance measures, the determination of an optimal solution is relatively simple. Most manufacturing companies nowadays prefer just-in-time (JIT) service or production systems. Earliness and tardiness are both undesirable in a JIT environment. CTV is one such non-regular performance measure that penalizes earliness as well as tardiness. It is also closely related to quality of service provided to each customer. CTV minimization is especially important when all jobs must spend the same time in the system, i.e., each job requires the same treatment. Smaller CTV ensures more stable service. CTV applications are found in many real-world applications such as production planning and internet data packet dispatching.

The first reported instance of the application of CTV minimization was proposed by Merten and Muller [1] in computer systems for solving the file organization problem where uniform response time was desired to be provided to users' requests to retrieve data files. Research has shown that an optimal CTV solution has a V-shape property [2]. In a V-shaped optimal solution, the jobs before the smallest job are scheduled in decreasing order of the processing times, while the jobs after the smallest job are scheduled in an ascending order of the processing times.

Since the first application of CTV by Merten and Muller [1], most of the scheduling literature addressed CTV minimization in a single machine environment, e.g., Nasini and Nessah [3], Wang and Lu [4], Viswanathkumar and Srinivasan [5], Sharma [6], Gupta et al. [7], etc. Minimization of CTV is considered *NP*-hard even for a single machine case [8]. However, very little work has been reported for CTV minimization in a flowshop environment.

Flowshop scheduling research has attracted wide attention among researchers due to its many industrial as well as economic applications. Some of the recent reviews highlighting the importance of flowshop scheduling has been presented by Çolak and Keskin [9], Lee and Loong [10] and González-Neira et al. [11]. In a regular flowshop, a set of n

-jobs are required to be processed on m -groups of machines. Each of the n jobs follows the same process sequence. Minimization of CTV in a flowshop ensures uniform turnover of jobs and delivery to customer.

2. Literature review

The first instance of CTV minimization was reported by Marangos et al. [12]. They applied a simulated annealing algorithm (SA) to minimize CTV in a two-machine flowshop. The authors reported that the proposed approach obtains a solution in the range of 5-10% of the lower bounds. Additionally, the SA algorithm can guarantee an optimal solution for problems having up to 10 jobs. Gowrishankar et al. [13] addressed CTV minimization in an n -jobs m -machine flowshop. The authors state that this is the first attempt to minimize CTV in an n -jobs m -machine flowshop. A breadth first branch & bound procedure as developed to solve the problem. Experimental analysis showed that the proposed method can effectively solve flowshop problems with CTV objective function.

Chandrasekaran et al. [14] applied a particle swarm optimization (PSO) algorithm to minimize CTV. Benchmark flowshop problems proposed by Taillard [15] are used to check the performance and efficiency of the proposed PSO. The authors reported that the results obtained are better when compared with the previously published results. Gajpal and Rajendran [16] applied an ant colony optimization (ACO) algorithm to minimize CTV. The authors developed a new ACO algorithm (NACO) for the problem. The performance of the proposed NACO is compared with existing ACO algorithms. Experimental analysis showed that, on average, the proposed ACO finds better results as compared to existing ACO algorithms.

Chandrasekaran et al. [17] proposed a PSO algorithm for multi-objective flowshop scheduling problems where the objective is simultaneous minimization of flowtime, makespan and CTV. The performance of the proposed PSO is enhanced by a local search mechanism. Experimental analysis showed that the proposed PSO yields more non-dominated solutions as compared to other approaches. Chen et al. [18] considered CTV minimization in an unrestricted parallel machine scheduling problem. A heuristic algorithm is proposed to solve the problem. The proposed heuristic algorithm can generate almost optimal schedules for small-sized problems. Ponnambalam et al. [19] proposed a discrete PSO to solve a multi-objective flowshop scheduling problem where the objective is simultaneous minimization of flowtime, makespan and CTV. Computational analysis showed that the proposed discrete PSO performs better in terms of quality of solution as compared to previous studies. Srirangacharyulu and Srinivasan [20] proposed a GA and a heuristic algorithm to minimize CTV in multi and single machine systems. Comparative analysis demonstrated that the proposed algorithms can provide better results as compared to the existing algorithms.

Li et al. [21] considered minimization of CTV in an unrestricted parallel machine environment and proposed an efficient heuristic Wavy Assignment, Verified Schedule (WAVS). The proposed algorithm produces near optimal solutions for small sized problems while outperforming existing algorithms for large sized instances. Rameshkumar et al. [22] proposed two PSO algorithms: an ACO-inspired PSO and a discrete PSO. Extensive computational analysis was carried out to compare the performance of the two proposed algorithms with previous studies for Taillard [15] benchmark problems. On average, the proposed algorithm provides good and promising solutions. Krishnaraj et al. [23] proposed an adaptation of ACO to minimize CTV in n -jobs m -machines flowshop scheduling problem. Although the proposed algorithm does not provide best schedules for Taillard [15] benchmark problems, it does provide good and promising solutions for most of the instances. Mehta et al. [24] proposed a two-phase local search algorithm to minimize CTV in a permutation flowshop. In the first phase, an initial V -shaped solution is constructed using a local search method and then improved it to obtain a locally optimal V -shaped solution. In the second phase, any possible sequences are traversed that may improve the solution.

Eren [25] considered minimization of CTV for a two-machine flowshop. A non-linear programming (NLP) method is proposed to solve the problem. The proposed algorithm can effectively solve problems up to 30 jobs. Problems up to 500 jobs are solved using a heuristic method. The author demonstrated that the proposed algorithm is very practical in solving real-world problems more effectively than the NLP models. Lee et al. [26] considered minimization of total weighted completion time (TWCT) for identical parallel machine scheduling. A two-agent based solution method is used to solve the problem. For fewer number of jobs, a branch & bound procedure is also developed. Rajkanth et al. [27] considered minimization of CTV for single and identical parallel machines. A heuristic method is proposed to solve the problem. The proposed heuristic method produces results better than the existing heuristics. It can also be modified and adapted to handle other non-regular performance measures like flow time variance and mean squared deviation as well as solve non-identical parallel machine problems.

Krishnaraj et al. [28] proposed a simulated annealing (SA) algorithm to minimize CTV in n -job m -machine flowshop scheduling problem. The proposed algorithm works in three phases. In the first phase, the algorithm minimizes CTV of jobs without right shifting of completion time (RSCT) of jobs. In phase two, the algorithm minimizes CTV of jobs with RSCT of jobs, except the last job. In phase three, sequences are converted to follow a V -shaped property with respect to processing time of jobs on the last machine, followed by RSCT except that of the last job so that the makespan and

the machine utilization in the shop floor remain the same. Extensive computational analysis was carried out with the existing heuristic algorithms. The proposed algorithm is able to find good results. Krishnaraj and Thiagarajan [29] also proposed a two-phase SA algorithm to minimize CTV in a flowshop environment. The performance of the proposed SA algorithm is tested on 90 benchmark flowshop problems from Taillard [15]. The proposed algorithm performs well in minimizing CTV especially in medium- and large-sized problems.

In this paper, we address the minimization of CTV in a flowshop environment using a general-purpose domain-independent GA embedded in a Microsoft Excel spreadsheet environment. The performance of the proposed methodology is compared with existing heuristics for a wide range of benchmark problems taken from the literature. Comparison is made with single machine, parallel machine and flowshop environments.

3. Problem definition & formulation of permutation flowshop scheduling problem

In a permutation flowshop scheduling problem, where the objective is to minimize completion time variance, n jobs from a set of $j = \{1, 2, 3, \dots, n\}$ are to be processed on m machines $k = \{1, 2, 3, \dots, m\}$. Each job j has a set of operations $\{o_{j1}, o_{j2}, o_{j3}, \dots, o_{jk}\}$ that are required to be processed through m machines requiring p_{jk} processing time.

Other assumptions used in the modeling of flowshop in this research application are as follows:

1. The number of n jobs and m machines are known before the planning horizon, i.e., $t = 0$.
2. All n jobs follow the same routing through the machines.
3. There is an unlimited buffer between the machines.
4. A different machine is required for the processing of each of the o_{jk} operations and there is only one machine of its kind in the flowshop.
5. A particular machine can process only one job at any given time and similarly, every job can only be processed by only one machine.
6. All processing times p_{jk} are known and deterministic.
7. Setup times for any operation, if any, are included in the processing time p_{jk} .
8. All machines are continuously available and there is no breakdown of machines.
9. Once started, an operation cannot be stopped, i.e., no pre-emption is allowed.

The problem then is to efficiently schedule the job to minimize the completion time variance (CTV). CTV for n jobs in a schedule is given by:

$$CTV = \frac{1}{n} \sum_{i=1}^n (C_i - \bar{C})^2$$

where C_i = completion time of job i

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n C_i$$

n = total number of jobs.

4. Proposed methodology

In the proposed methodology, a proprietary genetic algorithm Evolver 4.8 [30] is used for the optimization. Evolver works as an add-in to the Microsoft Excel spreadsheet. The flowshop model for minimization of CTV is developed within the spreadsheet using the built-in functions and formulae.

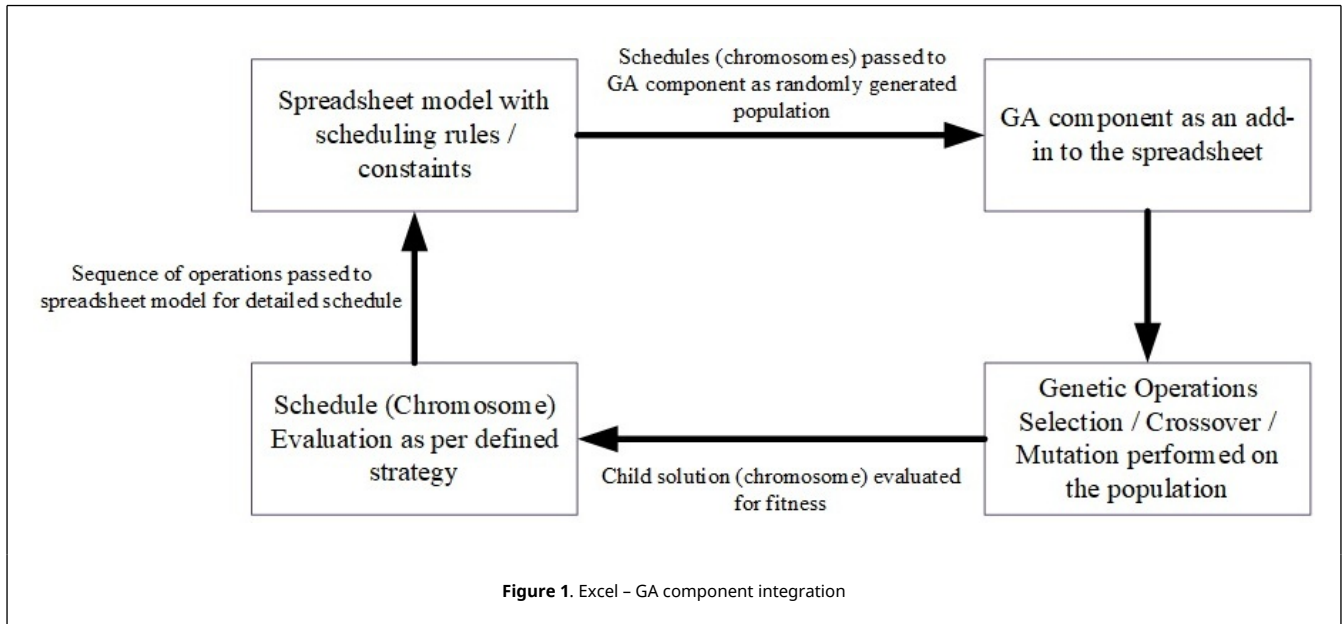
Research shows that, in the past few decades, spreadsheets have been extensively used in various fields of engineering. Spreadsheets are popular among the practitioners because they eliminate tedious and repetitive calculations. Some of the recent spreadsheet applications in engineering are: project scheduling [31], process planning and scheduling [32], airport ground crew scheduling [33], supply chain management (Amaral and Kuettnner [34] and Othman et al. [35]).

Evolver has also been used by many researchers for optimization problems of various applications. Some of the recent applications of using Evolver are: risk analysis of reservoir operation [36], choosing materials for eco-friendly buildings

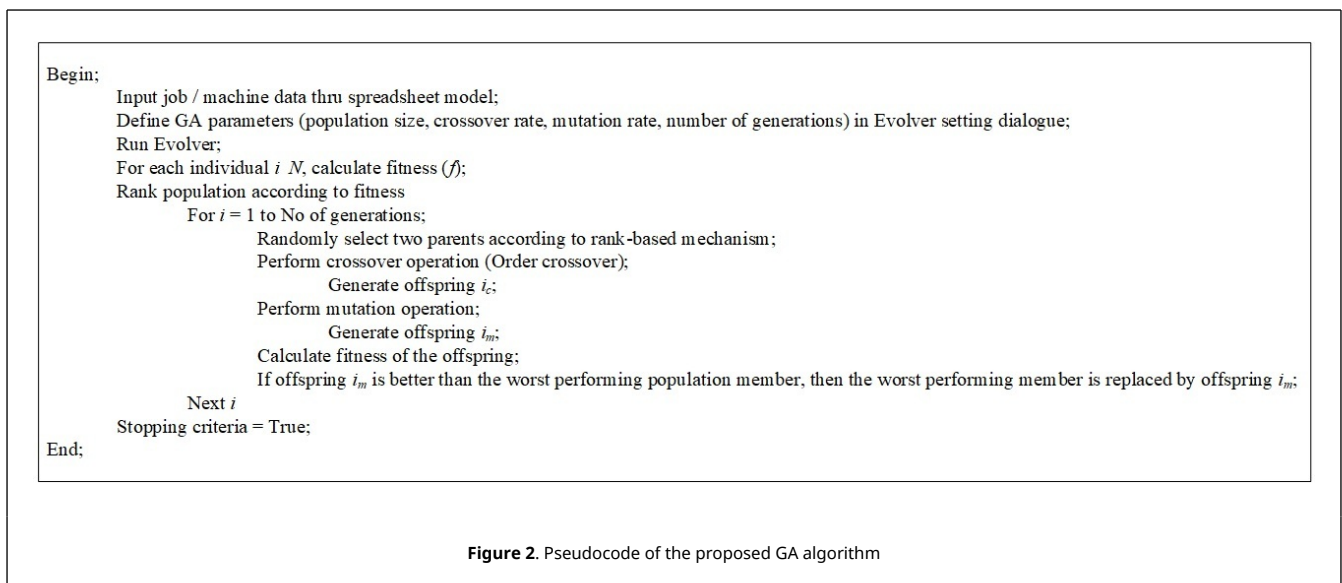
[37], energy efficient operation of submersible pumps [38], operational optimization of a heat pump system [39], optimization of repetitive scheduling [40], optimum crew selection for repetitive projects [41].

The use of Microsoft Excel and Evolver optimization software offers an easy-to-handle framework for the practitioner. This framework allows practitioners to implement a heuristic-based optimization tool without the need for advanced programming skills. Furthermore, Microsoft Excel and Evolver offer a user-friendly interface and facilitates user specification of the optimization parameters without the need for any programming skills.

The schematic diagram of spreadsheet – GA component integration is shown in [Figure 1](#).

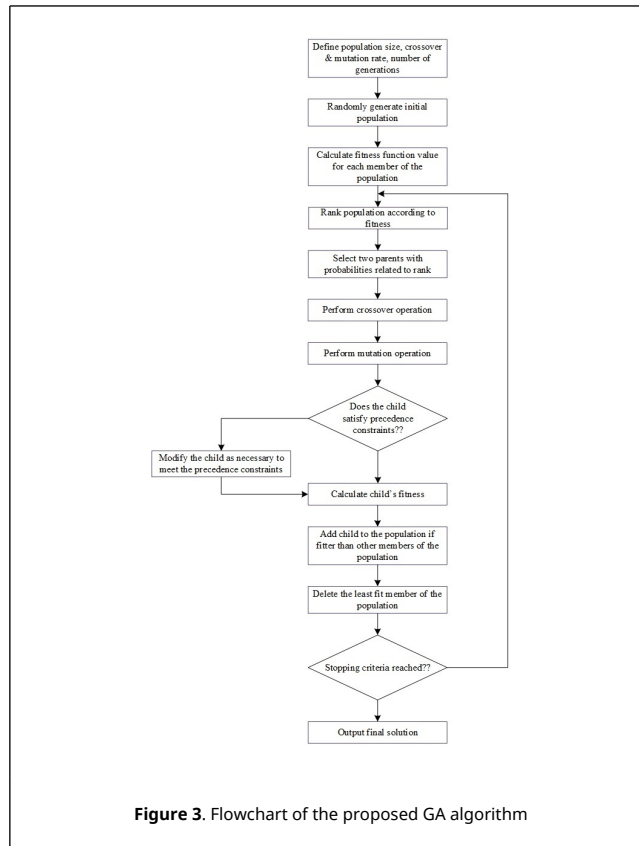


[Figures 2](#) and [3](#) show the pseudocode and flowchart respectively of the proposed algorithm.



4.1 Chromosome representation

Permutation representation for chromosome (solution) is required to be used for the minimization of CTV in a flowshop environment. To address a particular problem situation, Evolver has various solving methods. The Order



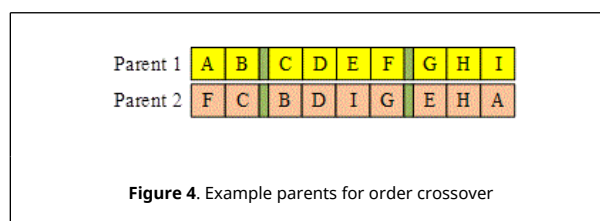
Solving method of Evolver is used to handle permutation representation. Permutation representation is described by considering six jobs D-C-A-B-E-F. In a permutation representation, the order of the genes are changed to formulate new solution strings. Chromosome F-C-B-A-E-D would be one of the possible chromosome representing permutation representation while another could be B-F-A-C-E-D.

4.2 Reproduction and selection

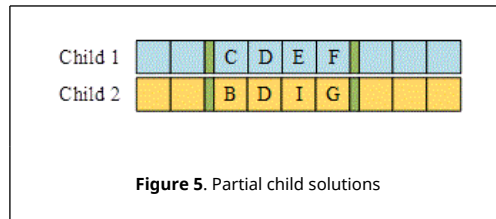
Steady state reproduction method [42] is used in Evolver. Unlike other reproduction techniques used in GA, steady state reproduction method produces only one child in each generation. The fitness of the child solution is compared to the fitness of other chromosomes in the population. In case the child solution is fitter, it replaces the worst performing member of the population, else it is discarded. For selection, Evolver uses rank-based selection method.

4.3 Crossover operator

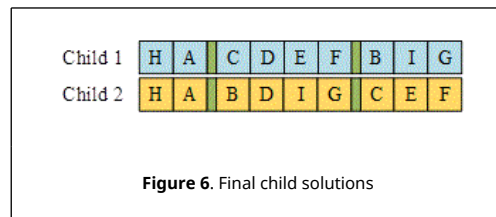
Crossover operator combines genes from two parents to formulate new solutions. Order crossover [43] is used in the Order solving method. Order crossover randomly selects genes from one parent, then finds their place in the second parent. The remaining genes are copied in the second parent in the same order as they appear in the first parent. This ensures that the sub-ordering in the original parents are preserved while creating some new sub-orderings. Consider the two representing solutions in [Figure 4](#).



The two cut points for each of the parents are indicated by green colour. The genes between the two cut points are copied in the child solution as they are in both parents, as shown in [Figure 5](#).

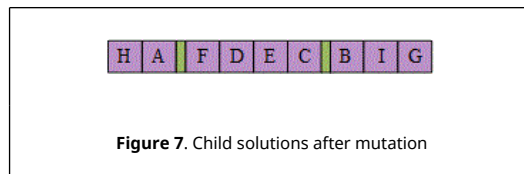


Next, for child 1, starting from the second cut point of parent 1, the jobs from the second parent are copied, except those which are already in child 1, we get H-A-B- I-G. This sequence is placed in the first child. Similarly, the procedure is applied for child 2, where the genes are now copied from parent 1. The resulting child solutions would be as in [Figure 6](#).



4.4 Mutation operator

With successive iterations, the population loses diversity and gets stuck in local optima. Mutation operator ensures that diversity is maintained in the population. The order solving method performs order-based mutation [43]. In an order-based mutation, two positions are selected randomly, and then the genes in these position are swapped. For example, at position 3 and 6 in child 1 in [Figure 6](#), we have genes C and F. By swapping these two genes, the resulting chromosome would be as shown in [Figure 7](#).



5. Computation analysis

Simulation experiments were carried out to check the performance of the proposed methodology. Benchmark problems were taken from already published literature to validate the usefulness and performance of the proposed methodology. Evolver version 4.8 was used for the GA while Excel version 2003 was used to develop the flowshop models. The following GA parameters were used for the simulation experiments: crossover rate = 0.65, mutation rate = 0.07 and population size = 70. Each of the benchmark problems was run for 30 replications. The simulation experiments were conducted on a 2nd generation i7 laptop computer with 8GB RAM.

Comparison is made with various heuristics developed in previous studies based on relative percentage deviation (RPD) from the best found solution among all heuristics. The RPD is calculated as follows:

$$RPD = 100 \times \frac{(CTV_{proposed} - CTV_{best})}{CTV_{best}}$$

where

$CTV_{proposed}$ = completion time variance result found by the proposed methodology, and

CTV_{best} = best completion time variance found among all previous heuristics.

A positive RPD means that the CTV result found by the given heuristic is worse than the best CTV result among all heuristics, while a negative RPD means that the result found by a given heuristics is worse than the best CTV result.

5.1. Problem set 1

The first problem set is taken from Gupta et al. [7]. The data set is a single machine scenario with jobs ranging from 10 to 20. The data set consists of 15 instances. Data for problem set 1 is presented in [Table A1](#).

The performance of the proposed methodology is compared with results from GGA (GA by Gupta et al. [7]), HAS (hybrid simulated annealing algorithm by Mittenthal et al. [44]) and TSA (tabu search algorithm by Al-Turki et al. [45]). The comparative analysis with three approaches is presented in [Table 1](#).

Table 1. Comparative analysis of proposed methodology with previous studies for data set 1

Prob Instance	No of Jobs	Optimum Solution	GGA	HAS	TSA	Proposed GA
P1	10	7027.96	0	0	0	0
P2	10	12269.76	0	0	0	0
P3	10	20903.36	0	0	0	0
P4	10	14094.01	0	0	0	0
P5	10	18884.8	0	0	0	0
P6	15	16052.2	0.0044	0.0066	0	0
P7	15	27082.8	0.0029	0	0	0
P8	15	29805.4	0.0060	0	0	0
P9	15	57713.76	0.0025	0.0024	0	0
P10	15	36005.18	0.0006	0.0002	0	0
P11	20	70843.39	0.1806	0.0192	0	0
P12	20	76050.66	0.0333	0.0001	0.0006	0
P13	20	58912.59	0.0669	0	0	0
P14	20	54589.11	0.1972	0.0028	0.0001	0
P15	20	50015.15	0.1093	0.0016	0.0003	0
Average RPD			0.0402	0.0022	0.0001	0.0000

From [Table 1](#), we can see that the average RPD for CGA, HAS, TSA and the proposed GA are 0.0402%, 0.0022%, 0.0001% and 0%, respectively. It is clear that, compared to CGA, HAS and TSA, the proposed GA methodology was able to find an optimal solution for all problem instances. The computation time required to find the optimum solution was less than 2 secs.

5.2. Problem set 2

Problem set 2 is taken from Srirangacharyulu and Srinivasan [20]. Problem set 2 is also a single machine scenario with jobs ranging from 15 to 25. The data set consists of nine instances. Data for problem set 2 is presented in [Table A2](#).

The performance of the proposed methodology is compared with results from EC1.1 and EC1.2 (heuristic methods by Eilon and Chowdhury [2]), JJK (heuristic method by Kanet [46]), GGB (GA by Gupta et al. [7]), MP (heuristic method by Prasad and Manna [47]), VS (verified spiral heuristic method by Ye et al. [48]), HA (hybrid GA algorithm by Srirangacharyulu and Srinivasan [20]), SMH1 (heuristic algorithm by Rajkanth et al. [27]). The comparative analysis with three approaches is presented in [Table 2](#).

Table 2. Comparative analysis of proposed methodology with previous studies for data set 2

Prob	No of Jobs	Optimum Result	EC1.1	EC1.2	JJK	MP	GGB	HA	VS	SMH1	Proposed GA
P1	15	38922.65	0.1669	0.1613	0.0003	0.0253	0.0003	0	0.0003	0.0012	0
P2	15	20102.38	0.2553	0.0882	0.0006	0.0049	0.0006	0	0.0006	0	0
P3	15	29217.09	0.0732	0.0061	0.0016	0.0016	0.0016	0	0.0016	0	0
P4	15	32551.32	0.1642	0.1351	0.0012	0.0072	0	0	0.0012	0	0
P5	20	64341.63	0.0898	0.0898	0.0030	0.0030	0.0009	0	0.0030	0	0
P6	20	51736.99	0.1830	0.1774	0.0049	0.0049	0.0474	0.0031	0.0049	0.0011	0
P7	25	107559.44	0.0550	0.0538	0.0003	0.0016	0.0002	0	0.0003	0.0003	0
P8	25	67358.88	0.0433	0.0434	0.0005	0.0006	0.0005	0	0.0005	0	0
P9	25	91018.42	0.0924	0.0686	0.0009	0.0013	0.0071	0.0006	0.0009	0.0015	0
Avg RPD			0.1248	0.0915	0.0015	0.0056	0.0065	0.0004	0.0015	0.0005	0

From [Table 2](#), we can see that the average RPD for EC1.1, EC1.2, JJK, MP, GGB, HA, VS, SMH1 and proposed GA are 0.1248%, 0.0915%, 0.0015%, 0.0056%, 0.0065%, 0.0004%, 0.0015%, 0.0005% and 0%, respectively. It is evident from the results in [Table 2](#) that the proposed GA was able to find an optimal solution for all problem instances of data set 2.

5.3. Problem set 3

Problem set 3 is taken from Li et al. [21]. The problem set is an unrestricted parallel machine scenario and consists of nine jobs to be scheduled on two parallel machines to minimize CTV. The problem set consists of four subsets with processing times following four kinds of distributions: the uniform distribution of *Uniform* (1, 59), the normal distribution of *Normal* (30; 10²), the exponential distribution of *Exponential* (30), and the Pareto distribution of *Pareto*

(1.0345,1). Each of the subsets has five problem instances. The comparative analysis with WAVS algorithm [21] is given in Table 3 to Table 7.

The comparative results in Table 3 to Table 6 shows the superior performance of the proposed GA methodology with respect to WAVS heuristics. The proposed GA methodology was able to find an optimal solution in all the instances. In all the twenty instances, the best solution was obtained within 4 secs on a 2nd generation i7 laptop computer with 8GB RAM.

Table 3. Comparative analysis of proposed methodology for data set 3 – Processing times from *Uniform* (1, 59)

Prob Instance	Optimal CTV	WAVS		Proposed GA	
		CTV	RPD	CTV	RPD
1	1512.85	1515.84	0.198	1512.85	0
2	831.28	840.4	1.097	831.28	0
3	322.74	324.98	0.694	322.74	0
4	737.5	737.5	0.000	737.50	0
5	630.59	632.53	0.308	630.59	0

Table 4. Comparative analysis of proposed methodology for data Set 3 – Processing times from *Normal* (30; 10²)

Prob Instance	Optimal CTV	WAVS		Proposed GA	
		CTV	RPD	CTV	RPD
1	948.09	948.98	0.094	948.09	0
2	1719.99	1732.44	0.724	1719.99	0
3	849.74	858.49	1.030	849.74	0
4	1073.74	1074.59	0.079	1073.74	0
5	1525.98	1528.85	0.188	1525.98	0

Table 5. Comparative Analysis of Proposed Methodology for Data Set 3 – processing times from *Exponential* (30)

Prob Instance	Optimal CTV	WAVS		Proposed GA	
		CTV	RPD	CTV	RPD
1	333.9	335.23	0.398	333.90	0
2	1161.78	1162.74	0.083	1161.78	0
3	1102.99	1102.99	0.000	1102.99	0
4	2250.94	2255.94	0.222	2250.94	0
5	362.19	362.34	0.041	362.19	0

Table 6. Comparative analysis of proposed methodology for data set 3 – Processing times from *Pareto* (1.0345,1)

Prob Instance	Optimal CTV	WAVS		Proposed GA	
		CTV	RPD	CTV	RPD
1	225.5	316.38	40.302	225.50	0
2	70.19	70.19	0.000	70.19	0
3	78.99	78.99	0.000	78.99	0
4	137.84	139.09	0.907	137.84	0
5	41.13	41.19	0.146	41.13	0

5.4. Problem set 4

Problem set 4 consists of flowshop problems proposed by Taillard [15]. The problem set has six different settings of n -jobs and m -machines. The data set consists of 20×5 , 20×10 , 20×20 , 50×5 , 50×10 , 50×20 settings. Each setting has 10 problem instances thus, in total 60 different problems were solved to compare the performance of the proposed GA methodology with previously reported results. The problem instances can be accessed at <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>.

The performance is compared with the following heuristics: SA – 2020 (simulated annealing heuristics by Krishnaraj and Thiagarajan [29]), SA – 2019 (simulated annealing algorithm by Krishnaraj et al. [28]), ACO (ant colony algorithm by Gajpal and Rajendran [16]), MACO (modified ant colony algorithm by Krishnaraj et al. [23]), DPSO 1 (discrete particle swarm algorithm by Rameshkumar et al. [22]) and DPSO 2 (discrete particle swarm algorithm by Ponnambalam et al. [19]). The comparative analyses for CTV results for all 60 instances are presented in Table 7 to Table 12.

It can be seen from Table 7 to Table 12, that the proposed methodology was able to find a solution very close to previously best known solutions. For 20×5 problem instances, the proposed method provided the same solutions for five instances, and the average RPD was 0.03%. Overall, the proposed GA results were worse than SA – 2020 and DPSO 1.

For 20×10 problem instances, the proposed method provided the same solutions for three instances with an average RPD of 0.23%. The proposed methodology solution was only better than ACO.

For 20×20 problem instances, the proposed method provided the same solutions for two instances with an average RPD of 0.61%. In this case also, the proposed methodology solution was only better than ACO.

For 50×5 problem instances, the proposed method did not find the same solutions for any of the instances. The average RPD was 0.61% for the proposed method. In this case too, the proposed methodology solution was only better than ACO.

Table 7. Comparative analysis of proposed methodology with previous studies for data set 4 – 20×5

Instance	SA - 2020	SA - 2019	ACO	MACO	DPSO 1	DPSO 2	Proposed GA
1	0.00	0.00	1.36	0.00	0.00	0.00	0.00
2	0.00	0.00	1.85	0.00	0.00	0.00	0.00
3	0.06	0.03	0.27	0.06	0.00	0.19	0.00
4	0.00	0.00	3.12	0.00	0.00	0.00	0.00
5	0.00	0.27	0.73	0.27	0.00	0.27	0.27
6	0.00	0.00	1.20	0.00	0.00	0.00	0.00
7	0.00	0.00	1.05	0.00	0.00	0.00	0.00
8	0.00	0.06	0.60	0.06	0.00	0.00	0.00
9	0.00	0.00	1.44	0.00	0.00	0.00	0.06
10	0.00	0.02	1.09	0.02	0.00	0.00	0.02
Avg RPD	0.01	0.04	1.27	0.04	0.00	0.05	0.03

Table 8. Comparative analysis of proposed methodology with previous studies for data set 4 – 20×10

Instance	SA - 2020	SA - 2019	ACO	MACO	DPSO 1	Proposed GA
1	0.00	0.00	2.38	0.00	0.00	0.45
2	0.00	0.67	0.85	0.85	0.00	0.67
3	0.00	0.00	1.28	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.22
5	0.00	0.07	1.49	0.07	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.06
7	0.00	0.00	7.97	0.00	0.00	0.00
8	0.87	0.87	2.10	0.87	0.00	0.87
9	0.00	0.31	0.61	0.31	0.00	0.00
10	0.00	0.00	4.73	0.00	0.00	0.00
Avg RPD	0.09	0.19	2.14	0.21	0.00	0.23

Table 9. Comparative analysis of proposed methodology with previous studies for data set 4 – 20×20

Instance	SA - 2020	SA - 2019	ACO	MACO	DPSO 1	Proposed GA
1	0.00	0.00	3.86	0.00	0.00	0.67
2	0.00	0.00	1.90	0.00	0.00	0.65
3	0.30	0.30	0.30	0.30	0.00	0.47
4	0.00	0.00	1.23	0.00	0.00	0.00
5	0.00	0.00	1.21	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	1.30	0.00	0.00	0.96
8	0.00	0.00	4.51	0.00	1.27	3.13
9	0.00	0.99	1.07	0.99	0.00	0.00
10	1.48	0.00	3.03	0.00	0.00	0.24
Avg RPD	0.18	0.13	1.84	0.13	0.13	0.61

Table 10. Comparative analysis of proposed methodology with previous studies for data set 4 – 50×5

Instance	SA - 2020	SA - 2019	ACO	MACO	DPSO 1	Proposed GA
1	0.12	0.00	0.99	0.43	0.25	0.91
2	0.00	0.02	1.54	0.02	0.37	0.39
3	0.28	0.46	2.51	0.46	0.00	1.24
4	0.00	0.29	1.46	0.66	0.29	0.84
5	0.00	0.10	1.21	0.10	0.05	0.22
6	0.04	0.00	0.76	0.22	0.03	0.33
7	0.00	0.40	0.88	0.54	0.42	0.19
8	0.00	0.10	1.13	0.10	0.10	0.44
9	0.09	0.00	1.08	0.42	0.06	0.74
10	0.02	0.00	1.06	0.07	0.00	0.25
Avg RPD	0.05	0.14	1.26	0.30	0.16	0.55

Table 11. Comparative analysis of proposed methodology with previous studies for data set 4 – 50×10

Instance	SA - 2020	SA - 2019	ACO	MACO	DPSO 1	Proposed GA
1	0.22	0.00	1.03	0.54	0.80	2.14
2	0.00	0.51	3.45	1.02	2.93	3.22

3	0.55	0.00	1.21	1.13	0.04	3.43
4	0.13	0.00	2.53	0.61	0.83	1.98
5	0.00	0.14	0.54	0.14	0.10	1.42
6	0.66	0.86	2.79	0.86	0.00	2.29
7	0.00	0.51	1.79	0.87	0.18	1.86
8	0.00	1.30	2.37	1.30	1.72	2.02
9	0.00	0.65	2.27	1.17	0.45	2.34
10	0.02	0.22	0.86	0.86	0.00	1.23
Avg RPD	0.16	0.42	1.88	0.85	0.70	2.19

Table 12. Comparative analysis of proposed methodology with previous studies for data set 4 – 50×20

Instance	SA - 2020	SA - 2019	ACO	MACO	DPSO 1	Proposed GA
1	0.66	0.00	1.82	0.46	0.35	3.25
2	1.24	0.00	2.33	0.00	1.17	3.82
3	0.03	0.47	0.47	0.47	0.00	3.25
4	0.00	0.63	1.57	1.12	1.15	4.57
5	0.00	0.76	5.03	3.07	0.90	2.77
6	0.00	0.45	0.80	0.65	0.80	3.43
7	0.55	0.00	2.51	1.26	0.37	2.99
8	0.91	0.00	5.60	2.43	1.75	6.12
9	0.00	0.77	2.79	1.81	2.79	4.65
10	0.29	1.30	1.61	1.61	0.00	6.82
Avg RPD	0.37	0.44	2.45	1.29	0.93	4.17

For 50 × 10 problem instances, the proposed method did not find the same solutions for any of the instances. The average RPD was 2.19% for the proposed method. For 50 × 10 problem instances, the proposed method did not find the same solutions for any of the instances. The average RPD was 4.17% for the proposed method.

From the discussion in the preceding paragraphs, we can see that the proposed GA methodology found optimal solutions for all single machine and unrestricted parallel machine scenarios. For $n \times m$ flowshop problems, the solution quality deteriorated as the problem size increased. However, for small-to-medium-sized problems, the solution quality of the proposed methodology was comparable with other heuristic techniques.

6. Conclusions

Most of the research on flowshop scheduling addresses the problem with regular performance measures like, makespan, flowtime etc. Very little work has been carried out with non-regular performance measures. Completion time variance (CTV) is one of the non-regular objective functions that penalizes both earliness and tardiness. CTV has direct relevance with JIT where it is desired that a job is completed exactly on time, neither early, as it will incur inventory holding cost, nor late, as it will result in products being delivered late thus, resulting in bad reputation for the company.

In this research, minimization of completion time variance (CTV) in a flowshop was addressed. A general purpose domain-independent GA Evolver that works as an add-in to Microsoft Excel spreadsheet was used for the optimization of the schedules. The shop models for flowshop were made in the spreadsheet using the built-in functions and formulae.

The proposed approach offers an easy-to-handle framework for the practitioners who are used to tools like spreadsheets. The proposed methodology allows practitioners to implement a heuristic-based optimization tool without the need for advanced programming skills. Furthermore, due to the arrangement of data in linked cells via the spreadsheet, it is very convenient to carry out what-if analysis thus, looking at the effect of change due to various parameters. Additionally, the proposed approach can be used to optimize any objective function without the need to change the spreadsheet model of the basic GA routine. The proposed methodology can also be enhanced with the use of Visual Basic for Application (VBA) language available within the Excel spreadsheet.

Even though the performance of the proposed methodology was not very promising for large-sized problems, computational experiments demonstrated that the proposed methodology can find optimal solutions for small-to-medium-sized problems.

Acknowledgements

This research has been funded by Scientific Research Deanship at University of Ha'il – Saudi Arabia through Project number RG – 20 028.

References

- [1] Merten A.G., Muller M.E. Variance minimization in single machine sequencing problem. *Manage. Sci.*, 18(9):518-528, 1972.
- [2] Eilon S., Chowdhury I.G. Minimising waiting time variance in the single machine problem. *Manage. Sci.*, 23(6):567-575, 1977.
- [3] Nasini S., Nessah R. An almost exact solution to the min completion time variance in a single machine. *Eur. J. Oper. Res.*, 294(2):427-441, 2021.
- [4] Wang S., Lu Y. A branch and price algorithm for single-machine completion time variance. *Comput. Oper. Res.*, 109:188-199, 2019.
- [5] Viswanathkumar G., Srinivasan G. A branch and bound algorithm to minimize completion time variance on a single processor. *Comput. Oper. Res.*, 30(8):1135-1150, 2003.
- [6] Sharma P. Permutation polyhedra and minimisation of the variance of completion times on a single machine. *J. Heuristics*, 8(4):467-485, 2002.
- [7] Gupta M.C., Gupta Y.P., Kumar A. Minimizing flow time variance in a single machine system using genetic algorithms. *Eur. J. Oper. Res.*, 70(3):289-303, 1993.
- [8] Kubiak W. Completion time variance minimization on a single machine is difficult. *Oper. Res. Lett.*, 14(1):49-59, 1993.
- [9] Çolak M., Keskin G.A. An extensive and systematic literature review for hybrid flowshop scheduling problems. *Int. J. Ind. Eng. Comput.*, 13(2):185-222, 2022.
- [10] Lee T.-S., Loong Y.-T. A review of scheduling problem and resolution methods in flexible flow shop. *Int. J. Ind. Eng. Comput.*, 10(1):67-88, 2019.
- [11] González-Neira E.M., Montoya-Torres J.R., Barrera D. Flow-shop scheduling problem under uncertainties: Review and trends. *Int. J. Ind. Eng. Comput.*, 8(4):399-426, 2017.
- [12] Marangos C.A., Govande V., Srinivasan G., Zimmers E.W. Algorithms to minimize completion time variance in a two machine flowshop. *Comput. Indus. Eng.*, 35(1):101-104, 1998.
- [13] Gowrishankar K., Rajendran C., Srinivasan G. Flow shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. *Eur. J. Oper. Res.*, 132(3):643-665, 2001.
- [14] Chandrasekaran S., Ponnambalam S.G., Suresh R.K., Vijayakumar N. An application of particle swarm optimization algorithm to permutation flowshop scheduling problems to minimize makespan, total flowtime and completion time variance. *IEEE International Conference on Automation Science and Engineering*, pp. 513-518, 2006.
- [15] Taillard E. Benchmarks for basic scheduling problems. *Eur. J. Oper. Res.*, 64(2):278-285, 1993.
- [16] Gajpal Y., Rajendran C. An ant-colony optimization algorithm for minimizing the completion-time variance of jobs in flowshops. *Int. J. Prod. Econ.*, 101(2):259-272, 2006.
- [17] Chandrasekaran S., Ponnambalam S.G., Suresh R.K., Vijayakumar N. Multi-objective particle swarm optimization algorithm for scheduling in flowshops to minimize makespan, total flowtime and completion time variance. *IEEE Congress on Evolutionary Computation*, pp. 4012-4018, 2007.
- [18] Chen Y., Li X., Kumanan S. Job completion time variance minimization on identical parallel machines. In *Proceedings of the 2007 Industrial Engineering Research Conference*, G. Bayraksan et al., Editors, Pittsburgh, PA, 2007.
- [19] Ponnambalam S.G., Jawahar N., Chandrasekaran S. Discrete particle swarm optimization algorithm for flowshop scheduling. In *Particle Swarm Optimization*, A. Lazinic (Ed.), IntechOpen, pp. 397-422, 2009.
- [20] Srirangacharyulu B., Srinivasan G. Completion time variance minimization in single machine and multi-machine systems. *Comput. Oper. Res.*, 37(1):62-71, 2010.
- [21] Li X., Chen Y., Sun Y. Minimizing job completion time variance for service stability on identical parallel machines. *Comput. Indus. Eng.*, 58(4):729-738, 2010.
- [22] Rameshkumar K., Rajendran C., Mohanasundaram K.M. Discrete particle swarm optimisation algorithms for minimising the completion-time variance of jobs in flowshops. *Int. J. Ind. Syst. Eng.*, 7(3):317-340, 2011.
- [23] Krishnaraj J., Pugazhendhi S., Rajendran C., Thiagarajan S. A modified ant-colony optimisation algorithm to minimise the completion time variance of jobs in flowshops. *Int. J. Prod. Res.*, 50(20):5698-5706, 2012.
- [24] Mehta P., Pandit P., Philip D., Sharma P. An efficient local search for minimizing completion time variance in permutation flow shops. *Comput. Oper. Res.*, 39(5):1000-1009, 2012.
- [25] Eren T. Minimizing completion time variance in a flowshop scheduling problem with a learning effect. *Gazi Univ. J. Sci.*, 26(3):389-397, 2013.
- [26] Lee W.-C., Wang J.-Y., Lin M.-C. A branch-and-bound algorithm for minimizing the total weighted completion time on parallel identical machines with two competing agents. *Knowl. Based. Syst.*, 105:68-82, 2016.
- [27] Rajkanth R., Rajendran C., Ziegler H. Heuristics to minimize the completion time variance of jobs on a single machine and on identical parallel machines. *Int. J. Adv. Manuf. Technol.*, 88(5):1923-1936, 2017.
- [28] Krishnaraj J., Pugazhendhi S., Rajendran C., Thiagarajan S. Simulated annealing algorithms to minimise the completion time variance of jobs in permutation flowshops. *Int. J. Ind. Syst. Eng.*, 31(4):425-451, 2019.
- [29] Krishnaraj J., Thiagarajan S. A two-phase simulated annealing algorithm to minimise the completion time variance of jobs in flowshops. *Int. J. Process. Manag. Benchmarking*, 10(2):261-281, 2020.
- [30] Palisade. Evolver: The genetic algorithm super solver Version 4. Palisade Corporation, New York, USA, 1998.
- [31] Agrama F.a.E.-M. Linear projects scheduling using spreadsheets features. *Alex. Eng. J.*, 50(2):179-185, 2011.
- [32] Chaudhry I.A., Usman M. Integrated process planning and scheduling using genetic algorithms. *Teh. Vjesn.*, 24(5):1401-1409, 2017.
- [33] Milutinović L.D., Makajić-Nikolić D., Antić S., Živić M., Lisec A. Control model for ground crew scheduling problem at small airports: case of Serbia. *Transport*, 36(3):235-245, 2021.
- [34] Amaral J., Kuettner D. Analyzing supply chains at HP using spreadsheet models. *INFORMS J. Appl. Anal.*, 38(4):228-240, 2008.
- [35] Othman S.N., Mustafa N.H., Sallehuddin R. Supply chain spreadsheet simulation optimization. *2012 Fourth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 188-193, 2012.
- [36] Li J., Huang J., Liang P., Lund J.R. Fuzzy representation of environmental flow in multi-objective risk analysis of reservoir operation. *Water Resour. Manag.*, 35(9):2845-2861, 2021.
- [37] Jeon B.-J., Kim B.-S. Development of material combination model considering economics and construction efficiency for G-SEED certification. *Sustainability-Basel*, 13(6):3535, 2021.

[38] Sperlich A., Pfeiffer D., Burgschweiger J., Campbell E., Beck M., Gnirss R., Ernst M. Energy efficient operation of variable speed submersible pumps: Simulation of a ground water well field. *Water*, 10(9):1255, 2018.

[39] Schellenberg C., Lohan J., Dimache L. Operational optimisation of a heat pump system with sensible thermal energy storage using genetic algorithm. *Therm. Sci.*, 22(5):2189-2202, 2018.

[40] Salama T., Moselhi O. Multi-objective optimization for repetitive scheduling under uncertainty. *Engineering, Construction and Architectural Management*, 26(7):1294-1320, 2019.

[41] Arabpour Roghabadi M., Moselhi O. Optimized crew selection for scheduling of repetitive projects. *Eng. Constr. Archit. Manag.*, 28(6):1517-1540, 2021.

[42] Whitley D., Kauth K. GENITOR: A different genetic algorithm. *Proceedings of the 1988 Rocky Mountain Conference on Artificial Intelligence*, pp. 118-130, 1988.

[43] Davis L. *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York, USA, 1991.

[44] Mittenthal J., Raghavachari M., Rana A.I. A hybrid simulated annealing approach for single machine scheduling problems with non-regular penalty functions. *Comput. Oper. Res.*, 20(2):103-111, 1993.

[45] Al-Turki U., Fedjki C., Andjijani A. Tabu search for a class of single-machine scheduling problems. *Comput. Oper. Res.*, 28(12):1223-1230, 2001.

[46] Kanet J.J. Minimizing variation of flow time in single machine systems. *Manage. Sci.*, 27(12):1453-1459, 1981.

[47] Prasad V.R., Manna D.K. Minimization of expected variance of completion times on single machine for stochastic jobs. *Nav. Res. Logist.*, 44(1):97-108, 1997.

[48] Ye N., Li X., Farley T., Xu X. Job scheduling methods for reducing waiting time variance. *Comput. Oper. Res.*, 34(10):3069-3083, 2007.

Appendix A

Table A1. Problem set 1 data

Problem Instance	Job Processing Times																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	70	33	17	6	12	22	40	52	61	88										
P2	98	37	22	12	14	26	50	78	79	99										
P3	82	68	37	11	34	48	72	73	79	94										
P4	76	46	39	10	34	41	48	52	70	96										
P5	77	52	44	10	42	48	62	65	75	98										
P6	90	39	29	25	20	11	62	14	24	26	32	47	80	83	98					
P7	91	63	45	35	24	16	62	20	27	41	59	64	72	77	97					
P8	91	58	55	43	26	13	62	18	40	50	57	62	63	77	97					
P9	97	82	73	64	42	15	62	30	61	68	79	83	92	95	98					
P10	85	67	52	42	32	16	62	26	40	51	59	77	82	84	96					
P11	97	75	71	59	57	46	32	17	8	9	26	44	49	58	61	73	77	86	88	99
P12	92	84	77	74	56	43	30	13	2	9	21	36	47	70	75	83	87	88	90	97
P13	94	74	69	53	46	38	28	20	7	9	27	29	40	48	63	71	77	79	89	97
P14	77	69	63	56	47	44	25	14	2	4	22	35	46	55	60	66	72	73	74	82
P15	92	70	63	47	40	36	27	14	6	10	24	31	37	41	48	65	78	83	89	94

Table A2. Problem set 2 data

Prob	# of Mch	Processing Times																								
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
P1	15	83	77	76	73	65	65	63	62	59	55	46	40	19	18	5										
P2	15	86	86	65	63	61	56	51	44	38	29	17	16	14	12	7										
P3	15	93	92	69	65	61	52	51	49	47	38	37	36	29	11	8										
P4	15	99	96	92	87	83	67	53	48	41	40	33	21	21	20	7										
P5	20	97	91	91	91	86	85	79	70	61	56	51	45	43	40	38	36	22	13	9	1					
P6	20	97	96	95	93	81	80	71	65	60	55	52	45	40	38	19	13	12	12	8	1					
P7	25	95	92	91	88	88	86	86	79	78	74	69	64	57	53	47	44	38	37	33	32	29	21	11	10	5
P8	25	94	86	86	85	74	65	63	61	58	56	56	51	50	44	44	38	29	22	17	17	16	16	14	12	7
P9	25	100	98	85	81	81	80	74	70	68	68	62	61	53	52	49	49	40	33	32	30	21	10	9	9	1

Table A3. Problem set 3 data – Processing times from *Uniform* (1, 59)

Prob Instance	No of Jobs	Processing times								
		1	2	3	4	5	6	7	8	9
1	9	51	37	26	48	57	45	7	6	47
2	9	45	22	21	44	58	26	7	4	34
3	9	22	48	17	11	25	49	20	4	1
4	9	56	15	10	47	58	30	9	6	31
5	9	44	29	10	35	51	30	4	2	31

Table A4. Problem set 3 data – Processing times from *Normal* (30; 10²)

Prob Instance	No of Jobs	Processing times								
		1	2	3	4	5	6	7	8	9
1	9	51	28	12	15	27	57	30	22	33
2	9	40	36	22	25	32	45	38	29	39

3	9	41	28	11	18	21	34	29	20	30
4	9	39	34	30	36	43	39	10	13	30
5	9	38	33	21	22	32	39	34	30	36

Table A5. Problem set 3 data – Processing times from *Exponential* (30)

Prob Instance	No of Jobs	Processing times								
		1	2	3	4	5	6	7	8	9
1	9	29	28	3	8	10	55	21	7	23
2	9	132	24	17	6	43	58	29	20	47
3	9	133	39	9	4	42	68	25	13	52
4	9	123	85	4	5	20	23	96	86	29
5	9	33	20	6	3	25	110	22	9	21

Table A6. Problem set 3 data – Processing times from *Pareto* (1.0345,1)

Prob Instance	No of Jobs	Processing times								
		1	2	3	4	5	6	7	8	9
1	9	449	8	4	3	2	5	6	153	51
2	9	18	9	3	2	10	12	11	5	6
3	9	25	7	3	2	11	53	5	4	16
4	9	40	17	3	4	5	28	15	2	18
5	9	10	7	2	3	5	108	8	4	6

r>

(¹) Corresponding Author

E-mail: i.chaudhry@uoh.edu.sa (Imran Ali Chaudhry)