Taylor & Francis
Taylor & Francis Group

International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Green permutation flowshop scheduling problem with sequence-dependent setup times: a case study

Reza Ramezanian, Mohammad Mahdi Vali-Siar & Mahdi Jalalian

To cite this article: Reza Ramezanian, Mohammad Mahdi Vali-Siar & Mahdi Jalalian (2019) Green permutation flowshop scheduling problem with sequence-dependent setup times: a case study, International Journal of Production Research, 57:10, 3311-3333, DOI: 10.1080/00207543.2019.1581955

To link to this article: https://doi.org/10.1080/00207543.2019.1581955

Published online: 23 Feb 2019.

International Journal of Production Research, 2019
Vol. 57, No. 10, 3311–3333, https://doi.org/10.1080/00207543.2019.1581955

Green permutation flowshop scheduling problem with sequence-dependent setup times: a case study

Reza Ramezanian*, Mohammad Mahdi Vali-Siar and Mahdi Jalalian

Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran

(*Received 15 February 2018; accepted 7 February 2019*)

Increasing global energy consumption, large variations in its cost and the environmental degradation effects are good reasons for the manufacturing industries to become greener. Green shop floor scheduling is increasingly becoming a vital factor in the sustainable manufacturing. In this paper, a green permutation flowshop scheduling problem with sequence-dependent setup times is studied. Two objectives are considered including minimisation of makespan as a measure of service level and minimisation of total energy consumption as a measure of environmental sustainability. We extend a bi-objective mixed-integer linear programming model to formulate the stated problem. We develop a constructive heuristic algorithm to solve the model. The constructive heuristic algorithm includes iterated greedy (CHIG) and local search (CHLS) algorithms. We develop an efficient energy-saving method which decreases energy consumption, on average, by about 15%. To evaluate the effectiveness of the constructive heuristic algorithm, we compare it with the famous augmented ε-constraint method using various small-sized and large-sized problems. The results confirm that the heuristic algorithm obtains high-quality non-dominated solutions in comparison with the augmented ε-constraint method. Also, they show that the CHIG outperforms the CHLS. Finally, this paper follows a case-study, with in-depth analysis of the model and the constructive heuristic algorithm.

Keywords: green manufacturing; energy consumption; constructive heuristic; sequence-dependent setup times; flowshop scheduling

1. Introduction

Industries are the main energy consumer. They consume about half of the world's energy (Mouzon, Yildirim, and Twomey 2007) and emit one-third of the total Carbon dioxide (CO_2) emissions (Konstantinaviciute and Bobinaite 2015). This puts higher pressure on industries to improve their energy consumption and reduce their detrimental impact on the environment.

In addition, world population growth and demands increase have been changing the patterns of energy consumption (Feng et al. 2016; Gahm et al. 2016; Dai et al. 2013), which means that industries need more energy in order to respond to new demands. So, proper use of energy is needed.

Energy is consumed in different sectors of industries. In this paper, we only consider the energy consumed by machines in the manufacturing industries where different machines perform various operations to complete an order.

Various methods have been provided to reduce energy consumption by machines in the manufacturing industries. These solutions include scheduling with a restriction on the peak power consumption of machines (Fang et al. 2013), turning off the machines in idle times (Mouzon, Yildirim, and Twomey 2007; Mouzon and Yildirim 2008; Liu et al. 2016; Lu et al. 2017), and designing power-efficient machines (Li et al. 2011; Mori et al. 2011). Sometimes, these solutions are not practical in the manufacturing industries (Ding, Song, and Wu 2016). For example, the energy used to turn on the machines is large, moreover, repeatedly turning on/off the machines may damage them (Zhang et al. 2014). So, it is necessary to provide more sustainable solutions for reducing the energy consumption. Green scheduling is so effective to improve manufacturing sustainability and reduce energy consumption, as it has become a major issue in this paper. In doing so, we propose an efficient energy-saving method to reduce energy consumption without increasing makespan.

We study *m*-machine permutation flowshop scheduling (PFS) problem with sequence-dependent setup times (SDST) that has many applications in the manufacturing industries such as textile, bottle, container, chemical compound, and plastic industries (Eren 2010; Allahverdi, Gupta, and Aldowaisan 1999; França et al. 1996; Das, Gupta, and Khumawala 1995). Also, setup times are used in the metal, pharmaceutical, and paper industries (Allahverdi, Gupta, and Aldowaisan 1999; Yang 1999).

*Corresponding author. Email: ramezanian@kntu.ac.ir

We extend a bi-objective mathematical model for the stated problem with the aim of minimising makespan as a measure of service level and minimising the total energy consumption (TEC) as a measure of environmental sustainability. This model is based on the two machine flowshop scheduling model developed by Mansouri, Aktas, and Besikci (2016).

The energy consumption of machines depends on different characteristics including machine power, operating speed, and processing time (Bansal, Kimbrel, and Pruhs 2007). The manufacturer can change the operating speed and processing time to control machine power and its energy consumption. Based on this issue, Mansouri, Aktas, and Besikci (2016) stated that the mentioned objectives have a conflict in nature. Consider Figure 5(b) that each Pareto frontier has its own schedule and sequence. In Pareto A, TEC is at its highest level since all jobs are performed at the fastest speed level. This speed level finishes the jobs as soon as possible and minimises the makespan. To reduce energy consumption, the jobs should be performed at the lower speed level. As we proceed to Pareto B, energy consumption is reduced until the jobs are performed at the lowest speed level. In Pareto B, TEC is at its lowest level since all the jobs are performed at the lowest speed level. This speed level maximises the makespan.

The contributions of this paper are presented as follows:

- Extending a bi-objective mathematical model for *m*-machine permutation flowshop scheduling problem with sequence-dependent setup times focusing on makespan and TEC,
- Developing a constructive heuristic algorithm containing an efficient energy-saving method to find the high-quality non-dominated solutions in a short time, and
- Promoting the concept of green scheduling for *m*-machine permutation flowshop scheduling problem with sequence-dependent setup times.

The remaining part of the paper proceeds as follows: Section 2 reviews the relevant literature. Section 3 presents the mathematical model. Section 4 describes the heuristic algorithm. Section 5 introduces computational results and numerical analysis. Section 6 provides the case study. Finally, section 7 finishes the paper and provides suggestions for future studies.

2. Literature review

A large and growing body of literature has investigated shop floor scheduling, however, there are still some critical issues on this topic including energy consumption and carbon footprint (Ding, Song, and Wu 2016). A recent review of the literature on energy-efficient scheduling in manufacturing, conducted by Gahm et al. (2016), found that the sustainable scheduling has been attracting the attention of manufacturers. Mouzon, Yildirim, and Twomey (2007) proposed several dispatching rules and a bi-objective mathematical programming model to minimise energy consumption and total completion time. They used a turn on/off framework to reduce the energy consumption. Mouzon and Yildirim (2008) used the same framework to minimise energy consumption and total tardiness for a single-machine problem. In these two studies, the authors did not consider the energy consumption when machines are operating. Fang et al. (2011) proposed a mathematical programming model in a flowshop scheduling with two machines and three objectives including peak power load, carbon footprint, and makespan. Fang et al. (2013) considered a flowshop scheduling problem with a restriction on peak power consumption. They presented two bi-objective mixed integer programming models to minimise peak power/energy consumption and makespan. Liu et al. (2013) investigated a permutation flowshop scheduling problem. They decomposed energy consumption into two useful and wasted parts. They developed a branch and bound algorithm to solve the problem. Liu and Huang (2014) studied a batch-processing machine scheduling problem to minimise the total weighted tardiness and carbon footprint. Dai et al. (2013) presented a bi-objective mathematical model to minimise makespan and energy consumption in a flexible flowshop scheduling problem. They used a genetic-simulated annealing algorithm to solve the problem. Zhang et al. (2014) developed a time-indexed integer programming model for a flowshop scheduling problem to minimise electricity cost and carbon footprint.

Li et al. (2016) studied a green parallel machine scheduling problem with the assumption that each machine has an exclusive energy and cleanup cost. Zhang and Chiong (2016) studied a job shop scheduling problem to optimise the total weighted tardiness and *TEC*. Ding, Song, and Wu (2016) studied a permutation flowshop scheduling problem with two objectives including makespan and carbon emissions. They developed two heuristics to solve the problem. Liu et al. (2016) presented a bi-objective mathematical model in the job shop scheduling problem to minimise total non-processing electricity consumption and total weighted tardiness. They applied turn on/off framework to save the electricity.

Mansouri, Aktas, and Besikci (2016) analysed the trade-off between makespan and *TEC* in two-machine permutation flowshop scheduling problem with sequence-dependent setup times. They developed a heuristic to obtain a trade-off analysis between the objectives. Mansouri and Aktas (2016) studied the same problem. They developed a heuristic and a multi-objective genetic algorithm to solve the problem. Both studies are limited to two-machine flowshop environment which is a special case in the flowshop scheduling problem. Mokhtari and Hasani (2017) considered energy-efficient scheduling in flexible job shop scheduling problem. They developed a tri-objective mathematical model which the third objective was minimising total energy cost of production and maintenance operations. They proposed an enhanced evolutionary algorithm for solving the problem. Che et al. (2017) studied a single machine scheduling problem to minimise TEC and maximise tardiness using power-down mechanism. They developed a mixed integer linear programming model and proposed a basic ε-constraint method to obtained Pareto fronts. Lu et al. (2017) have considered energy efficiency in m-machine flowshop scheduling with sequence-dependent setup times. They studied energy-efficient permutation flowshop scheduling problem based on turn on/off framework. In order to solve the problem, they used hybrid multi-objective backtracking search algorithm. The turn on /off framework was used in a same problem in the work of Jiang and Wang (2018). They presented a bi-objective mathematical model to minimise makespan and energy consumption. They developed a decomposition-based multi-objective evolutionary algorithm for solving the problem. Aghelinejad, Ouazene, and Yalaoui (2018) studied a single machine scheduling problem to minimise energy consumption cost. They proposed two mathematical models for formulating the problem under electricity time-varying prices, and presented heuristic and genetic algorithms for solving the problem. Biel et al. (2018) investigated the application of flowshop scheduling problem in onsite wind power generation facilities in manufacturing. They considered scenario-based uncertainty for energy supply from wind turbine and developed a bi-objective two stage stochastic programming model to minimise total weighted flow time and the expected total energy cost. Meng et al. (2018) considered the hybrid flowshop scheduling problem with unrelated parallel machines. Five mathematical models were formulated and an improved genetic algorithm was proposed for solving the problem. They applied turn on/off framework as the energy saving method. Recently, Wu and Che (2019) studied energy-efficient unrelated parallel machines scheduling problem. They presented a bi-objective mathematical model to minimise makespan and TEC. They proposed memetic differential evolution algorithm for solving the problem. Meng et al. (2019) proposed six mixed-integer linear programming (MILP) models with turning on/off framework in flexible job shop scheduling problem. The objective of all models was minimising TEC.

There exists a little attention to the flowshop scheduling problem with sequence dependent setup times considering energy consumption. As far as we know, only the works of Lu et al. (2017) and Jiang and Wang (2018) have considered energy-efficient m-machine SDST flowshop scheduling problem which both of them used turn on/off framework. There is still a need for discussion on energy-efficient m-machine SDST PFS, which is addressed in this paper.

3. Problem formulation

We consider m-machine permutation flowshop scheduling problem with sequence-dependent setup times (m-machines SDST PFS). The intended problem is denoted by $F_m/ST_{sd}/C_{max},TEC$ based on Graham's three-field notation ($\alpha/\beta/\gamma$) (Graham et al. 1979). This means that n jobs should be processed on m-machines sequentially, while involving SDST with the goal of minimising makespan (or C_{max}) and TEC. This problem is NP-hard because the problem with notation $F_2/ST_{sd}/C_{max}$ is known to be NP-hard (Gupta and Darrow 1986).

The assumptions of the problem are as follows:

- All machines are always available and there is no breakdown.
- All jobs are available at time zero.
- Each of n jobs belonging to set $J = \{1,2, \dots ,n\}$ should be processed on m-machines belonging to set $M = \{1,2, \dots ,m\}$ sequentially in the same order. Each machine can only process one job at a time.
- There is a starting setup time for the first job of the sequence on all machines.
- Setup times are not included in the processing times. They are sequence dependent, i.e. the duration of the setup times depends on both the current and immediately preceding job.
- Setup times are anticipatory, i.e. the setup times can be started before the corresponding job becomes available on the machine.
- There may be idle times for machines.
- Each machine will be turned off after completion the last job.

The notation of the problem is as follows:

Indexes

j, k Index for jobs $j, k = 1, 2, \dots, n$

i Index for machines $i = 1, 2, \dots, m$

ℓ Index for processing speeds $\ell = 1, 2, 3$ (for fast, normal and slow speeds, respectively)

Parameters

n Number of jobs

m Number of machines

sp_ℓ Processing speed factor
p_{ij} Processing time for job j on machine i
st_{ijk} Sequence-dependent setup time for changing from job j to job k on machine i (for$j = k$, st_{ijj} denotes the setup time for job j if it is the first job in the sequence)
λ_ℓ Conversion factor for processing speed ℓ
ϑ_i Conversion factor for idle time on machine i
π_i Power of machine i
M A big number
Decision Variables
s_{ij} Start time of job j on machine i
o_{ij} Setup offset for job j on machine i if job j is the first job in the sequence; $\forall i | i \neq 1$
c_{ij} Completion time of job j on machine i
θ_i Idle time on machine i
∂_j 1 if job j is the first job,0 otherwise
x_{jk} 1if job j precedes job k, 0 otherwise ($j \neq k$)
$y_{ij\ell}$ 1if job j is processed at speed level ℓ on machine i, 0 otherwise
$C_{\max}$ Makespan, the completion time of the last job denoted by n on the last machine denoted by m
TEC Total energy consumption

The bi-objective MILP model for minimising $C_{\max}$ and TEC is presented as follows:

$$Min \;\; C_{\max} \tag{1}$$

$$Min \;\; TEC \tag{2}$$

Subject to:

$$C_{\max} \geq c_{mj} \quad j = 1, \ldots, n \tag{3}$$

$$M(1 - \partial_j) + o_{ij} \geq st_{ijj} - c_{(i-1)j} \quad i = 2, \ldots, m \;\; j = 1, \ldots .n \tag{4}$$

$$s_{ij} \geq c_{(i-1)j} + o_{ij} \quad i = 2, \ldots, m \;\; j = 1, \ldots .n \tag{5}$$

$$s_{1j} \geq st_{1jj}\partial_j \quad j = 1, \ldots, n \tag{6}$$

$$c_{ij} \geq s_{ij} + \frac{p_{ij}}{sp_\ell} y_{ij\ell} \quad i = 1, \ldots, m \;\; j = 1, \ldots, n \;\; l = 1, 2, 3 \tag{7}$$

$$s_{ik} \geq c_{ij} - M(1 - x_{jk}) + st_{ijk}x_{jk} \quad i = 1, \ldots, m \;\; j, k = 1, \ldots, n \;\; \lfloor j \neq k \tag{8}$$

$$s_{ij} \geq c_{ik} - Mx_{jk} + st_{ikj}(1 - x_{jk}) \quad i = 1, \ldots, m \;\; j, k = 1, \ldots, n \;\; \lfloor j \neq k \tag{9}$$

$$\sum_{j=1}^{n} \partial_j = 1 \tag{10}$$

$$\sum_{\ell=1}^{3} y_{ij\ell} = 1 \quad i = 1, 2, \ldots, m \;\; j = 1, \ldots .n \tag{11}$$

$$\theta_i = C_{\max} - \sum_{j=1}^{n}\sum_{\ell=1}^{3} \frac{p_{ij}}{sp_\ell} y_{ij\ell} \quad i = 1, \ldots, m \tag{12}$$

$$TEC = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{\ell=1}^{3} \frac{\pi_i p_{ij\ell}\lambda_\ell}{60 sp_\ell} y_{ij\ell} + \sum_{i=1}^{m} \frac{\vartheta_i \pi_i}{60}\theta_i \tag{13}$$

$$s_{ij} \geq 0, c_{ij} \geq 0, o_{ij} \geq 0, \theta_i \geq 0, \partial j \in \{0, 1\}, x_{jk} \in \{0.1\}, y_{ij\ell} \in \{0, 1\}, \;\; C_{\max} \geq 0, TEC \geq 0 \tag{14}$$

Objective functions (1) and (2) represent the minimisation of C_{max} and *TEC*, respectively. C_{max} is the representative of service level and *TEC* is known as sustainability metric. Constraint (3) computes C_{max}. It occurs after the completion of the last job on the last machine. Since the anticipatory setup times are assumed, Constraint (4) computes the setup offset duration only for the first job on all machines except the first machine.

This allows the setup on each machine to start before the first job on the previous machine is completed. Constraint (5) ensures that the start time of a job on all machines, except the first machine, is greater than or equal to the completion time of the same job on the previous machine plus its setup offset duration (if the setup offset is exist). Constraint (6) states that a job starts on the first machine after finishing its corresponding setup time. Constraint (7) ensures that the completion time of a job on a machine is greater than or equal to its start time plus its processing time. Constraints (8) and (9) determine the sequence in such a way that the start time of a job is greater than or equal to the completion time of the preceding job on the same machine plus setup changeover. Constraint (10) ensures that there is only one first job. Constraint (11) ensures that there is only one speed level for each job on each machine. Constraints (12) and (13) compute the idle time on the machines and *TEC*, respectively.

The mentioned problem states a bi-objective optimisation problem where there is a conflict between the objectives. There is not any absolute optimal solution for this problem. A set of Pareto non-dominated solutions can be obtained based on the trade-off between C_{max} and *TEC*. A bi-objective optimisation problem is formulated as $min\ \{f_1(x), f_2(x)\}$ subject to $x\in$ S. A solution x is said to dominate solution y (written as $x \succ y$), if (): $f_p(x) \le f_p(y)$; $\forall p\in\{1,2\}$ and (II): $f_p(x) < f_p(y)$; $\exists\ p\in\{1,2\}$.

Algorithm 1: Constructive heuristic

Input: Set of machines and set of unscheduled jobs, sp_l, p_{ij}

Begin

- **Step 0. (initialization)**
- Set counter $z = 0$;
- Let set B represents all feasible solutions;
- B=[];
- Set the speed matrix at fast speed level (i.e. V_z: $[v_{ij}=sp_1]$ $i=1,\ldots,m$ $j=1,\ldots,n$);
- **Step 1. (initial solution)**
- Apply the NEHT-RB to schedule jobs based on the speed matrix V_z;
- Let seq_z represents the obtained result;
- **Step 2.**
- Apply local search (Algorithm 3) or iterated greedy algorithm (Algorithm 4) on seq_z;
- Update seq_z;
- $B = B \cup seq_z$;
- Let $z = z + 1$;
- Let $[k]$ denotes the job in position k of seq_z and $op_{i[k]}$ denotes the operation in position k on machine i;
- Let op_{n_slow} denotes the set of operations that their speed levels are not slow (i.e. $v_{i[k]} \neq sp_3$);
- **Step 3. (looking for energy-efficient solutions)**
- **While** $op_{n\_slow} \neq []$ **do**
  - Find the operation $op_{i[\alpha]}$ such that $op_{i[\alpha]} = \min_k [p_{i[k]} / v_{i[k]}],\ v_{i[k]} \in V_z \qquad i = 1,\ldots,m$;
  - Decrease the speed of operation $op_{i[\alpha]}$ by one level (i.e. $v_{i[\alpha]}$: $sp_l \rightarrow sp_{l+1}$) and update V_z;
  - Apply the NEHT-RB to reschedule jobs based on the speed matrix V_z;
  - Let seq_z represents the obtained result;
  - Apply local search (Algorithm 3) or iterated greedy algorithm (Algorithm 4) on seq_z;
  - Update seq_z;
  - Apply the energy-saving method (Algorithm 5) on seq_z;
  - Update seq_z;
  - Set $B = B \cup seq_z$;
  - Update op_{n_slow} ;
  - Let $z = z+1$;
- **End**
- **Step 3.** Apply non-dominated solution selection algorithm (Algorithm 6) on set B;

End

Output: set of non-dominated energy-efficient solutions

4. Constructive heuristic

Minimising C_{max} and *TEC* in the *m*-machines SDST PFS is modelled as MILP, and belongs to the NP-hard class. Due to the combinatorial complexity and time constraints, the heuristic methods entail to solve the large-sized problems (Hejazi and Saghafian 2005).

Nawaz, Enscore, and Ham (1983) proposed an efficient algorithm named NEH to minimise makespan in the PFS problem, but they did not consider the setup times. This algorithm has been claimed to be the most efficient one for the PFS (Ruiz and Maroto 2005). Taillard (1990) modified the NEH algorithm. He developed a speed-up method to reduce the computational complexity. Also, Ríos-Mercado and Bard (1998) modified the mentioned algorithm named NEHT-RB to minimise makespan in the PFS problem with consideration of the sequence-dependent setup times.

In this section, we develop a constructive heuristic (CH) algorithm to provide high-quality Pareto solutions, as shown in Algorithm 1. The CH algorithm is based on finding the energy-efficient solutions. This algorithm contains three levels. In the first level, all jobs are set at the fast speed level and the NEHT-RB (Algorithm 2) runs to find a solution. In the second level, the CH algorithm implements a loop to find the energy-efficient solution. In each loop, the CH algorithm selects the job with the minimum ratio of processing time to the speed level and reduces its speed by one level (i.e. fast to normal and

Algorithm 2: NEHT-RB
Input: unscheduled jobs, p_{ij}, $V_z=[v_{ij}]$, n, m
Begin
 $FS=[]$;
 Let U denotes the set of unscheduled jobs;
 Format the matrix pr such a way that $pr_{ij} = p_{ij}/v_{ij}$ $(v_{ij} \in V_z)$ $i=1,\ldots,m$ $j=1,\ldots,n$;
 Calculate $T_j = \sum_{i=1}^{m} pr_{ij}$ $j=1,\ldots,n$;
 Sort the jobs in set U to form an LPT priority list from T_j;
 While card$\{U\} > 0$ **do**
  Choose the first job h from set U;
  Compute the partial makespan $\gamma(k)$ for every position $k=1,2,\ldots,|Fs+1|$;
  Find $o = \min_k \{\gamma(k)\}$;
  Insert the job h at position o in FS;
  Remove the job h from U;
 End
End
Output: feasible solution seq_z

Algorithm 3: Local search
Input: seq_z, n
Begin
 Let $seq1 = seq_z$;
 For $j=1$ to n-1 **do**
  Swap job j and job j+1 of seq_z;
  If seq_z is not dominated by $seq1$ **then**
   $seq1 = seq_z$;
  Else
   $seq_z = seq1$;
  End
 End
End
Output: improved solution seq_z

Algorithm 4: Iterated greedy algorithm

Input: seq_z, n
Begin
    T_0 = initial temperature;
    $T_{min}=10^{-6}$;
    $Tem = T_0$;
    Let p denotes the cooling rate;
    Let $seq1 = seq_z$;
    Let $seq\_best = seq1$;
    While $Tem > T_{min}$ **do**
        r = An integer in [0,100];
        seq_R=[] ;
        $seq2=seq1$;
        $d=[r\% \times n]$;
        For i=1 to d **do**
            delete one job randomly from $seq2$ and put it in seq_R;
        End
        For i=1 to d **do**
            Let $seq_{R[i]}$ represents the i-th job in seq_R;
            Find the best sequence obtained by inserting job $seq_{R[i]}$ in all possible position of $seq2$;
            Update $seq2$;
        End
        If $seq2$ is not dominated by $seq1$ **then**
            $seq1 = seq2$;
            If $seq2$ is not dominated by seq_best **then**
                $seq\_best = seq2$;
            End
        Else if $rnd \le \exp\left(-\frac{1}{Tem} \times \left(\ln\left[\frac{c_{\max}(seq2)}{c_{\max}(seq1)}\right] + \ln\left[\frac{TEC(seq2)}{TEC(seq1)}\right]\right)\right)$ **then**
            $seq1 = seq2$;
        End
        $Tem = (1-p)\% \times Tem$;
    End
End

Output: improved solution seq_z

normal to slow). Based on new speed levels, the NEHT-RB runs to find a new solution, and ultimately our proposed energy-saving method (Algorithm 5) is implemented to improve the energy consumption without changing makespan. The second level continues until the speed level of all jobs is minimised. A Pareto solution is obtained at the end of each loop. In the first and second levels, after running the NEHT-RB, the local search algorithm (CHLS) or iterated greedy algorithm (CHIG) is implemented to improve the solution (Algorithm 3 and Algorithm 4). In the third level, the non-dominated solution selection algorithm is used to find the non-dominated solutions from the obtained solutions at the end of the second level.

In the Step 1 of the Algorithm 1, the NEHT-RB runs to find a solution based on the specified speed level. Algorithm 2 shows the pseudo code of the NEHT-RB.

We refer readers to Ríos-Mercado and Bard (1998) to figure out how to calculate the value of partial makespan $\gamma(k)$.

Two different algorithms including the local search algorithm (used in the CHLS) and the iterated greedy algorithm (used in the CHIG) are carried out to improve the solutions. Algorithm 3 shows the pseudo code of the local search algorithm, and Algorithm 4 shows the pseudo code of the iterated greedy algorithm proposed by Ruiz and Stützle (2008). In this paper, we used a modified version of the iterated greedy algorithm.

We develop the energy-saving method, which is the last procedure in the Algorithm 1. At first, this method finds the gap between two consecutive jobs on one machine, and finds the gap for one job between two consecutive machines (Step 1). Then, the energy-saving method checks whether jobs can be done at a lower speed level without changing makespan. Among these jobs, the energy-saving method selects a job that saves more energy and then reduces its speed

Algorithm 5: energy-saving method

Input: p_{ij}, $V_Z = [v_{ij}]$, seq_z ,n, m, sp_l

Begin

Step 0. (initialization)
Let m denotes the last machine and n denotes the last job;
Compute e_{ij} and form matrix $E = [e_{ij}]$ i=1,…,m j=1,…,n ;
Step 1. (find the Gap)
Let [j] and [j+1] denote the job in position j and the job in the next position j+1;
Let $Gap\_J_{i,[j]}$ denotes the gap between the job [j] and the job [j+1] on machine i ;
Let $Gap\_M_{i,[j]}$ denotes the gap for the job [j] between machine i and machine i+1;
Compute $pr_{ij} = p_{ij}/v_{ij}$ ($v_{ij} \in V_z$) i=1,…,m j=1,…,n ;
For i=1 to m **do**
  For j=1 to n-1 **do**
    $Gap\_J_{i,[j]} = E_{i,[j+1]} - pr_{i,[j+1]} - E_{i,[j]}$
  End
End
For i=1 to m-1 **do**
  $Gap\_J_{i,n}$ = inf ;
End
For i=1 to m-1 **do**
  For j=1 to n **do**
    $Gap\_M_{i,[j]} = E_{i+1,[j]} - pr_{i+1,[j]} - E_{i,[j]}$;
  End
End
For j=1 to n-1 **do**
  $Gap\_M_{m,j}$ = inf ;
End
$Gap\_J_{m,n} = 0$ and $Gap\_M_{m,n} = 0$;
Gap_{ij} = min ([$Gap\_J_{i,j}$],[$Gap\_M_{i,j}$]) i=1,…,m j=1,…,n ;
Let G denotes the matrix of machines and jobs which satisfy $Gap_{ij}>0$ (The first column shows the machines and the second column shows the related jobs to the machines);
Let β denotes the number of rows in G;
Step 2. (find the best energy saving)
While $\sum_{i\in m}\sum_{j\in n} Gap_{ij} > 0$ **do**
  $V_z' = V_z$ (i.e. $v_{ij}' = v_{ij}$ i=1,…,m j=1,…,n);
  Compute energy consumed by each job on each machine from matrix G;
  Let $EC1$ denotes the resultant matrix;
  For q=1: β **do**
    If $V_{G(q,1)\,G(q,2)} \neq SP_3$ **then**
      $\Delta_{G(q,1)\,G(q,2)} = Gap_{G(q,1)\,G(q,2)}$, $\Delta'_{G(q,1)\,G(q,2)} = P_{G(q,1)\,G(q,2)}/sp_{l+1}$;
      While $\Delta_{G(q,1)\,G(q,2)} - \Delta'_{G(q,1)\,G(q,2)} \geq 0$ **do**
        Update matrix V_z (i.e. $v_{G(q,1)\,G(q,2)} = SP_{l+1}$);
        If $V_{G(q,1)\,G(q,2)} = SP_3$ **then**
          Break the loop;
        End
        Update $\Delta_{G(q,1)\,G(q,2)}$, $\Delta'_{G(q,1)\,G(q,2)}$;
      End
    End
  End
  Compute new energy consumed by each job on each machine from matrix G;
  Let EC_2 denotes the resultant matrix;
  $EC_3 = EC_2 - EC_1$;
  Mx = max(EC_3);
  Let a and b denote the related machine and job for the element MX;
  Update matrix V_z' (i.e. $v_{ab}' = v_{ab}$);
  $pr_{ab} = p_{ab}/v_{ab}'$;
  Step 3. (create new gap)
  If $\Delta'_{G(q,1)\,G(q,2)} < \Delta_{G(q,1)\,G(q,2)}$ **then**
    $e_{ab} = e_{ab} + Gap_{ab}$;
  End
  $V_z = V_z'$ (i.e. $v_{ij} = v_{ij}'$ i=1,…,m j=1,…,n);
  Update $pr_{ij} = p_{ij}/v_{ij}$ ($v_{ij} \in V_z$) and $E = [e_{ij}]$ i=1,…,m j=1,…,n ;
  Compute $Gap\_J_{i,j}$ and $Gap\_M_{i,j}$ (similar to Step 1);
  Compute Gap_{ij} ;
  Compute matrix G;
End

End

Output: improved solution seq_z and [C_{max}, TEC]

Algorithm 6: Non-dominated solution selection

Input: set of solutions (B)
Begin
Let NDS denotes the set of non-dominated solutions;
  For each $b \in B$
    $S_b = []$;
    $n_b = 0$;
    For each $q \in B$
      If $(b \prec q)$ (i.e if b dominates q) **then**
        $S_b = S_b \cup$ q;
      Else if $(q \prec b)$ **then**
        $n_b = n_b + 1$;
      End
      If n_b=0 **then**
        NDS= NDS $\cup$ $\{b\}$;
      End
    End
  End
End
Output: set of non-dominated solutions

by one level (Step 2). If the selected job does not cover the entire gap, then the end of this job will be moved forward to the end of the remaining gap (Step 3). Therefore, new gaps are found and new chances are created for the algorithm to reduce energy consumption. Algorithm 5 shows the pseudo code of the energy-saving method. The readers can refer to the Appendix for better understanding this method.

In the Algorithm 5, the earliest completion time of job j on machine i (e_{ij}) is computed as follows:

$$e_{i0} = 0 \quad i = 1, \ldots, m \tag{15}$$

$$e_{0j} = r_j \quad j = 1, \ldots o - 1 \tag{16}$$

$$e_{ij} = \max\{e_{i-1,j}, e_{i,j-1} + s_{i,j-1,j}\} + pr \quad i = 1, \ldots, m \;\; j = \; 1, \ldots o - 1 \tag{17}$$

Where, r_j indicates the release time of job j which is assumed to be zero.

A non-dominated solution selection algorithm is implemented in the step 3 of the Algorithms 1. The fast non-dominated sort approach proposed by Deb et al. (2002) is used. In this procedure, n_b is the number of solutions dominated by the solution b, and S_b is the set of solutions dominated by the solution b. Algorithm 6 shows the pseudo code of this procedure.

The CH algorithm is compared with the famous augmented ε-constraint method proposed by Mavrotas (2009). The augmented ε-constraint method is a novel and improved version of the conventional ε-constraint method. Here, we briefly explain the implementation steps of this method for a minimisation problem. At first, a payoff table should be created. The payoff table contains the results of individual optimisation of all objective functions. In the following, one objective function is considered as the main objective. Using the payoff table, the range of each subsidiary objective function should be divided into equal intervals based on the number of grid points (g_k). Each time the problem formulation (18) is solved for obtaining Pareto solutions.

$$\begin{aligned} &Min f_1(x) + \varepsilon(s_2/r_2 + \ldots + s_{p-1}/r_{p-1} + s_p/r_p) \\ &subject\ to: \\ &x \in X(Feasible\ region) \\ &f_k(x) + s_k = e_k k = 2, \ldots.., p \end{aligned} \tag{18}$$

Where, $e_k = ub_k - (i_k \times r_k)/g_k$. There are P objective functions in formulation (18). ub_k and r_k are the upper bound and the range of the objective function k, respectively. ε is a very small number (10^{-3}–10^{-6}). $s_k$is a non-negative surplus variable and i_k is the counter of iterations. More details can be found in Mavrotas (2009). In this paper, there are two objective functions. By setting C_{max} as the main objective function, we have: $TEC + s_2 = e_2$. The feasible region is constituted by constraints (3)-(14).

5. Computational results and analysis

This section tests the model and proposed algorithms. The augmented ε-constraint method was coded in GAMS V.24.1.3 and the CH algorithm (including CHIG and CHLS) was coded in Matlab R2015a. All the work on the computer was carried out using a PC equipped with 8GB of RAM and Intel Core^{TM} i7 CPU operating by Microsoft Windows 10 (64-bit). We used instances generated by Ruiz, Maroto, and Alcaraz (2005) for the SDST PFS-*Cmax*. Also, we studied the impact of setup times. These instances were used in three sets called SDST10, SDST50 and SDST100 with the setup times uniformly distributed in [1,9], [1,49] and [1,99]. These instances are retrievable on http://www.upv.es. The instances were divided in two categories including small-sized and large-sized instances. Here, a problem with m machines and n jobs is denoted as $m \times n$. The levels and ranges of the factors used in the instances are as follows:

- Number of machines (m): 2/3/4 for the small-sized instances, and 5/10/15/20 for the large-sized instances (based on Ruiz, Maroto, and Alcaraz 2005).
- Number of jobs (n): 6/8/10 for the small-sized instances, and 10/20/50/100 for the large-sized instances (based on Ruiz, Maroto, and Alcaraz 2005)
- Ratio of setup to processing time: U(1,9), U(1,49),U(1,99) (based on Taillard 1990; Ruiz and Stützle 2008; Mansouri, Aktas, and Besikci 2016).
- $\vartheta_i Z$: 0.05, π_i: U(50,70), sp_l: {0.8, 1, 1.2}, λ_ℓ: {1,5, 1,0.6} (based on Mansouri, Aktas, and Besikci 2016).
- Various measures due to bi-objective optimisation are needed to evaluate solution methods. So, we used five performance measures as summarised in Table 1.

Table 1. Performance measures.

Measure	explanation	Formula
Number of non-dominated Pareto solutions (*CRD*)	• Larger is better	-
Solving time (*CPU time*)	• Smaller is better	-
Coverage metric (*Cov*) (Zitzler 1999)	• Larger is better • This measure compares two solution sets A and B obtained by two solution methods.	$\operatorname{cov}(A,B) = \frac{\lvert\{b \in B \mid \exists a \in A : a \succ b \text{ or } a = b\}\rvert}{\lvert B\rvert}$ (19)
Spacing metric (*SPC*) (Tan et al. 2006)	• Smaller is better • This measure shows the evenly distribution of Pareto solutions set A along the Pareto front.	$SPC = \left[\frac{1}{CRD\{A\}} \sum_{i=1}^{CRD\{A\}} (d_i - \bar{d})^2\right]^{1/2}$, $\bar{d} = \frac{1}{CRD\{A\}} \sum_{i=1}^{CRD\{A\}} d_i$ (20)
Distance metrics (*Dist1*,*Dist2*) (Czyzżak and Jaszkiewicz 1998)	• Smaller is better • These measures determine the quality of Pareto solutions set A that are relevant to reference set *Re*. • $Dist_1$ is the average distance from $x_{Re} \in R_e$ to the nearest solution in set A, and $Dist_2$ tells us about the worst case.	$Dist_1 = \frac{1}{card\{\mathrm{Re}\}} \sum_{x_{\mathrm{Re}} \in \mathrm{Re}} \{\min_{x_A \in A}\{d(x_A, x_{\mathrm{Re}})\}\}$, $Dist_2 = \max_{x_{\mathrm{Re}} \in \mathrm{Re}} \{\min_{x_A \in A}\{d(x_A, x_{\mathrm{Re}})\}\}$ (21)

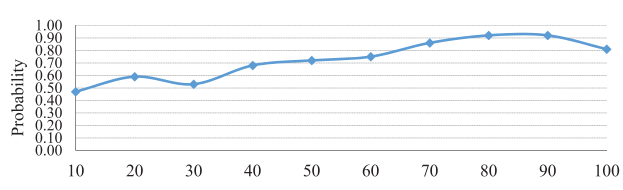

Figure 1. Probability of accepting the dominated Pareto solution based on the values of T_0.

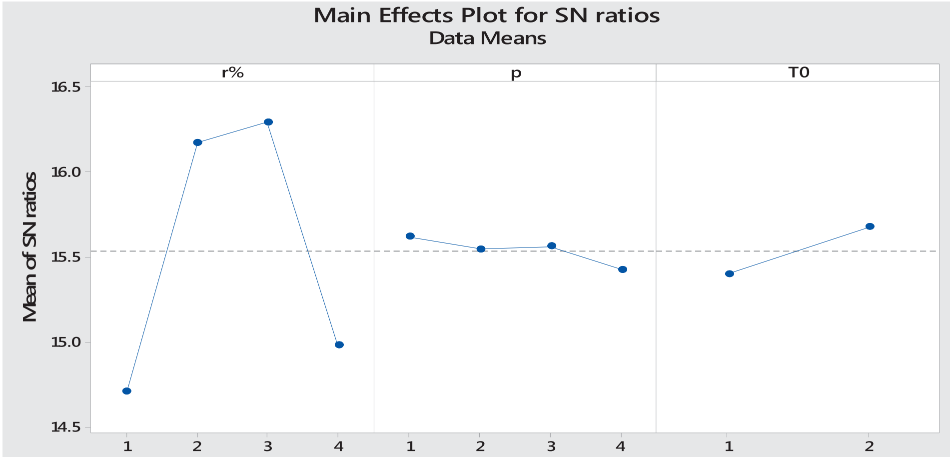

Figure 2. S/N chart for setting the CHIG parameters.

Table 2. Comparison of the measures between augmented ε-constraint, CHIG and CHLS in the small-sized instances.

		Augmented ε-constraint				CHIG					CHLS				
m × n		$Dist_1$	$Dist_2$	SPC	CPU time	$Dist_1$	$Dist_2$	SPC	CRD	CPU time	$Dist_1$	$Dist_2$	SPC	CRD	CPU time
2 × 6	SDST10	0.00	0.00	0.95	75.40	0.03	0.18	0.94	13	0.34	0.03	0.12	0.95	13	0.22
	SDST50	0.02	0.04	0.95	54.97	0.05	0.14	0.95	12	0.38	0.07	0.14	0.95	12	0.18
	SDST100	0.02	0.06	0.95	12.31	0.07	0.16	0.96	11	0.30	0.08	0.16	0.96	11	0.25
2 × 8	SDST10	0.00	0.00	0.95	5732.64	0.01	0.07	0.94	16	0.45	0.04	0.07	0.96	15	0.17
	SDST50	0.00	0.00	0.95	6667.03	0.09	0.18	0.96	15	0.45	0.09	0.18	0.96	15	0.23
	SDST100	0.01	0.05	0.95	7829.63	0.10	0.19	0.96	14	0.45	0.11	0.20	0.97	12	0.27
2 × 10	SDST10	0.01	0.06	0.93	10,201.50	0.01	0.07	0.96	20	0.88	0.04	0.09	0.94	16	0.24
	SDST50	0.02	0.06	0.95	10,068.69	0.01	0.07	0.96	12	0.97	0.08	0.18	0.95	15	0.30
	SDST100	0.02	0.11	0.96	10,032.40	0.04	0.15	0.96	12	0.88	0.12	0.22	0.97	9	0.23
3 × 6	SDST10	0.00	0.00	0.95	55.76	0.08	0.20	0.96	13	0.44	0.08	0.14	0.94	12	0.17
	SDST50	0.00	0.00	0.95	69.00	0.08	0.14	0.96	12	0.44	0.10	0.20	0.95	12	0.19
	SDST100	0.00	0.02	0.95	94.65	0.08	0.15	0.96	12	0.41	0.19	0.28	0.96	9	0.21
3 × 8	SDST10	0.00	0.00	0.95	6769.22	0.03	0.18	0.96	16	0.62	0.11	0.14	0.94	13	0.25
	SDST50	0.02	0.00	0.95	5832.73	0.06	0.13	0.96	13	0.27	0.14	0.20	0.95	12	0.53
	SDST100	0.02	0.08	0.95	4594.50	0.13	0.11	0.96	12	0.54	0.20	0.30	0.96	10	0.26
3 × 10	SDST10	0.00	0.00	0.94	9026.68	0.02	0.20	0.94	18	1.10	0.11	0.22	0.94	16	0.29
	SDST50	0.00	0.03	0.95	7466.77	0.03	0.20	0.96	15	1.08	0.14	0.24	0.95	13	0.28
	SDST100	0.03	0.08	0.95	8045.00	0.10	0.09	0.97	12	1.12	0.15	0.25	0.97	8	0.30
4 × 6	SDST10	0.00	0.01	0.95	61.87	0.04	0.16	0.95	13	0.46	0.06	0.16	0.95	11	0.22
	SDST50	0.00	0.02	0.95	54.78	0.07	0.22	0.96	13	0.42	0.08	0.23	0.95	11	0.22
	SDST100	0.00	0.04	0.95	47.17	0.10	0.20	0.96	11	0.46	0.10	0.24	0.95	10	0.20
4 × 8	SDST10	0.00	0.00	0.94	6020.53	0.06	0.16	0.94	16	0.62	0.07	0.16	0.93	16	0.20
	SDST50	0.00	0.00	0.95	7024.02	0.12	0.24	0.95	12	0.63	0.17	0.24	0.95	16	0.23
	SDST100	0.00	0.01	0.95	8972.89	0.17	0.19	0.97	9	0.65	0.24	0.32	0.97	11	0.24
4 × 10	SDST10	0.00	0.02	0.93	10,728.60	0.03	0.21	0.95	15	1.30	0.09	0.18	0.95	19	0.34
	SDST50	0.02	0.07	0.94	8375.57	0.04	0.15	0.96	13	1.32	0.12	0.19	0.96	14	0.29
	SDST100	0.02	0.07	0.95	9374.08	0.06	0.14	0.96	10	1.35	0.15	0.28	0.96	13	0.28
	Average	**0.01**	**0.03**	**0.95**	**5307**	**0.06**	**0.16**	**0.96**	**13**	**0.68**	**0.11**	**0.20**	**0.95**	**13**	**0.25**

In Equation (20), d_i is the Euclidean distance between solution i and its closest solution. In Equation (21), $d(x_A, x_{Re}) = max_{j=1,\ldots,n}\left\{\frac{f_j(x_A)-f_j(x_{Re})}{\Delta_j}\right\}$ and Δ_j is the range of objective f_j. Also, Re includes non-dominated solutions obtained by the solution methods.

5.1. Parameter setting

The CHIG algorithm contains four adjustable parameters including T_0, r, p and T_{min} (Algorithm 4). We used the Taguchi method for parameter setting. So, it is necessary to determine the initial levels of these parameters.

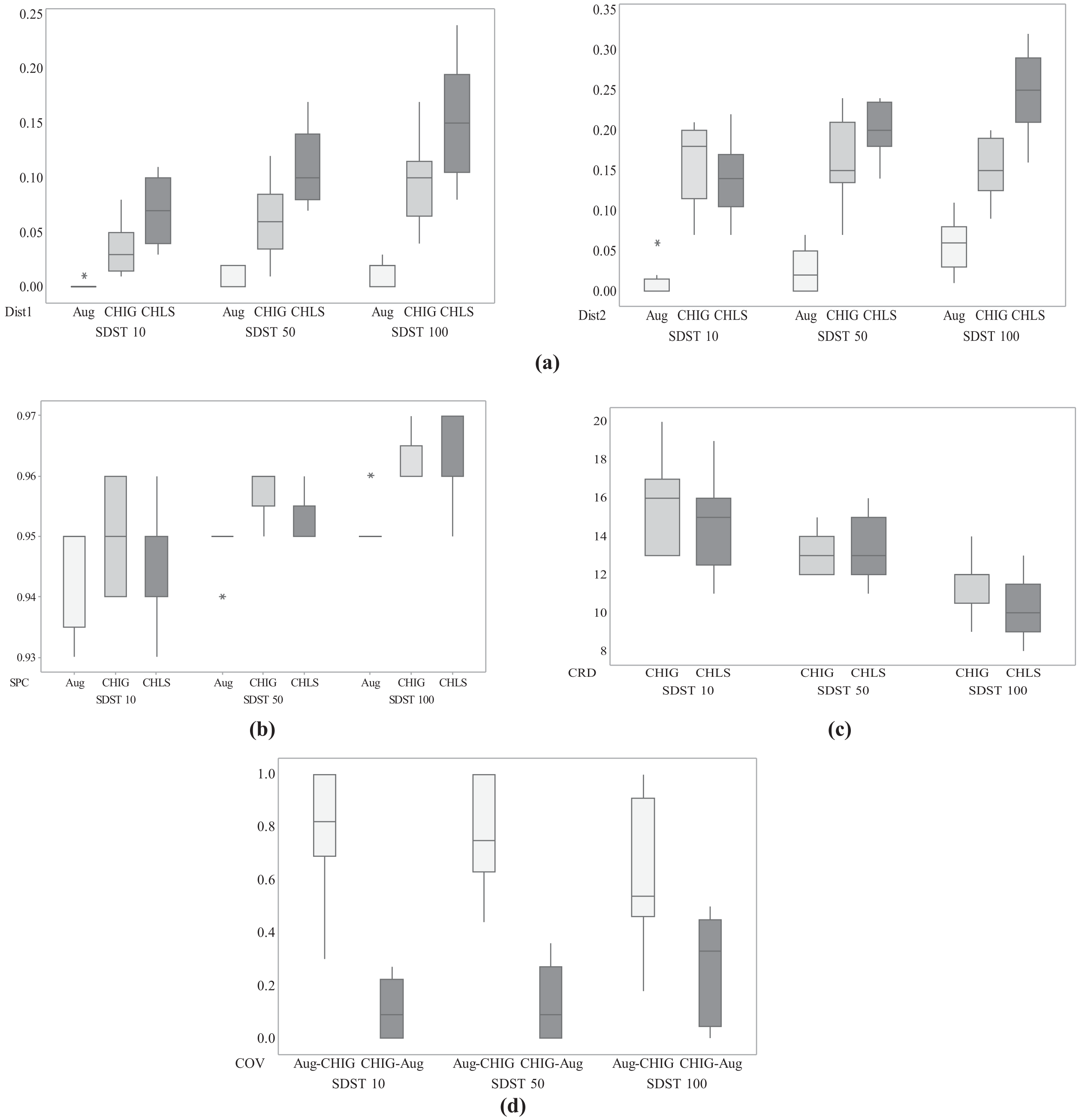

Figure 3. Results for the performance measures in the small-sized instances. (a). Results for distance metrics. (b). Results for *SPC.* (c). Results for *CRD.* (d). Results for *Cov*.

We designed an experiment to determine the initial levels of T_0. In this experiment, all parameters had constant values except for T_0, which its value is in [0,100]. In each experiment, Parameter r was considered 20 and the CHIG algorithm was implemented at T_0without temperature reduction. So, the effects of parameters *r, p* and T_{min}were eliminated while determining T_0. Accepting the dominated Pareto solutions extends the search space. Therefore, in each experiment, whenever a dominated Pareto solution was replaced by the current solution, then the probability of accepting the dominated Pareto solution was recorded. Each experiment was repeated 5 times and Figure 1 reports the average values.

T_0 with values 80 and 90 obtained the maximum probability of accepting the dominated Pareto solutions. These temperatures were taken as the initial levels of T_0. Meanwhile, a small value (10^{-6}) was selected for T_{min}. The initial levels for r and p (r $=$ 5, 10, 15, 20 and p $=$ 0.7, 0.8, 0.9, 0.95) were chosen based on the trial and error.

Table 3. Comparison of coverage metric between augmented ε-constraint and CHIG in the small-sized instances.

	SDST10		SDST50		SDST100	
m × n	Cov(Aug,CHIG)	Cov(CHIG, Aug)	Cov(Aug,CHIG)	Cov(CHIG, Aug)	Cov(Aug,CHIG)	Cov(CHIG, Aug)
2 × 6	1.00	0.00	0.64	0.18	0.50	0.36
2 × 8	0.63	0.27	1.00	0.09	1.00	0.09
2 × 10	0.30	0.27	0.72	0.17	0.54	0.50
3 × 6	0.83	0.09	1.00	0.00	0.82	0.00
3 × 8	1.00	0.00	0.62	0.36	0.45	0.33
3 × 10	1.00	0.00	0.75	0.00	0.47	0.44
4 × 6	0.82	0.18	0.92	0.09	0.18	0.46
4 × 8	0.75	0.00	1.00	0.00	1.00	0.00
4 × 10	0.82	0.17	0.44	0.36	0.54	0.20
Average	**0.79**	**0.11**	**0.79**	**0.14**	**0.61**	**0.26**

We used the Taguchi method to determine the optimal values of the parameter. This method proposed 16 experiments. Each experiment obtained different Pareto solutions. For each experiment, the performance measures were summed based on their weights to register a unique value. We used analytical hierarchy process (AHP) to weigh the performance measures based on their importance. The weights of the performance measures: $Dist_1 = 0.33$, $Dist_2 = 0.33$, $Cov = 0.33$ $SPC = 0.009$, $CPU\ time = 0.000094$ and $CRD = 0.000094$. Each experiment was repeated 5 times and the average value was reported. Figure 2 includes the results of Taguchi method based on the signal to noise (S/N) ratio approach. The selected values are $r = 15\%$, $p = 0.7$ and $T_0 = 90$.

5.2. Comparison of augmented ε-constraint, CHIG and CHLS (small-sized instances)

Nine instances were generated for each ratio of setup to processing time. We considered 10 grid points for the augmented ε-constraint method. Table 2 includes the comparison of algorithms based on the distance metrics, *SPC*, *CRD* and *CPU time*. Each problem was implemented 5 times and the average values were reported.

For all the methods, Table 2 and Figure 3(a) show that the better values of distance metrics are obtained by the smaller ratio of setup to processing time. They also indicate that the augmented ε-constraint method outperforms the two other algorithms. Based on $Dist_1$ and $Dist_2$, the CHIG obtains better solutions compared with the CHLS, especially in the larger ratio of setup to processing time.

For all the methods, Table 2 and Figure 3(b) show that the better values of *SPC* are obtained by the smaller ratio of setup to processing time. Based on this measure, they indicate that the augmented ε-constraint method has somewhat better performance compared with the two other algorithms.

For all the methods, Table 2 and Figure 3(c) show that the better value of *CRD* are obtained by the smaller ratio of setup to processing time. Based on this measure, they indicate that the CHIG has a better performance.

$Dist_1$, $Dist_2$ and *SPC* show that the augmented ε-constraint method outperforms the two other algorithms. This method, with high computation time, obtains better quality in these measures, which is not efficient in comparison with the CHIG or the CHLS. On average, the augmented ε-constraint method consumes about 14,000 times more *CPU time* than the two other algorithms. *CPU time* reported in Table 2 shows that the CHLS and the CHIG outperform the augmented ε-constraint method and there is no significant difference between the CHLS and the CHIG.

Taken together, the above results suggest that the CHIG outperforms the CHLS. So, the CHIG and the augmented ε-constraint method were compared based on coverage metric. Table 3 reports the results.

Table 3 indicates that the quality of augmented ε-constraint method is better than the CHIG. But the CHIG obtains high-quality non-dominated solutions in very smaller *CPU time* which is very valuable in shop floor scheduling. For CHIG, Figure 3(d) shows that the better values of the coverage metric are obtained by the larger setup to processing time ratio.

5.3. Comparison of CHIG and CHLS (large-sized instances)

We set the time limit of 15,000 s for the solution methods. The CHIG and the CHLS obtained solutions, but the augmented ε-constraint method did not find Pareto solutions. This method cannot solve the large-sized instances due to the complexity of the problem. This subsection analyses the CHIG and the CHLS. Table 4 includes the comparison of these two solution methods based on the distance metrics, *SPC*, *CRD* and *CPU time*. Each problem was implemented 5 times and the average values were reported.

Table 4. Comparison of the measures between CHIG and CHLS in the large-sized instances.

		CHIG					CHLS				
m × n		$Dist_1$	$Dist_2$	SPC	CRD	CPU time	$Dist_1$	$Dist_2$	SPC	CRD	CPU time
5 × 10	SDST10	0.00	0.00	0.97	19	1.43	0.01	0.06	0.97	16	0.24
	SDST50	0.00	0.05	0.96	14	1.62	0.04	0.12	0.96	14	0.31
	SDST100	0.01	0.06	0.96	12	1.60	0.10	0.17	0.96	12	0.32
5 × 20	SDST10	0.00	0.01	0.98	30	10.50	0.02	0.06	0.98	28	1.03
	SDST50	0.00	0.02	0.97	22	10.22	0.03	0.09	0.98	17	1.08
	SDST100	0.00	0.03	0.96	15	10.99	0.04	0.13	0.98	19	1.09
5 × 50	SDST10	0.00	0.00	0.99	58	191.15	0.01	0.03	0.99	54	13.05
	SDST50	0.00	0.01	0.99	42	185.83	0.01	0.05	0.99	38	13.10
	SDST100	0.00	0.01	0.98	45	187.45	0.02	0.06	1.00	28	13.32
5 × 100	SDST10	0.00	0.00	0.99	108	2071.50	0.00	0.01	1.00	94	100.78
	SDST50	0.00	0.01	0.99	65	2072.92	0.00	0.01	0.99	58	101.46
	SDST100	0.00	0.01	0.99	41	2014.77	0.00	0.02	0.99	39	101.45
10 × 10	SDST10	0.00	0.01	0.96	15	2.53	0.03	0.09	0.97	16	0.43
	SDST50	0.00	0.02	0.95	15	2.50	0.04	0.10	0.96	13	0.41
	SDST100	0.00	0.05	0.96	13	2.60	0.05	0.12	0.97	13	0.48
10 × 20	SDST10	0.00	0.00	0.98	28	18.66	0.02	0.06	0.98	22	1.99
	SDST50	0.00	0.01	0.97	25	18.93	0.05	0.09	0.98	19	2.07
	SDST100	0.00	0.01	0.96	19	19.90	0.08	0.14	0.97	18	2.04
10 × 50	SDST10	0.00	0.01	0.99	55	352.97	0.01	0.03	0.99	47	27.10
	SDST50	0.00	0.01	0.99	43	352.64	0.02	0.06	0.99	37	27.35
	SDST100	0.00	0.01	0.98	41	367.70	0.02	0.06	0.99	31	27.32
10 × 100	SDST10	0.00	0.00	1.00	98	4007.83	0.00	0.01	1.00	106	213.74
	SDST50	0.00	0.00	0.99	72	3975.12	0.00	0.02	0.99	72	215.60
	SDST100	0.00	0.01	0.99	40	4119.66	0.00	0.02	0.99	42	215.02
15 × 10	SDST10	0.00	0.00	0.97	17	3.70	0.01	0.04	0.97	17	0.57
	SDST50	0.00	0.02	0.97	16	3.90	0.03	0.11	0.97	14	0.60
	SDST100	0.00	0.03	0.95	15	4.26	0.05	0.13	0.97	11	0.60
15 × 20	SDST10	0.00	0.01	0.98	28	35.96	0.02	0.06	0.98	24	3.02
	SDST50	0.00	0.01	0.97	23	27.73	0.04	0.08	0.98	21	3.06
	SDST100	0.00	0.02	0.97	19	27.89	0.06	0.10	0.97	18	3.14
15 × 50	SDST10	0.00	0.01	0.99	63	515.62	0.01	0.02	0.99	49	41.83
	SDST50	0.00	0.01	0.99	46	518.22	0.01	0.05	0.99	39	41.86
	SDST100	0.00	0.01	0.98	30	567.10	0.02	0.06	0.98	29	42.57
15 × 100	SDST10	0.00	0.01	0.99	96	5827.67	0.00	0.01	0.99	74	328.56
	SDST50	0.00	0.01	0.99	63	6065.23	0.00	0.02	0.99	59	328.90
	SDST100	0.00	0.01	0.99	45	5955.39	0.00	0.02	0.99	42	327.70
20 × 10	SDST10	0.00	0.03	0.97	15	4.61	0.02	0.05	0.97	18	0.69
	SDST50	0.00	0.02	0.96	16	4.92	0.03	0.09	0.97	13	0.72
	SDST100	0.00	0.02	0.96	14	4.60	0.04	0.10	0.96	13	0.72
20 × 20	SDST10	0.00	0.01	0.98	27	35.47	0.03	0.07	0.98	23	3.94
	SDST50	0.00	0.01	0.98	27	35.33	0.03	0.08	0.98	22	3.98
	SDST100	0.00	0.02	0.98	23	37.65	0.05	0.09	0.98	21	4.01
20 × 50	SDST10	0.00	0.00	0.98	60	679.03	0.01	0.03	0.99	46	57.11
	SDST50	0.00	0.01	0.99	48	708.03	0.01	0.04	0.99	43	56.75
	SDST100	0.00	0.01	0.98	34	679.85	0.01	0.05	0.98	27	56.35
20 × 100	SDST10	0.00	0.01	0.99	97	7582.82	0.00	0.01	0.99	75	444.45
	SDST50	0.00	0.01	0.99	74	7710.30	0.00	0.01	0.99	69	444.64
	SDST100	0.00	0.01	0.99	52	7883.81	0.00	0.01	0.99	50	443.04
Average		**0.00**	**0.01**	**0.98**	**39**	**1352.50**	**0.02**	**0.06**	**0.98**	**35**	**77.49**

For both methods, Table 4 and Figure 4(a) show that the better value of the distance metrics are obtained by the smaller ratio of setup to processing time.

For both methods, Table 4 and Figure 4(b) show that the better value of the *SPC* are obtained by the larger ratio of setup to processing time. Table 4 and Figures 4(c) show that the better value of the *CRD* are obtained by the smaller ratio of setup to processing time. Based on the mentioned measures, the results confirm that the CHIG outperforms the CHLS.

CPU time reported in Table 4 shows that the CHLS has better performance than the CHIG. The comparison based on the coverage metric is provided in Table 5.

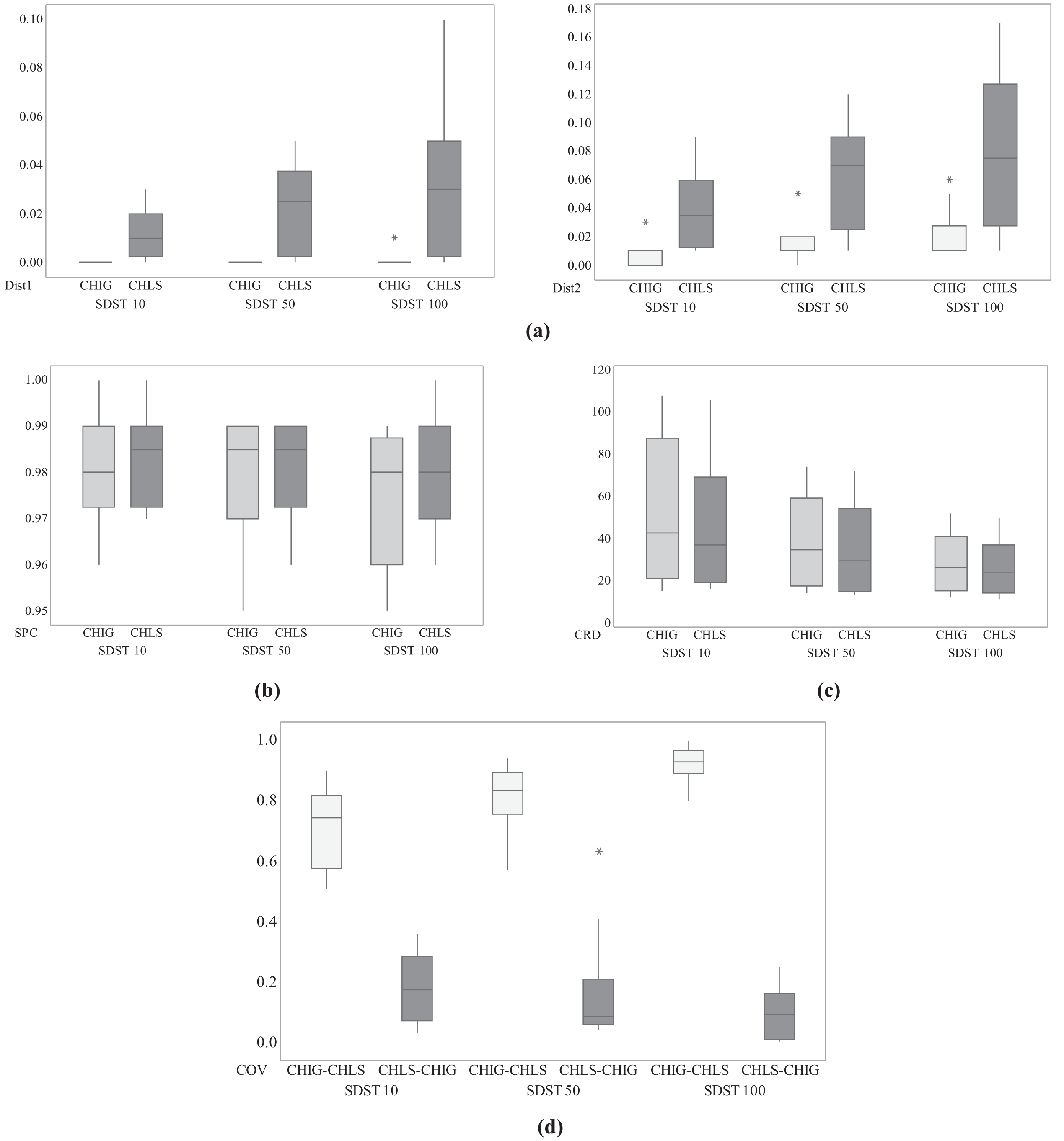

Figure 4. Results for performance metrics in the large-sized instances. (a). Results for distance metrics. (b). Results for *SPC*. (c). Results for *CRD*. (d). Results for *Cov*.

In all instances, it can be observed that Cov(CHIG,CHLS) > Cov(CHLS,CHIG). So, the CHIG performs better than the CHLS based on the coverage metric.

5.4. Statistical comparison for solution methods

We conducted Statistical tests using the IBM SPSS Statistics 22 applying a significance level of 0.05. The Kolmogorov–Smirnov test revealed that all the metrics are not normally distributed. Therefore, we use non-parametric Wilcoxon test.

Table 5. Comparison of coverage metric between CHIG and CHLS in the large-sized instances.

	SDST10		SDST50		SDST100	
m × n	Cov(CHIG, CHLS)	Cov(CHLS, CHIG)	Cov(CHIG, CHLS)	Cov(CHLS, CHIG)	Cov(CHIG, CHLS)	Cov(CHLS, CHIG)
5 × 10	0.52	0.18	0.67	0.07	1.00	0.00
5 × 20	0.90	0.03	0.88	0.09	0.89	0.00
5 × 50	0.77	0.13	0.87	0.29	0.80	0.06
5 × 100	0.82	0.30	0.81	0.63	0.94	0.12
10 × 10	0.85	0.06	0.92	0.08	0.92	0.13
10 × 20	0.72	0.03	0.94	0.04	0.84	0.00
10 × 50	0.66	0.11	0.90	0.14	0.96	0.21
10 × 100	0.55	0.36	0.80	0.06	0.97	0.18
15 × 10	0.77	0.24	0.57	0.18	1.00	0.00
15 × 20	0.80	0.07	0.90	0.04	0.89	0.10
15 × 50	0.57	0.20	0.74	0.41	0.87	0.14
15 × 100	0.51	0.33	0.81	0.05	0.97	0.17
20 × 10	0.60	0.17	0.69	0.12	0.92	0.08
20 × 20	0.81	0.07	0.87	0.07	0.91	0.04
20 × 50	0.87	0.18	0.86	0.22	0.96	0.25
20 × 100	0.68	0.32	0.81	0.06	0.94	0.04
Average	**0.71**	**0.17**	**0.82**	**0.16**	**0.92**	**0.10**

Table 6 reports the values of asymptotic significance and Z statistic for comparing the solution methods in the small-sized and the large-sized instances. The columns containing 'LS' in their title are related to the large-sized instances.

For the small-sized instances, Table 6 reveals that the augmented ε-constraint method and the CHIG have statistically significant differences in all measures, except for *SPC* under SDST100. The CHIG and the CHLS have a better performance for *SPC* under SDST10 and SDST50. The CHIG and the CHLS have statistically significant differences in all measures, except for $Dist_2$ under SDST10, *CRD* under SDST50, and *SPC* in all ratios of setup to processing time. For the large-sized instances, Table 6 reveals that the CHIG and the CHLS have statistically significant differences in all measures, except for *SPC* under SDST10.

5.5. Analysis of variance

We conducted balanced analysis of variance (ANOVA) on the performance measures. This analysis highlighted the impact of *n*, *m*, *setup* and interactions between them. Table 7 shows the results obtained from the preliminary analysis of the above performance measures. This table contains source of variation (Source), degrees of freedom (DF), F-statistics (F_value) and probability Pr(> F).

Table 7 shows that *n* and *setup* have statistically significant impacts on all measures, and *m* only has significant impact on $Dist_2$. The interaction between *n* and *setup* is statistically significant impacts on the $Dist_1$, $Dist_2$ and *CRD*.

5.6. Effectiveness of energy-saving (E.S.) method in the algorithms

We proposed the energy-saving method to improve the energy consumption without increasing makespan. This method significantly reduces fuel consumption, its costs and its detrimental impact on the environment. For example, Figure 5 is depicted for a 10 × 50 instance, in which the Pareto solutions for the CHIG and the CHLS are shown before and after implementing the energy-saving method.

In both algorithms, the energy-saving method significantly reduced *TEC*, but did not change C_{max}. Also, *CPU time* for the CHIG without implementing the energy-saving method was 352.41 s, while the energy-saving method consumed only 0.23 s. With this short time, the energy-saving method reduced the average of *TEC* by 1511 Kwh. *CPU time* for the CHLS without implementing the energy-saving method was 26.87 s, while the energy-saving method consumed only 0.22 s. Here, the energy-saving method reduced the average of *TEC* by 1126.5 Kwh. As expected, the results prove that the energy-saving method has a significant effect on the CH algorithm.

For a more careful analysis, we calculated energy consumption in the small-sized and the large-sized instances by implementing the energy-saving method and without implementing this method. The CHIG has a better performance than the CHLS, so we reported the results only for the CHIG. This algorithm obtains different Pareto solutions. Thus, in each

Table 6. Wilcoxon signed rank test for the solution methods.

		SDST 10				SDST 50				SDST 100			
		CHIG-Aug	CHLS-Aug	CHIG-CHLS	CHIG-CHLS (LS)	CHIG-Aug	CHLS-Aug	CHIG-CHLS	CHIG-CHLS (LS)	CHIG-Aug	CHLS-Aug	CHIG-CHLS	CHIG-CHLS (LS)
$Dist_1$	Z	− 2.68	− 2.67	− 2.26	− 2.99	− 2.38	− 2.52	− 2.20	− 3.07	− 2.49	− 2.67	− 2.67	− 3.08
	Assymp.sig.	0.01	0.01	0.02	0.00	0.02	0.01	0.03	0.00	0.01	0.01	0.01	0.00
$Dist_2$	Z	− 2.67	− 2.67	− 1.84	− 3.44	− 2.67	− 2.67	− 2.03	− 3.31	− 2.67	− 2.67	− 2.53	− 3.19
	Assymp.sig.	0.01	0.01	0.07	0.00	0.01	0.01	0.04	0.00	0.01	0.01	0.01	0.00
SPC	Z	− 2.38	− 2.23	− 2.06	− 1.73	− 2.59	− 2.03	0.00	− 2.24	− 1.40	− 3.33	− 1.6	− 2.25
	Assymp.sig.	0.02	0.03	0.06	0.08	0.01	0.04	1.00	0.03	0.16	0.07	0.10	0.03
CRD	Z	-	-	− 2.03	− 2.51	-	-	− 0.38	− 3.19	-	-	− 2.59	− 2.22
	Assymp.sig.	-	-	0.04	0.01	-	-	0.7	0.00	-	-	0.01	0.03
Cov	Z	− 2.68	− 2.68	− 2.66	− 3.52	− 2.67	− 2.68	− 2.68	− 3.52	− 2.07	− 2.66	− 2.67	− 3.52
	Assymp.sig.	0.01	0.01	0.01	0.00	0.01	0.01	0.01	0.00	0.04	0.01	0.01	0.00

Table 7. Analysis of variance on $Dist_1$, $Dist_2$, *SPC* and *CRD*.

Metric	Source	DF	F_value	Pr (> F)
$Dist_1$	m	3	1.170	0.349
	n	3	40.020	0.000
	Setup	2	16.700	0.000
	$m \times n$	9	1.150	0.379
	$m \times$ Setup	6	0.710	0.645
	$n \times$ Setup	6	3.940	0.011
	Error	18		
	Total	47		
$Dist_2$	m	3	4.420	0.017
	n	3	130.900	0.000
	Setup	2	46.620	0.000
	$m \times n$	9	1.020	0.461
	$m \times$ Setup	6	1.540	0.222
	$n \times$ Setup	6	5.680	0.002
	Error	18		
	Total	47		
SPC	m	3	0.390	0.763
	n	3	79.730	0.000
	Setup	2	3.920	0.039
	$m \times n$	9	1.140	0.386
	$m \times$ Setup	6	0.810	0.574
	$n \times$ Setup	6	0.530	0.779
	Error	18		
	Total	47		
CRD	m	3	0.820	0.501
	n	3	236.790	0.000
	Setup	2	52.560	0.000
	$m \times n$	9	1.270	0.319
	$m \times$ Setup	6	1.420	0.261
	$n \times$ Setup	6	13.130	0.000
	Error	18		
	Total	47		

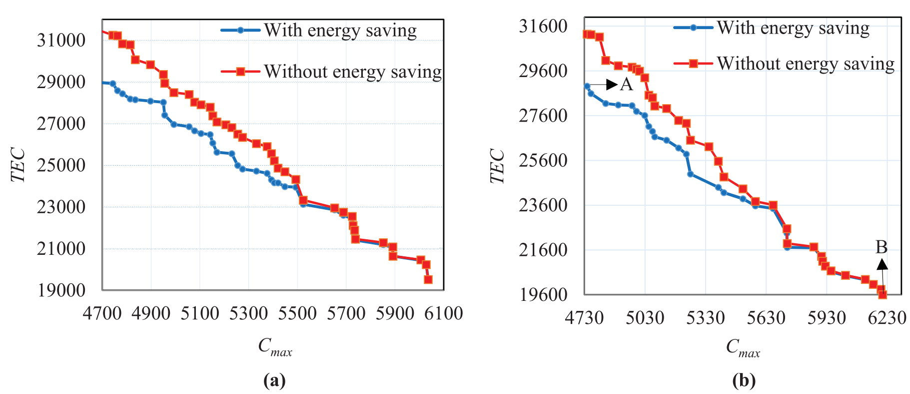

Figure 5. Implementing the energy-saving method in CHLS and CHIG algorithms. (a). CHIG algorithm for 10×50 instance. (b). CHLS algorithm for 10×50 instance.

instance, the average of *TEC* and *CPU time* across all Pareto solutions is computed. Finally, we reported the average of *TEC* and *CPU time* across all of the instances in Table 8.

The results show that the energy-saving method can reduce *TEC* by an average 9.71% and increase *CPU time* by an average 4.24%.

Table 8. Amount of change in *TEC* and *CPU time* occurring by the energy-saving method in CHIG.

	SDST 10			SDST 50			SDST 100		
	Without E.S.	With E.S.	Diff%	Without E.S.	With E.S.	Diff%	Without E.S.	Without E.S.	With E.S.
Average *TEC*	2.05E + 04	1.92E + 04	9.12	2.01E + 04	1.86E + 04	9.62	1.97E + 04	1.80E + 04	10.39
Average *CPUtime*	869.47	853.91	4.43	866.47	867.98	4.31	859.62	875.66	3.99

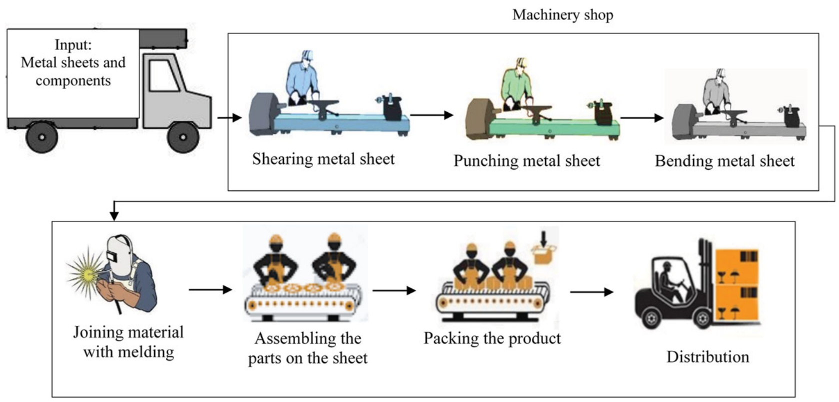

Figure 6. Description of the production process.

6. Case study: SDST flowshop with *m*-machines

The presented problem in this paper is being implemented at a factory in Khoramdasht Industrial Zone near Tehran (Iran). Extractor hood is produced at the factory. Figure 6 shows the production process of this product.

Here, we briefly explain the production process. Metal sheets and components are imported into the factory. In the stage 1, the metal sheets are cut by CNC Guillotine shear machine (Machine 1 with power 20 kW). In the stage 2, holes on the metal sheet are created by CNC punching machine (Machine 2 with power 20 kW). In the stage 3, the metal sheet is bended by the hydraulic bending machine (Machine 3 with power 7.5 kW). These three machines have three speed levels that can be controlled by technicians. In the remaining stages, machinery is not used and manpower completes the process by various tools. In this section, only the machinery shop is studied.

The extractor hood includes floor frame (Job 1 with process times 4, 9 and 4 on machines 1, 2 and 3 respectively), back frame (Job 2 with process times 2.5, 5 and 1 on machines 1, 2 and 3 respectively) and sides (Job 3 with process times 9, 12 and 15 on machines 1, 2 and 3 respectively). Each of these jobs passes the stages 1, 2 and 3. In the production process, the sequence of jobs on these stages/machines can change and the setup times vary for these stages/machines depending on the size and thickness of the metal sheets. Due to the high volume of setup times, they are not presented, but they are in the range of [2-19] minutes.

We solved the mentioned problem by the model and compared it with the published schedule by the factory. The model obtained 11 Pareto solutions. A unique Pareto solution should be selected for running and comparison. We used TOPSIS (technique for order performance by similarity to ideal solution) procedure proposed by Yoon and Hwang (1981). Figure 7 shows the published schedule by the factory and the proposed schedule by our model.

Figures 7(a) and 7(b) comprise three lines with different shapes and colours. Each type of line represents a machine that includes horizontal and vertical lines. The vertical lines represent the start and finish time of jobs on the machines, while the horizontal lines represent energy consumption for different times. For example, in the Figure 7(a), two vertical lines are plotted at times 3 and 8, indicating the start and finish times of Job 1 (J1) on Machine 1. The horizontal line connecting these two vertical lines also represents the energy consumed for J1 on Machine 1 at any moment between times 3 and 6. Figure 7(a) shows the published schedule by the factory, where the sequence of jobs is J1-J2-J3, $TEC = 18$ and $C_{max} = 45$. Also, Figure 7(b) shows the proposed schedule by our model, which has major differences with the published schedule. In the proposed schedule, the model obtained a new schedule based on the speed control, where the sequence of jobs is J3-J1-J2, $TEC = 15$ and $C_{max} = 42$. Energy consumption and makespan enhanced in the proposed schedule than the published schedule. It is worth noting that the CH algorithm presented a schedule, where $TEC = 18.95$ and $C_{max} = 50$.

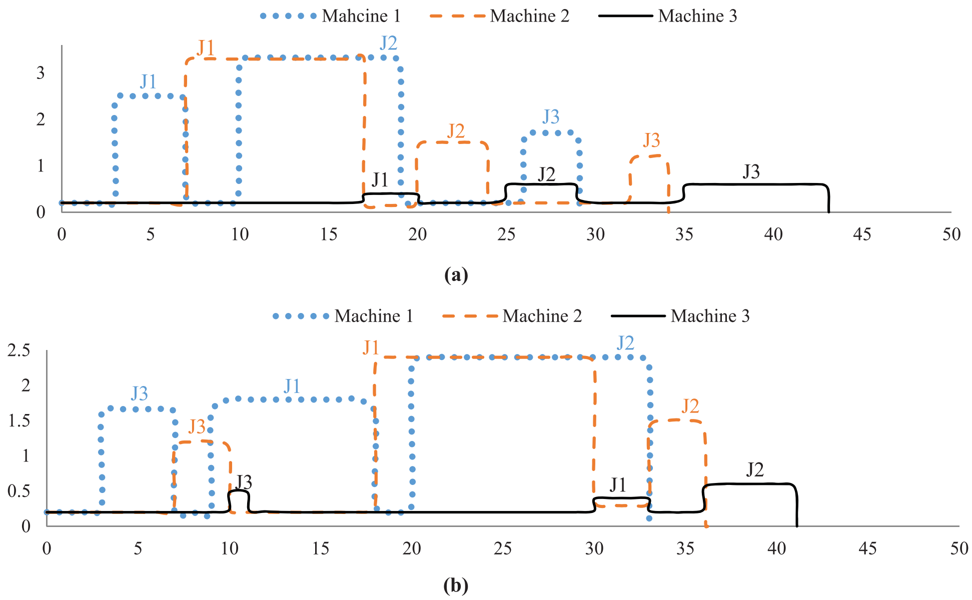

Figure 7. *TEC* and C_{max} of 3-machines and 3-jobs. (a). Published schedule by the factory. (b). Proposed schedule by the model.

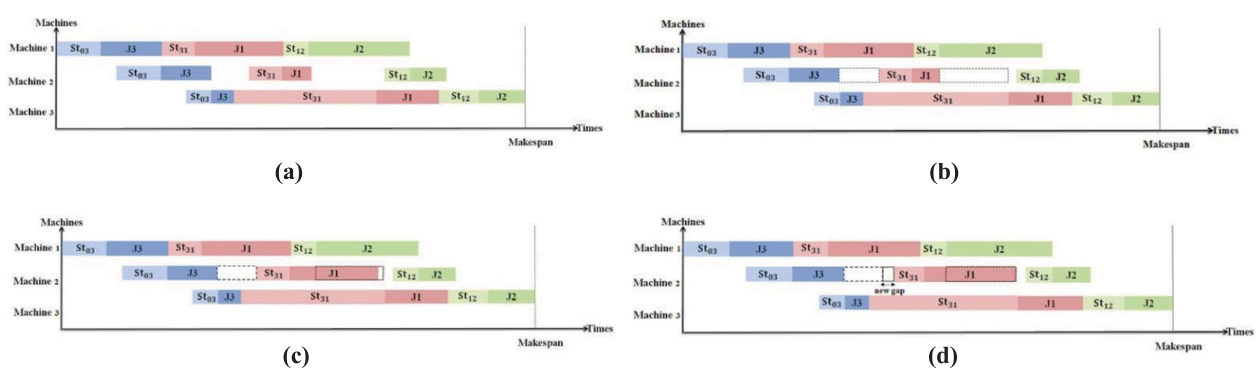

Figure 8. Implementing the energy-saving method. (a). Original sequence. (b). Finding gaps. (c). Decreasing the speed of the selected job. (d). Moving the selected job to the end of the gap.

7. Conclusion

In this paper, we studied an *m*-machine permutation flowshop scheduling problem with sequence-dependent setup times that aims to balance the makespan as a measure of service level and the TEC as a measure of environmental sustainability. A bi-objective MILP model was extended based on a pre-existing mathematical model in the literature to formulate the problem. To cope with the high computational complexity, a constructive heuristic algorithm was developed to solve the model. Two algorithms were proposed including the CHIG which exploits the iterated greedy algorithm and CHLS which exploits local search algorithm. For studying the performance of the CHIG and CHLS, the augmented ε-constraint method was applied and compared with the proposed algorithms. The results reveal that the CHIG outperforms the CHLS. Moreover, the CHIG is more efficient than the augmented ε-constraint method, because it can obtain high-quality non-dominated solutions at reasonable times. We applied Wilcoxon signed rank tests to make the results statistically convincing. In addition, we implemented an analysis of variance on the performance measures to highlight the impact of number of jobs, machines, setup times and interaction between them. In the end, for acquiring a deeper insights, a case study was conducted which demonstrated the high efficiency of the model.

The noteworthy contributions of this paper are twofold. First, we extended a mathematical model with the aim of minimising makespan and TEC in the flowshop schedulng with sequence-dependent setup times. Second, we developed a constructive heuristic algorithm containing an efficient energy-saving method that can significantly reduce the energy consumption without changing makespan. Implementing the proposed energy-saving method can bring a great impact on decreasing the detrimental environmental effects and reducing the energy costs. This method decreased TEC by an average of about 15%.

Due to the existing trade-off, decision makers can utilise the obtained Pareto solutions and choose one that is suitable for their manufacturing process. The proposed algorithms and points can be used by the manufacturers, production planners and responsible engineers.

We propose that further research should be undertaken in the following areas: Developing efficient meta-heuristic algorithms and comparing with the proposed CH algorithm, considering finite capacities between machines, applying other policies for reducing energy consumption such as restriction on peak power consumption.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Aghelinejad, M., Y. Ouazene, and A. Yalaoui. 2018. “Production Scheduling Optimisation with Machine State and Time-Dependent Energy Costs.” *International Journal of Production Research* 56 (16): 5558–5575.

Allahverdi, A., J. N. Gupta, and T. Aldowaisan. 1999. “A Review of Scheduling Research Involving Setup Considerations.” *Omega* 27 (2): 219–239.

Bansal, N., T. Kimbrel, and K. Pruhs. 2007. “Speed Scaling to Manage Energy and Temperature.” *Journal of the ACM* 54 (1): 1.

Biel, K., F. Zhao, J. W. Sutherland, and C. H. Glock. 2018. “Flow Shop Scheduling with Grid-Integrated Onsite Wind Power Using Stochastic MILP.” *International Journal of Production Research* 56 (5): 2076–2098.

Che, A., X. Wu, J. Peng, and P. Yan. 2017. “Energy-efficient bi-Objective Single-Machine Scheduling with Power-Down Mechanism.” *Computers & Operations Research* 85: 172–183.

Czyżżak, P., and A. Jaszkiewicz. 1998. “Pareto Simulated Annealing – a Metaheuristic Technique for Multiple-Objective Combinatorial Optimization.” *Journal of Multi-Criteria Decision Analysis* 7 (1): 34–47.

Dai, M., D. Tang, A. Giret, M. A. Salido, and W. D. Li. 2013. “Energy-efficient Scheduling for a Flexible Flow Shop Using an Improved Genetic-Simulated Annealing Algorithm.” *Robotics and Computer-Integrated Manufacturing* 29 (5): 418–429.

Das, S. R., J. N. Gupta, and B. M. Khumawala. 1995. “A Savings Index Heuristic Algorithm for Flowshop Scheduling with Sequence Dependent set-up Times.” *Journal of the Operational Research Society* 46 (11): 1365–1373.

Deb, K., A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan. 2002. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Ding, J. Y., S. Song, and C. Wu. 2016. “Carbon-efficient Scheduling of Flow Shops by Multi-Objective Optimization.” *European Journal of Operational Research* 248 (3): 758–771.

Eren, T. 2010. “A Bicriteria m-Machine Flowshop Scheduling with Sequence-Dependent Setup Times.” *Applied Mathematical Modelling* 34 (2): 284–293.

Fang, K., N. Uhan, F. Zhao, and J. W. Sutherland. 2011. “A new Approach to Scheduling in Manufacturing for Power Consumption and Carbon Footprint Reduction.” *Journal of Manufacturing Systems* 30 (4): 234–240.

Fang, K., N. A. Uhan, F. Zhao, and J. W. Sutherland. 2013. “Flow Shop Scheduling with Peak Power Consumption Constraints.” *Annals of Operations Research* 206 (1): 115–145.

Feng, L., L. Mears, C. Beaufort, and J. Schulte. 2016. “Energy, Economy, and Environment Analysis and Optimization on Manufacturing Plant Energy Supply System.” *Energy Conversion and Management* 117: 454–465.

França, P. M., M. Gendreau, G. Laporte, and F. M. Müller. 1996. “A Tabu Search Heuristic for the Multiprocessor Scheduling Problem with Sequence Dependent Setup Times.” *International Journal of Production Economics* 43 (2-3): 79–89.

Gahm, C., F. Denz, M. Dirr, and A. Tuma. 2016. “Energy-efficient Scheduling in Manufacturing Companies: a Review and Research Framework.” *European Journal of Operational Research* 248 (3): 744–757.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. R. Kan. 1979. “Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey.” *Annals of Discrete Mathematics* 5: 287–326.

Gupta, J. N., and W. P. Darrow. 1986. “The two-Machine Sequence Dependent Flowshop Scheduling Problem.” *European Journal of Operational Research* 24 (3): 439–446.

Hejazi, S. R., and S. Saghafian. 2005. “Flowshop-scheduling Problems with Makespan Criterion: a Review.” *International Journal of Production Research* 43 (14): 2895–2929.

Jiang, E. D., and L. Wang. 2018. “An Improved Multi-Objective Evolutionary Algorithm Based on Decomposition for Energy-Efficient Permutation Flow Shop Scheduling Problem with Sequence-Dependent Setup Time.” *International Journal of Production Research*, 1–16.

Konstantinaviciute, I., and V. Bobinaite. 2015. “Comparative Analysis of Carbon Dioxide Emission Factors for Energy Industries in European Union Countries.” *Renewable and Sustainable Energy Reviews* 51: 603–612.

Li, W., A. Zein, S. Kara, and C. Herrmann. 2011. An Investigation Into Fixed Energy Consumption of Machine Tools. *Glocalized Solutions for Sustainability in Manufacturing: Proceedings of the 18th CIRP International Conference on Life Cycle Engineering*, Technische Universität Braunschweig, Braunschweig, Germany, 268–273.

Li, K., X. Zhang, J. Y. T. Leung, and S. L. Yang. 2016. "Parallel Machine Scheduling Problems in Green Manufacturing Industry." *Journal of Manufacturing Systems* 38: 98–106.

Liu, Y., H. Dong, N. Lohse, and S. Petrovic. 2016. "A Multi-Objective Genetic Algorithm for Optimisation of Energy Consumption and Shop Floor Production Performance." *International Journal of Production Economics* 179: 259–272.

Liu, C. H., and D. H. Huang. 2014. "Reduction of Power Consumption and Carbon Footprints by Applying Multi-Objective Optimisation via Genetic Algorithms." *International Journal of Production Research* 52 (2): 337–352.

Liu, G. S., B. X. Zhang, H. D. Yang, X. Chen, and G. Q. Huang. 2013. "A Branch-and-Bound Algorithm for Minimizing the Energy Consumption in the PFS Problem." *Mathematical Problems in Engineering* 2013: 1–6.

Lu, C., L. Gao, X. Li, Q. Pan, and Q. Wang. 2017. "Energy-efficient Permutation Flow Shop Scheduling Problem Using a Hybrid Multi-Objective Backtracking Search Algorithm." *Journal of Cleaner Production* 144: 228–238.

Mansouri, S. A., and E. Aktas. 2016. "Minimizing Energy Consumption and Makespan in a two-Machine Flowshop Scheduling Problem." *Journal of the Operational Research Society* 67 (11): 1382–1394.

Mansouri, S. A., E. Aktas, and U. Besikci. 2016. "Green Scheduling of a two-Machine Flowshop: Trade-off Between Makespan and Energy Consumption." *European Journal of Operational Research* 248 (3): 772–788.

Mavrotas, G. 2009. "Effective Implementation of the ε-Constraint Method in Multi-Objective Mathematical Programming Problems." *Applied Mathematics and Computation* 213 (2): 455–465.

Meng, L., C. Zhang, X. Shao, and Y. Ren. 2019. "MILP Models for Energy-Aware Flexible job Shop Scheduling Problem." *Journal of Cleaner Production* 210: 710–723.

Meng, L., C. Zhang, X. Shao, Y. Ren, and C. Ren. 2018. "Mathematical Modelling and Optimisation of Energy-Conscious Hybrid Flow Shop Scheduling Problem with Unrelated Parallel Machines." *International Journal of Production Research*, 1–27.

Mokhtari, H., and A. Hasani. 2017. "An Energy-Efficient Multi-Objective Optimization for Flexible job-Shop Scheduling Problem." *Computers & Chemical Engineering* 104: 339–352.

Mori, M., M. Fujishima, Y. Inamasu, and Y. Oda. 2011. "A Study on Energy Efficiency Improvement for Machine Tools." *CIRP Annals* 60 (1): 145–148.

Mouzon, G., and M. B. Yildirim. 2008. "A Framework to Minimise Total Energy Consumption and Total Tardiness on a Single Machine." *International Journal of Sustainable Engineering* 1 (2): 105–116.

Mouzon, G., M. B. Yildirim, and J. Twomey. 2007. "Operational Methods for Minimization of Energy Consumption of Manufacturing Equipment." *International Journal of Production Research* 45 (18-19): 4247–4271.

Nawaz, M., E. E. Enscore, and I. Ham. 1983. "A Heuristic Algorithm for the m-Machine, n-job Flow-Shop Sequencing Problem." *Omega* 11 (1): 91–95.

Ríos-Mercado, R. Z., and J. F. Bard. 1998. "Heuristics for the Flow Line Problem with Setup Costs." *European Journal of Operational Research* 110 (1): 76–98.

Ruiz, R., and C. Maroto. 2005. "A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics." *European Journal of Operational Research* 165 (2): 479–494.

Ruiz, R., C. Maroto, and J. Alcaraz. 2005. "Solving the Flowshop Scheduling Problem with Sequence Dependent Setup Times Using Advanced Metaheuristics." *European Journal of Operational Research* 165 (1): 34–54.

Ruiz, R., and T. Stützle. 2008. "An Iterated Greedy Heuristic for the Sequence Dependent Setup Times Flowshop Problem with Makespan and Weighted Tardiness Objectives." *European Journal of Operational Research* 187 (3): 1143–1159.

Taillard, E. 1990. "Some Efficient Heuristic Methods for the Flow Shop Sequencing Problem." *European Journal of Operational Research* 47 (1): 65–74.

Tan, K. C., C. K. Goh, Y. J. Yang, and T. H. Lee. 2006. "Evolving Better Population Distribution and Exploration in Evolutionary Multi-Objective Optimization." *European Journal of Operational Research* 171 (2): 463–495.

Wu, X., and A. Che. 2019. "A Memetic Differential Evolution Algorithm for Energy-Efficient Parallel Machine Scheduling." *Omega* 82: 155–165.

Yang, W. H. 1999. "Survey of Scheduling Research Involving Setup Times." *International Journal of Systems Science* 30 (2): 143–155.

Yoon, K., and C. L. Hwang. 1981. *Multiple Attribute Decision Making: Methods and Applications*. Berlin: Springer-Verlag.

Zhang, R., and R. Chiong. 2016. "Solving the Energy-Efficient job Shop Scheduling Problem: a Multi-Objective Genetic Algorithm with Enhanced Local Search for Minimizing the Total Weighted Tardiness and Total Energy Consumption." *Journal of Cleaner Production* 112: 3361–3375.

Zhang, H., F. Zhao, K. Fang, and J. W. Sutherland. 2014. "Energy-conscious Flow Shop Scheduling Under Time-of-use Electricity Tariffs." *CIRP Annals* 63 (1): 37–40.

Zitzler, E.. 1999. "Evolutionary algorithms for multiobjective optimization: Methods and applications." PhD diss., Federal Institute of Technology Zurich.

Appendix

In order to illustrate the implementation of the energy-saving method, Figure 8(a–d) is depicted for the case study problem. The energy-saving method tries to reduce the energy consumption by adjusting the jobs at the lower speed level without changing the makespan. In doing so, two conditions must be met.

1. The selected job does not delay the same job on the next machine.
2. The selected job does not delay the next job on the same machine.

By following the both conditions, the energy-saving method is implemented as follows:

Step 1. All jobs are processed according to the sequence seq_z obtained in the Algorithm 2 and their adjusted speed level (Figure 8(a)).

Step 2. It should be checked whether jobs can be set at the lower speed level or not. Therefore, in addition to the two above conditions, there must be an idle time for setting the jobs at the lower speed level.

Step 3. In Figure 8(b), job 3 on machine 2 cannot be processed at the lower speed level, as it causes the delay for the same job on machine 3, but the speed level of job 1 on machine 2 can be decreased by one level.

If there are two or more jobs which their speed levels can be decreased, then the energy-saving method chooses the job that reduces the energy consumption more than others. Figure 8(c) shows the obtained new sequence.

Step 4. If the selected job does not fill the gap completely (like Job 1 on Machine 2), then the end of this job moves towards the end of the gap. Thereby, new gaps are found and new opportunities are being created to reduce the energy consumption (Figure 8(d)).