INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 28 (2021) 996–1017
DOI: 10.1111/itor.12566

Bi-criteria formulation for green scheduling with unrelated parallel machines with sequence-dependent setup times

Luciano P. Cota[a,b], Vitor N. Coelho[c], Frederico G. Guimarães[d] and Marcone J. F. Souza[e,*]

[a] *Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, MG 31270-901, Brazil*
[b] *Vale Institute of Technology, ITV Mining, Ouro Preto, MG 35400-000, Brazil*
[c] *Institute of Computer Science, Universidade Federal Fluminense, Niterói, RJ 24210-346, Brazil*
[d] *Department of Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, MG 31270-010, Brazil*
[e] *Department of Computer Science, Universidade Federal de Ouro Preto, Ouro Preto, MG 35400-000, Brazil*
E-mail: lucianocota@ufmg.br [Cota]; vncoelho@gmail.com [Coelho]; fredericoguimaraes@ufmg.br [Guimarães]; marcone@ufop.edu.br [Souza]

Received 10 January 2017; received in revised form 15 May 2018; accepted 17 May 2018

Abstract

Given the important role of machine scheduling in manufacturing industry, we discuss power consumption in sequencing jobs in a scheduling problem, assuming variable speed operation in machines. The problem involves defining the allocation of jobs to machines, the order of processing jobs and the speed of processing each job in each machine. This problem can be viewed as a type of green scheduling problem, dealing with sustainable use of energy consumption and environmental effects. We propose a mixed integer linear programming (MILP) model for the unrelated parallel machine-scheduling problem with sequence-dependent setup times, with independent and non-preemptible jobs, minimizing the makespan and the total consumption of electricity. Furthermore, we employ a novel math-heuristic algorithm, named multi-objective smart pool search matheuristic (or simply smart pool), for finding solutions near the Pareto front, in a restricted computational budget. As a case study, a new set of instances is created for the problem. Those instances are solved using the classical ϵ-constrained method and the smart pool method. The obtained sets of non-dominated solutions indicate the conflict between both objectives, highlighting the relevance of the suggested approach to industry. From the obtained results, it was verified that the smart pool achieved good convergence towards the true Pareto front, as indicated by the hyper-volume metric, presenting lower average time for finding solutions on the Pareto front. In small to medium size instances, the smart pool search method can achieve very good approximations of the Pareto front with less computational effort than traditional methods.

Keywords: green scheduling; multi-objective optimization; parallel machines; mixed integer linear programming

*Corresponding author.

1. Introduction

In the manufacturing sector, the predominant energy sources are natural gas and electricity. According to the U.S. Energy Information Administration, the industrial sector consumes about 54% of the world's total delivery energy. Energy is used in the industrial sector for a wide range of purposes, such as process and assembly, steam and cogeneration, process heating and cooling, lighting, heating and air conditioning for buildings. The energy consumption in industrial activities is also related to the emission of greenhouse gases in the atmosphere.

Given the growing awareness and demand for sustainability and rational use of natural resources, the research for novel techniques for an efficient use of materials and energy resources has intensified in the recent years (Mansouri et al., 2016). These research directions involve the study of materials reuse, recycling, lower energy consumption, lower generation of pollutants, conscious use of water resources, decentralization of the energy system (Coelho et al., 2017b) and investments on micro-grids powered by renewable energy sources (Coelho et al., 2016b), among others. Sustainable development relies on the efficient use of energy as well as development of energy management systems.

In Brazil, 44% of electricity consumption is in the industrial sector (Sauer et al., 2015). Moreover, three-phase induction electrical motors account for 68% of the industrial consumption (Sauer et al., 2015), motors are responsible for 35% of the electrical energy consumption in Brazil. This figure is higher in many developed countries, with electricity consumption due to motors ranging from 50% in United Kingdom to 80% in Canada. The energy generation in Brazil is composed mainly of hydroelectric dams and thermoelectric plants to supply additional energy when needed. Despite the increasing use of distributed generation in the last years, the participation of renewable energy sources is still timid, below 5% of national production of energy. Thus, reducing total energy consumption (*TEC*) in industry is paramount in order to decrease greenhouse gase emissions and environmental impact and also indirectly reduce production costs in industry.

Machine scheduling plays an important role in manufacturing industry, allowing production managers and production engineers to maximize productivity by allocating the jobs to the resources available in an optimal way. In this paper, we discuss electric energy consumption in sequencing jobs in a machine scheduling problem. We propose a multi-objective formulation in which one of the objectives is to minimize the *TEC* in the scheduling while processing all the required jobs. In particular, the variant called unrelated parallel machine scheduling problem with setup times is addressed. We select this problem due to its practical application in industry, since it is a reality in many industrial processes of different companies such as textile, chemical, semiconductor, inks and papers (Franca et al., 1996; Randhawa and Kuo, 1997; Radhakrishnan and Ventura, 2000; Kurz and Askin, 2001; Rabadi et al., 2006).

Formally, the unrelated parallel machine scheduling problem with setup times involves a set $N = \{1, \ldots, n\}$ of independent and non-preemptible jobs and a set $M = \{1, \ldots, m\}$ of machines. The job processing times and setup times are dependent on the sequence. In this work, we focus on allocating all the jobs to the machines seeking to minimize the makespan (the time to finish the last job in the schedule) and the total consumption of electricity by machines. The scheduling problem considering only the minimization of makespan is usually represented by $R_M|S_{ijk}|C_{\max}$, according to the three-field standard notation introduced by Graham et al. (1979), where R_M indicates unrelated machines, S_{ijk} indicates that the setup times are sequence-dependent and $C_{\max}$ defines the *makespan*.

We introduce in this paper the bi-objective variant herein defined as $R_M|S_{ijk}|(C_{\max}, TEC)$, where TEC represents the total consumption of electricity spent by machines as an additional objective. The model will be further detailed in Section 3.

The proposed formulation $R_M|S_{ijk}|(C_{\max}, TEC)$ has the following characteristics: (a) each job should be allocated to only one machine;[1] (b) each job has a processing time to complete in a machine; (c) there is a setup time to calibrate each machine to process a job. The setup time depends on the order of allocation of jobs in the machine; (d) each machine has a given power consumption according to speed of operation to process the job in the machine and (e) there is a discrete set of processing speeds to process a job in a machine. In this problem, the objective is to allocate all jobs in the machines, to define their sequences on machines and to find their processing speed, minimizing the makespan and the total consumption of electricity.

The objectives makespan and total consumption of electricity have great importance in this problem. The minimization of the makespan usually implies in good use of the machines available in the production (Pinedo, 2008). Additionally, the minimization of the total consumption of electricity implies in reduction of costs for the industries and the conscious use of environmental resources. The conflicting nature of the objectives $C_{\max}$ and TEC is due to the trade-off between maximizing production at higher speeds of operation, which would lead to greater power consumption in scheduling. The majority of electrical motors used in industry are induction machines, given their high efficiency, the low maintenance costs and the variable-speed control allowed by modern frequency inverters (variable frequency drive method). In variable speed applications, changing frequency is a common method to control the speed of an induction motor, particularly with the available power electronic devices, a variable supply frequency to the stator can be used to control speeds. However, we have to make sure that the voltage–frequency ratio remains constant to maintain constant flux in the machine. Therefore, the use of a machine in a greater speed leads to a corresponding increase in power input (Sen, 2013; Wildi, 2013). These characteristics are assumed in the formulation $R_M|S_{ijk}|(C_{\max}, TEC)$.

Research involving sustainability in scheduling problems have been called green scheduling (Yu, 2010; Bampis et al., 2015; Mansouri and Aktas, 2016; Mansouri et al., 2016). In recent papers, the makespan has been minimized with other objectives related to consumption of energy for other machine scheduling problems. Mansouri et al. (2016) and Mansouri and Aktas (2016) minimize the makespan and the total consumption of energy in different problems. Also, Wang et al. (2016) and Cheng et al. (2017) minimize the makespan and the total energy costs in a different formulation.

The problem tackled here has great importance, because it is present in production processes of many industries (Rabadi et al., 2006; Pereira Lopes and de Carvalho, 2007) and belongs to the $\mathcal{NP}$-hard class of problems, since it is a generalization of the parallel machine scheduling problem with identical machines and without setup times (Karp, 1972; Garey and Johnson, 1979). Machine scheduling problems with setup times have been attracting attention of several researchers, because of its importance to the industry and the challenges imposed for its resolution (Zhu and Wilhelm, 2006). These problems can be found, for example, in furniture manufacturing (Agnetis et al., 2001) and paint factories (Mansouri, 2005). Several works in the literature dealt with the unrelated parallel machine scheduling problem with setup times, but few approaches had analysed

[1]Except the fictitious job 0, which is usually allocated at the beginning of each machine in the implementations of mathematical models.

energy consumption. In this sense, this present work contributes with a more sustainable view regarding classical machine schedules. For this purpose, a mixed integer linear programming (MILP) mathematical model is designed, implemented and analysed. In particular, the proposed model is inspired from the work of Mansouri et al. (2016), in which the authors propose a mathematical model for green scheduling in a flow shop problem.

In order to tackle the proposed MILP model and generate good quality sets of non-dominated solutions, in a restricted computational time, we use the multi-objective smart pool search matheuristic (or simply smart pool for short), recently proposed by Coelho et al. (2016a). The main core of this method is to transform a multi-objective problem into multiple single-criterion weighted-sum problems. A new set of instances is designed in order to validate the proposal. Obtained solutions from the classical ϵ-constrained method and the smart pool are presented and compared. The trade-off between both objective functions is verified, showing the ability of the proposed framework to provide efficient schedules with low energy consumption.

The rest of this paper is organized as follows. Section 2 presents a literature review. The proposed mathematical model is introduced in Section 3. Smart pool method is detailed in Section 4. In Section 5, the results on computational experiments are reported and discussed. Finally, in Section 6, the conclusions and future work are presented.

2. Literature review

In Glass et al. (1994), the authors proposed a genetic algorithm, a simulated annealing (SA), and a tabu search (TS) for a problem without setup time. A TS is proposed by Srivastava (1997) for the same problem. In Weng et al. (2001), seven constructive heuristics are proposed to solve a problem with machine-independent sequence-dependent setup times with the objective of minimizing the total weighted completion time. In Kim et al. (2002, 2003), the authors proposed a SA to minimize the total tardiness in a problem with machine-independent sequence-dependent setup times. In Logendran et al. (2007), six different TS are proposed to minimize the total tardiness in a problem with machine-independent sequence-dependent setup times.

In the following, we review some works that approached the problem of minimizing only the makespan (defined as $R_M|S_{ijk}|C_{\max}$). In Al-Salem (2004), the authors developed a partitioning heuristic that combined a constructive heuristic, a local search strategy and a heuristic based on the asymmetric traveling salesman problem. In the work of Rabadi et al. (2006), a metaheuristic for randomized priority search (Meta-RaPS) is compared with the exact method applied to solve the proposed mathematical model. An Ant Colony Optimization (ACO) algorithm is proposed by Arnaout et al. (2010), for solving an special case of the problem where the distance between jobs and machines is considerably large. In Ying et al. (2012), a restricted simulated annealing (RSA) method is proposed, mainly based on an efficient elimination of inefficient jobs. Two different genetic algorithm are proposed by Vallada and Ruiz (2011), as well as a new mathematical model. A bee colony algorithm is developed by Lin and Ying (2014). In Avalos-Rosales et al. (2013), the authors propose a new mathematical model built upon some ideas in the model of Vallada and Ruiz (2011). This new model has better performance than the models of Rabadi et al. (2006) and Vallada and Ruiz (2011). In the work of Arnaout et al. (2014), an improvement in the ACO of Arnaout et al. (2010) is done. In the works of Haddad et al. (2014, 2015) and Cota et al. (2014a, 2014b), the authors proposed different algorithms that combine the iterated local search (Lourenço et al., 2003) and variable neighbourhood descent (Hansen et al., 2008) metaheuristics, as well as other techniques. The so-called AIRP algorithm, designed by Cota et al. (2014a), is the one that presented better performance. In Avalos-Rosales et al. (2013, 2015), the authors proposed improvements in the mathematical model presented in Vallada and Ruiz (2011). In Santos et al. (2019), the authors presented an analysis of several stochastic local search methods, with previous calibration of parameters. In Tran et al. (2016), the authors proposed two hybrid models based on logic-based Benders decomposition and branch and check. An adaptive large neighbourhood search with learning automata is proposed by Cota et al. (2017).

Sustainable use of electricity in scheduling problems is still under-explored in the literature. In Zhang et al. (2014), a time-indexed mathematical model for solving flow shop scheduling problem is proposed for dealing with two objectives: minimizing the electricity costs and carbon emissions. This paper analysed the variation in the electricity costs during consumption time. In the work of Ding et al. (2016), the authors dealt with another flow shop scheduling problem for minimizing makespan and total carbon emissions, using a multi-objective iterated greedy algorithm. In Mansouri et al. (2016), a two-machine flow shop problem, with setup times, is tackled aiming at minimizing makespan and *TEC*. For its resolution, a mathematical model and a constructive heuristic were proposed for analysing trade-off between objectives.

In Wang et al. (2016), the authors treat the single machine batch scheduling problem with non-identical job sizes, time-of-use (TOU) electricity prices and different energy consumption rates of the machine. The objectives are to minimize the makespan and to minimize the total energy costs. The authors proposed a mathematical model for the problem and implemented an exact epsilon-constraint method. In Liu et al. (2016), the authors proposed a mathematical model for the bi-objective optimization problem that minimizes the total non-processing electricity consumption and total weighted tardiness in a job shop. For its resolution, they developed a multi-objective genetic algorithm based on NSGA-II (Deb et al., 2002). A two-machine sequence-dependent permutation flow shop problem with the objectives to minimize the makespan and the energy consumption is treated in Mansouri and Aktas (2016). The authors proposed constructive heuristics and multi-objective genetic algorithms for the resolution. A single-machine batch scheduling problem with TOU electricity prices is treated in Cheng et al. (2017). The objectives of the problem are to minimize the makespan and the total electricity cost. The authors proposed a mathematical model and an ϵ-constraint method.

In this paper, we propose a multi-objective formulation for green scheduling, considering the unrelated parallel machine scheduling problem with setup times with the objectives of minimizing the makespan and the *TEC*. The objective *TEC* is very important because there is great concern about the conscious use of energy in the present times, and these resources imply in high production costs for industries. This work intends to encourage the use of green scheduling in the problem and to provide a mathematical model that can be used in practical cases in the industry.

3. Mathematical model

In contrast to the classical approaches for the unrelated parallel machine scheduling problem with setup times, as shown within the works of Rabadi et al. (2006), Vallada and Ruiz (2011) and

Avalos-Rosales et al. (2013, 2015), which only minimize makespan, new characteristics are added to the proposed formulation. These features are necessary for the consideration of energy consumption, such as the variable processing speed and the machine power at different speeds. The inclusion of these characteristics is inspired by the work of Mansouri et al. (2016) for a flow shop scheduling problem. The new mathematical model proposed in this work is also based on the one introduced by Avalos-Rosales et al. (2015). It is worth mentioning that independent and non-preemptible jobs are considered.

The machine scheduling problem addressed in this study ($R_M|S_{ijk}|(C_{\max}, TEC)$) has the following detailed characteristics:

1. There are a set N of jobs, a set M of machines and a set L of different modes of operation, each one related to a corresponding speed of operation and power consumption.
2. The machines $i \in M$ are independent parallel, so the processing time of a job is different in each of the machines.
3. All jobs are available to be processed starting from time 0.
4. Each job $j \in N$ should be allocated to a single machine $i \in M$.
5. The time to process the job $j \in N$ on a given machine $i \in M$ is the processing time p_{ij}.
6. There is a setup time S_{ijk} to process the job $k \in N$ after the job $j \in N$ on the machine $i \in M$, in this order.
7. Each machine $i \in M$ has a power consumption π_i at normal speed of operation.
8. Each mode of operation $l \in L$ has a multiplication factor of speed v_l at normal operation. Thereby, the real processing time of the job $k \in N$ on the machine $i \in M$ and on mode operation l is $\frac{p_{ik}}{v_l}$.
9. Each mode of operation $l \in L$ has a multiplication factor of power λ_l at normal operation. This factor is used to calculate the energy consumption on each mode of operation.

The goal of the problem involves the allocation of all the jobs (N) on the machines (M) using the modes of operation (L), seeking to minimize two objectives. The first objective is to minimize the *makespan*, which is the time to finish the last job in the schedule. The second objective is to minimize the consumption of electricity, which is calculated using the power consumption, the multiplication factor of power, the multiplication factor of speed and the processing time.

The notation of the new mathematical model is given as follows:

- $M = \{1, \ldots, m\}$: set of machines, being m its cardinality;
- $N = \{1, \ldots, n\}$: set of jobs with n representing the number of jobs;
- $L = \{1, \ldots, q\}$: set of q different modes of operation, each mode is related to a corresponding speed of operation and power consumption (later called just modes of operation);
- $N_0 = N \cup \{0\}$: set of jobs with inclusion of the fictitious job 0;
- p_{ij}: processing time of job j in machine i [minutes];
- S_{ijk}: setup time necessary for allocating job k in machine i after job j [minutes];
- B: big constant;
- π_i: power consumption of machine i at normal speed of operation [kW];
- v_l: multiplying factor of speed at normal operation, with $l \in L$;
- λ_l: multiplying factor of power at normal speed, with $l \in L$.

We assume that the factors v_l and λ_l are the same on all machines. The values of the factors are inspired in the study proposed in Ahilan et al. (2013), which is also used by Mansouri and Aktas (2016) and Mansouri et al. (2016).

As an example, if the job j has processing time of 100 units of time in a certain machine and the factor $v_l = 0.8$, then the time to processing j becomes 125 units of time. If the factor is $v_l = 1.2$, then the time to processing j becomes 83.3. Similarly, machine speed and power influence the energy consumption, thus in our model, we assume that the higher the speed, the greater the energy consumption is. At normal speed of operation, $v_l = 1.0$ and $\lambda_l = 1.0$, and then the processing time of job j in machine k is given by p_{jk}, while the power consumption is given by π_k. The factor v_l is a non-decreasing function of λ_l. The relation between the factors v_l and λ_l is given below:

$$v_l \text{ and } \lambda_l = \begin{cases} v_l = 1 \text{ and } \lambda_l = 1, & \text{normal speed of machine operation,} \\ 0 < v_l < 1 \text{ and } 0 < \lambda_l < 1, & \text{speed slower than normal, then the machine} \\ & \quad \text{consumes less power,} \\ v_l > 1 \text{ and } \lambda_l > 1, & \text{speed greater than normal, then the machine} \\ & \quad \text{consumes more power.} \end{cases}$$

Decision variables used in the mathematical model are

$$x_{ijkl} = \begin{cases} 1, & \text{if job } k\text{, with mode of operation } l\text{, is allocated immediately after job } j \text{ in machine } i \\ 0, & \text{otherwise.} \end{cases}$$

Auxiliary variables used in the model are as follows:

- C_j: completion time of job j;
- O_i: completion time of machine i;
- $C_{\max}$: maximum processing time of all machines (makespan);
- TEC: total energy consumption (kWh).

This model uses a fictitious job 0 allocated at the beginning of each machine. This job has processing times $p_{i0} = 0\ \forall i \in M$ and setup time $S_{i0k} = 0\ \forall i \in M, \forall k \in N$. The mathematical model can be seen from Equations (1) to (12):

$$\min C_{\max} \tag{1}$$

$$\min TEC \tag{2}$$

Subject to:

$$\sum_{i=1}^{m} \sum_{\substack{j=0 \\ j \neq k}}^{n} \sum_{l=1}^{q} x_{ijkl} = 1 \quad \forall k \in N \tag{3}$$

$$\sum_{i=1}^{m} \sum_{\substack{k=1 \\ j \neq k}}^{n} \sum_{l=1}^{q} x_{ijkl} \leq 1 \quad \forall j \in N \tag{4}$$

$$\sum_{k=1}^{n}\sum_{l=1}^{q} x_{i0kl} \leq 1 \quad \forall\, i \in M \tag{5}$$

$$\sum_{\substack{k=0\\k\neq j}}^{n}\sum_{l=1}^{q} x_{ijkl} - \sum_{\substack{h=0\\h\neq j}}^{n}\sum_{l=1}^{q} x_{ihjl} = 0 \quad \forall j \in N, \forall i \in M \tag{6}$$

$$C_k - C_j + B\left(1 - x_{ijkl}\right) \geq S_{ijk} + \frac{p_{ik}}{v_l} \qquad \forall j \in N_0, \forall k \in N, \; j \neq k, \forall l \in L, \; \forall i \in M \tag{7}$$

$$C_0 = 0 \tag{8}$$

$$\sum_{j=0}^{m}\sum_{\substack{k=1\\k\neq j}}^{n}\sum_{l=1}^{q}\left(S_{ijk} + \frac{p_{ik}}{v_l}\right) x_{ijkl} = O_i \quad \forall i \in M \tag{9}$$

$$C_{\max} \geq O_i \quad \forall i \in M \tag{10}$$

$$TEC \geq \sum_{i=1}^{m}\sum_{j=0}^{n}\sum_{\substack{k=1\\j\neq k}}^{n}\sum_{l=1}^{q}\left(\lambda_l \times \frac{\pi_i}{60} \times \frac{p_{ik}}{v_l}\right) x_{ijkl} \tag{11}$$

$$x_{ijkl} \in \{0,\ 1\} \qquad \forall j \in N_0, \forall k \in N, \; j \neq k, \forall i \in M, \; \forall l \in L. \tag{12}$$

The objective of the model is to minimize makespan (1) and *TEC* (2). Constraints (3) ensure that each job will be allocated to only one machine, as well as having a predecessor, and working a unique operation mode. Constraints (4) define that each job will have, at most, one successor job. Analogously, in constraints (5) it is guaranteed that the dummy job has at most one successor job. Constraints (6) ensure right order for allocating job, if a job j is predecessor to a job k, there must be another job that comes before job j. Constraints (7) are responsible for calculating the accumulated time of each job, if $x_{ijkl} = 1$, then, accumulated time of k (C_k) is equal to the one of j (C_j) plus the sum of the time for preparing the machine (S_{ijk}) and the processing time of job k at mode of operation l ($\frac{p_{ik}}{v_l}$). If $x_{ijkl} = 0$, constant B will ensure the constraints are satisfied. In constraints (8), the accumulated time of the fictitious job 0 is ensured. Those described in constraints (9) perform calculations of the accumulated costs for each machine, this cost is given by the sum of the setup time and the processing time of all jobs allocated to a given machine. Constraints (10) define the value of $C_{\max}$. The calculation of the *TEC* is performed by constraint (11). To perform this calculation, one needs to consider the processing times (p_{ik}/v_l), the power input of each machine at normal speed

Table 1
Setup times of machines M_1 and M_2

M_1	1	2	3	4	5	6	M_2	1	2	3	4	5	6
1	0	1	8	1	3	9	1	0	5	1	6	1	7
2	4	0	7	3	7	8	2	6	0	7	7	6	2
3	7	3	0	2	3	5	3	7	6	0	9	6	9
4	3	8	3	0	5	2	4	3	7	3	0	1	7
5	8	3	7	9	0	5	5	5	8	5	6	0	9
6	8	8	1	2	2	0	6	7	4	1	7	9	0

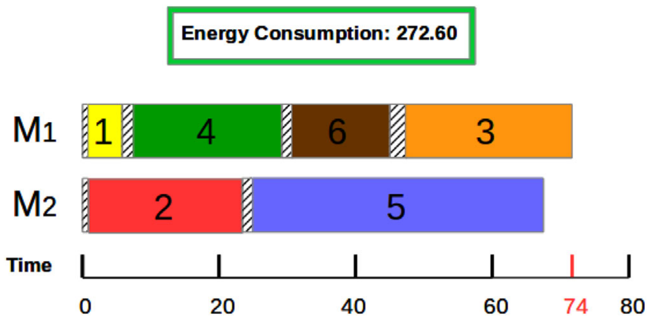

Fig. 1. Optimal solution for $R_M|S_{ijk}|C_{\max}$.

of operation (given by π_i), and the multiplying factors (λ_l) and (v_l). The machine power is divided by 60 because the power is given in kilowatt, while the processing time is given in minutes and the *TEC* is calculated in kilowatt hour. This constraint is equivalent to the integral calculation of the energy consumption. Finally, constraints (11) define the binary variables.

The proposed mathematical model has n^2mq binary variables and $n + m + 2$ continuous variables. The model also has $2n + 3m + nm + 2n^2mq + 2$ constraints. Therefore, adding the consideration of variable speed leads to a linear increase in the number of binary variables and constraints.

In order to illustrate this problem, we choose randomly an instance with six jobs, two machines and one operation mode (normal speed of machine operation with $v_l = 1$ and $\lambda_l = 1$). The processing times and the power consumption at normal speed of operation of the machines $M1$ and $M2$ are given below. Setup times of those machines are shown in Table 1:

$$M1 : p_{1j} = \{1, 87, 28, 32, 38, 9\}, \pi_1 = 70,$$

$$M2 : p_{2j} = \{4, 21, 68, 17, 43, 48\}, \pi_2 = 179.$$

This instance was solved with the IBM ILOG CPLEX, version 12.5. Figure 1 shows the optimal solution for the problem $R_M|S_{ijk}|C_{\max}$ by considering just the minimization of makespan. Jobs 1, 4, 6 and 3, in this order, are allocated to machine 1 and jobs 2 and 5, in this order, are allocated to machine 2. Figure 2 shows the optimal solution for the problem $R_M|S_{ijk}|TEC$ by minimizing just the *TEC*. In this case, jobs 6, 4, 1, 3 and 5, in this order, are allocated to machine 1 and only jobs 2, is allocated to machine 2.

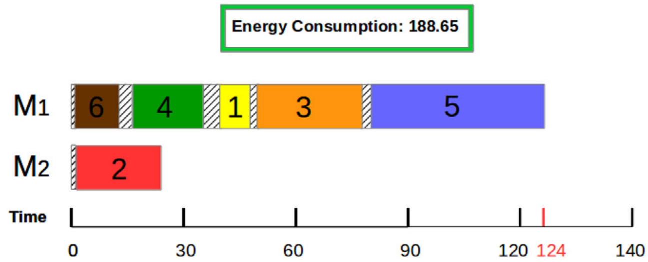

Fig. 2. Optimal solution for $R_M|S_{ijk}|TEC$.

It can be observed that there is a great variability in the allocation and order of the jobs and the values of the objectives. In Fig. 1, the makespan is 74 units of time and the *TEC* is 272.60 units of energy. In contrast, in Fig. 2, the makespan is 124 units of time and the *TEC* is 188.65 units of energy. This example shows the conflicting nature of the two objectives in the scheduling. An interesting trade-off could be found in which the jobs are processed in different speeds in different machines.

4. Multi-objective smart pool search matheuristic

The multi-objective smart pool search matheuristic method (or smart pool) (Coelho et al., 2016a) has the goal of finding non-dominated solutions near the Pareto front. This matheuristic method consists in solving multi-objective mathematical models by optimizations done by mathematical solvers, within a predefined time limit for each resolution called by the procedure. In order to achieve its goal, different mixed integer linear programming (MILP) problems are generated with different weights for aggregating both objective functions involved in the proposed formulation.

Algorithm 1 presents the smart pool search pseudo-code.

Algorithm 1: Multi-objective smart pool search matheuristic

Input: Number of weight vectors w_{max} and matheuristic solver time limit $timeLim$
Output: Set of non-dominated solutions X_e

$W \leftarrow$ `weightVectorGeneration`(w_{max});
$X_e \leftarrow \emptyset$;
foreach $\mathbf{w} \in W$ **do**
  $model \leftarrow$ MILP model with the vector of weights $\mathbf{w}$ for the objectives;
  $poolSol, poolEval \leftarrow$ Solver($model$,$timeLim$);
  $poolEval \leftarrow$ evaluations of each solution $s \in poolSol$ regarding the objective functions;
  for $n \leftarrow 1$ ***to*** $|poolSol|$ **do**
    $addSolution(X_e, poolSol[n], poolEval[n])$
  end
end
return X_e;

In this paper, the weights were generated using the method proposed by Scheffé (1958). This proposition defines the $\{r, w_{\max}\}$-simplex lattice, used in the experiments with mixtures. In the context of optimization problems, r is the number of objectives. In this method, $\binom{r+w_{\max}-1}{w_{\max}}$ points in r-dimensional space are generated, with $w_{\max}+1$ points equally spaced in the boundary of the simplex and satisfying the condition: $||\mathbf{w}||_1 = w_1 + w_2 + \cdots + w_r = 1$. This method for weight generation is very popular in multi-objective evolutionary algorithms such as NSGA-III and MOEA/D (Zhou et al., 2011; Deb and Jain, 2014). It is more efficient than the one used in the original version of the smart pool (Coelho et al., 2016a). Other approaches for weight vector generation could have been used too, see for instance Meneghini and Guimarães (2017).

In Algorithm 1, line 4 generates the single criteria (weighted sum aggregation) mathematical model with desired weights w_1 and w_2 for the objectives makespan and *TEC*, respectively. Subsequently, the line 5 calls the black-box solver for optimizing the model. During the search and optimization of each MILP problem, usually, different feasible solutions can be found. In this sense, the pool of feasible solutions obtained in each run of the optimizer are returned and stored in the set *poolSol*. Parameter *timeLim* sets the maximum time that can be spent by the solver within the search of each MILP problem with a unique set of weights.

Finally, line 8 checks which of these solutions are non-dominated, filtering *poolSol* with a simple Pareto dominance, considering the procedure addSolution, detailed by Lust and Teghem (2010) and depicted in Algorithm 2. The procedure keeps updating the set of non-dominated solutions X_e, initially empty (line 2 of Algorithm 1), filtering the solutions extracted from the BB tree (or other method used by the mathematical solver) and creating an approximation of the Pareto front.

Algorithm 2: addSolution

Input: Set of non-dominated solutions X_e; Solution s, and its evaluations $z(s)$
Output: X_e

Added $\leftarrow$ true;
forall the $x \in X_e$ **do**
  if $z(x) \preceq z(s)$ **then**
    Added $\leftarrow$ false;
    Break;
  end
  if $z(s) \prec z(x)$ **then**
    $X_e \leftarrow X_e \setminus x$;
  end
end
if *Added* $=$ *true* **then**
  $X_e \leftarrow X_e \cup s$;
end
return X_e;

Table 2
Characteristics of the problem instances

Parameters	Levels	Based on
Number of jobs (n):	6, 8, 10, 12, 15	Vallada and Ruiz (2011)
Number of machines (m):	2, 3, 4, 5	Vallada and Ruiz (2011)
Modes of operation (q):	3 or 5	Mansouri et al. (2016) and Ahilan et al. (2013)
Processing time (p_{ij}):	$U[1, 99]$	Vallada and Ruiz (2011)
Sequence dependent setup time (S_{ijk}):	$U[1, 9]$, $U[1, 49]$, $U[1, 99]$, $U[1, 124]$	Vallada and Ruiz (2011)
Machine power (π_i):	$U[40, 200]$	–
Multiplying factor of speed (v_l):	1.2, 1.1, 1, 0.9, 0.8	Mansouri et al. (2016) and Ahilan et al. (2013)
Multiplying factor of power (λ_l):	1.5, 1.25, 1, 0.8, 0.6	Mansouri et al. (2016) and Ahilan et al. (2013)

5. Computational results

Computational experiments were done considering a computer with Core i7, 1.9 GHz, 6 GB RAM and operational system Ubuntu 16.04. The mathematical model was implemented in the OptFrame 2.3 (Coelho et al., 2011) and solved with the IBM ILOG CPLEX, version 12.5.

5.1. Instances generation

In order to run the batch of experiments, a new set of instances was created for the multi-objective formulation $R_M|S_{ijk}|(C_{\max}, TEC)$. This set was inspired by the instances proposed by Vallada and Ruiz (2011) and available at SOA (2011). The new set of instance has 80 problems, considering combinations of 6, 8, 10, 12 and 15 jobs with 2, 3, 4 and 5 machines. The combinations of 6, 8, 10 and 12 jobs have 3 modes of operation ($q = 3$), representing normal speed, slow and fast, while the combinations of 15 jobs have five modes of operation ($q = 5$). Table 2 summarizes the characteristics of the test instances and the source of the original instance.

5.2. Method used for solving the proposed mathematical model

The ϵ-constrained method and the smart pool (described in Section 4) were used for solving the proposed model. In the ϵ-constrained method, one of the objective functions is kept and the other ones are placed as inequality constraints. Thus, the ϵ-constrained method converts a multi-objective problem into a single-objective one with additional constraints. The following lines (13) and (14) present an example of the method for a minimization problem using the ϵ-constrained strategy.

$$\min f_1(x) \tag{13}$$

Subject to:

$$f_i(x) \leq \epsilon_i, \quad i = 2, \ldots, r. \tag{14}$$

Table 3
Average time to solve single-objective versions of the problem (ϵ-constrained method)

Instance set	Minimizing the $C_{\max}$	Minimizing the TEC
6	**0.113**	0.333
8	**0.313**	1.360
10	**1.082**	42.744
12	**6.989**	556.885
15	**224.263**	2402.626

Table 4
Smart pool proposed variants

Acronym	$w_{\max}$	*timeLim* (seconds)	Number of MILP problems
$SmartPool_1$	10	20	10
$SmartPool_2$	20	20	20

In this problem, x is the decision variable and there is a total of r objectives. The objective f_1 is defined as the most important, being considered the only objective in this formulation. The other objectives f_i (with $i = 2, \ldots, r$) are converted into constraints. The parameter ϵ_i is the upper bound of objective f_i. Different solutions on the Pareto-optimal front can be achieved upon variations of ϵ_i.

For application in this work, initially we generate for each instance one single-objective mathematical model minimizing only the $C_{\max}$ and another single-objective mathematical model minimizing only the TEC. In each case, the other objective was kept as a constraint. A similar procedure was done in Toro et al. (2017), in the context of multi-objective vehicle routing problems, in order to identify which objective should be selected in the ϵ-constrained method. Table 3 presents the average time (in seconds) for the executions of the single-objective mathematical models. The best results are highlighted in bold. These values are grouped by instances with the same number of jobs.

It can be observed that the model minimizing the $C_{\max}$ is much more efficient. Thereby, in the ϵ-constraint method the objective of minimizing makespan ($C_{\max}$) is defined as the most important goal and the TEC is considered as a constraint.

The implementation of the ϵ-constrained method is detailed next. Initially, the maximum point f_b was defined, as well as the minimum point f_a for objective TEC, for each instance. For calculating f_a, the single-objective mathematical model minimizing only the TEC was used. Similarly, to calculate the f_b, the single-objective mathematical model minimizing only the $C_{\max}$ was used. Thus, it was possible to calculate the corresponding values of the TEC. Ten different values for ϵ were created by an uniform distribution ($U[f_a, f_b]$) for each test problem.

For the smart pool method, two different configurations were defined ($SmartPool_1$ and $SmartPool_2$), ensuring the method to solve approximately the same amount of problems generated by the ϵ-constrained method. The configuration $SmartPool_1$ solves 10 problems and the configuration $SmartPool_2$ solves 20 problems. Each problem could be optimized with a maximum time limit of 20 seconds. This time limit was obtained previously by empirical tests. We have obtained a

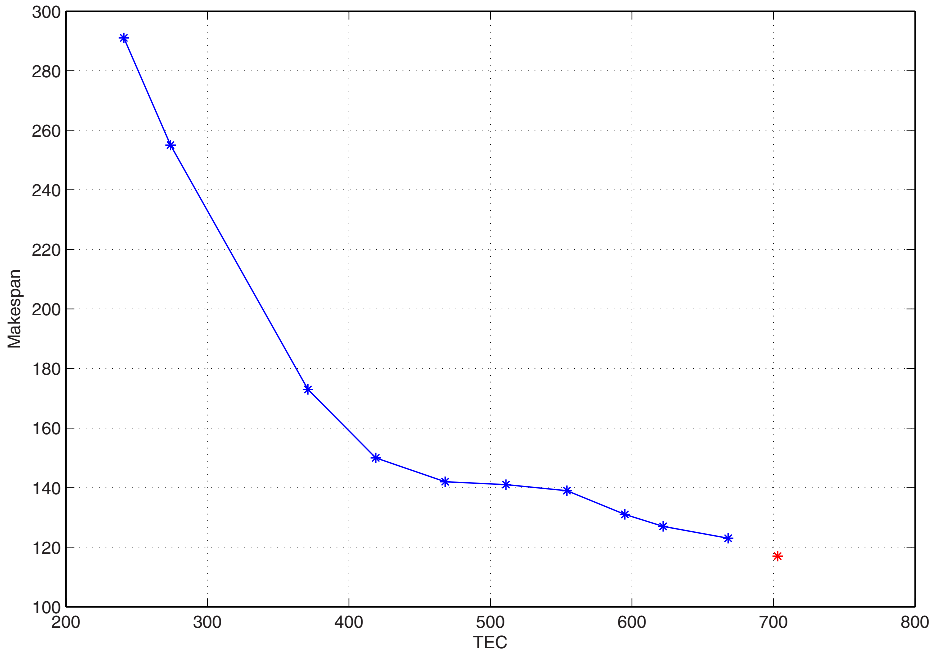

Fig. 3. Example 1: Pareto front for a given test instance taken at random.

minimum time limit to solve each problem satisfactorily. Table 4 shows the characteristics of these configurations.

5.3. Analysing the trade-off between both objectives

Next, the estimates of the Pareto front found by the ϵ-constrained method, considering two instances taken at random, are presented. These sets of non-dominated solutions are shown in Figs. 3 and 4. Blue points indicate the Pareto solutions for the $R_M|S_{ijk}|(C_{\max}, TEC)$ and red points show the optimal solution for the $R_M|S_{ijk}|C_{\max}$, considering the makespan as a single objective to be minimized.

It is observed that there is great variability in the values of the two goals. The point in red colour is important for highlighting that only this solution would be analysed in a single-objective version of the problem. As depicted in Fig. 3, for example, about 700 units of energy would be required to implement the solution that minimizes the makespan (hence maximizing production). These graphics illustrate that both objectives ($C_{\max}$ and TEC) are conflicting and emphasize the importance of this multi-objective approach for the decision maker. By selecting a different solution on the Pareto front, the decision maker could choose to sacrifice some units of makespan in order to achieve less energy consumption.

To better illustrate the conflict between both objectives, Table 5 shows the maximum and minimum values found, for each objective, by the ϵ-constrained method. These values are grouped by instances

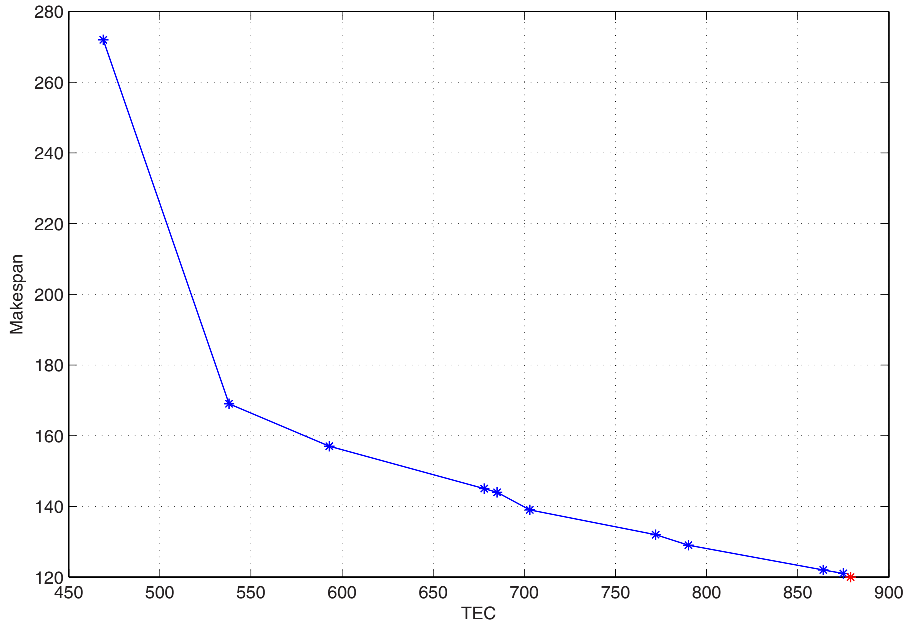

Fig. 4. Example 2: Pareto front for a given test instance taken at random.

Table 5
Maximum and minimum values of the solutions found by the ϵ-constrained method

	$C_{\max}$			TEC		
Instance set	Max	Min	Difference	Max	Min	Difference
6	383.0	23.0	360.0	713.0	57.0	656.0
8	359.0	36.0	323.0	1223.0	146.0	1077.0
10	524.0	55.0	469.0	1125.0	212.0	913.0
12	634.0	59.0	575.0	1738.0	173.0	1565.0
15	469.0	54.0	415.0	1649.0	220.0	1429.0

with the same number of jobs. It can be verified that there is a great difference between the maximum and minimum values of both goals. In the sets involving 12 jobs, there is a difference of 1565 units regarding total electricity consumption, showing the importance of this goal for the problem.

5.4. Comparing results between ϵ-constrained method and smart pool

Table 6 shows the average time taken for finding the non-dominated solutions in the first batch of experiment of the ϵ-constrained method. The average time is given by the time spent (in seconds) divided by the number of non-dominated solutions found. These values are grouped by the instances with same number of jobs.

Table 6
Average time for finding non-dominated solutions – ϵ-constrained method and smart pool configurations

Instance set	ϵ-constrained	$SmartPool_1$	$SmartPool_2$
6	0.208 (± 0.073)	**0.079** (± 0.026)	0.150 (± 0.058)
8	0.644 (± 0.289)	**0.312** (± 0.329)	0.779 (± 0.468)
10	1.780 (± 1.421)	**0.807** (± 0.231)	1.277 (± 0.459)
12	3.994 (± 3.781)	**1.371** (± 0.712)	1.953 (± 1.063)
15	54.663 (± 88)	**6.998** (± 5.104)	8.172 (± 5.009)

Table 7
Results of indicator HV

Instance set	ϵ-constrained	$SmartPool_1$	$SmartPool_2$
6	0.708 (± 0.086)	0.810 (± 0.070)	**0.821** (± 0.061)
8	0.768 (± 0.093)	0.814 (± 0.118)	**0.826** (± 0.116)
10	0.791 (± 0.055)	0.845 (± 0.078)	**0.858** (± 0.064)
12	0.863 (± 0.093)	0.934 (± 0.072)	**0.942** (± 0.068)
15	0.833 (± 0.088)	0.884 (± 0.113)	**0.909** (± 0.087)

The best results are highlighted in bold. It is observed that the $SmartPool_1$ method achieved the best results in all cases. It can also be verified that the ϵ-constrained method took a lot of time to solve the instances with 15 jobs.

In order to check the quality of the obtained sets of non-dominated solutions, two indicators of quality were considered: the hyper-volume (HV) and the coverage between two sets (CS) (Zitzler and Thiele, 1999). HV indicator of a set P calculates the volume of the region between the points $p_i \in P$ and a reference point rp. For each solution $p_i \in P$, a hyper-cube hy_i is constructed according to the reference point rp. In maximization problems, the reference point usually is (0,0), while in minimization ones, the upper limit should be determined for each objective. The HV of an estimate of the Pareto front is the sum of hyper-cubes each solution set contains. In this current work, the reference point rp is the worst solution for both objectives, which are calculated in the same way that these values are used for calculating the ϵ values.

The CS indicator determines the percentage of solutions of another set (X'') that a particular set (X') dominates. Equation (15) shows the calculation of the coverage indicator:

$$CS(X', X'') = \frac{|a'' \in X''; \exists a'' \in X' : a' \ covers \ a''|}{|X''|}. \tag{15}$$

Operation a' *covers* a'' determines that a' dominates a'' or a' is equal to a''. The results of CS indicator are mapped from [0,1]. If the result of CS is equal to 1 implies that all points X'' are dominated or equal to those in X'.

Tables 7 and 8 show the obtained results for both indicators. Values are also reported according to groups of instances with the same number of jobs.

The best results are highlighted in bold. For the HV metric, $SmartPool_2$ method achieved the best results in all cases; however, the values are very similar to each other. This indicates that all

Table 8
Results for indicator CS

Instance set	ϵ-constrained × $SmartPool_1$	$SmartPool_1$ × ϵ-constrained	ϵ-constrained × $SmartPool_2$	$SmartPool_2$ × ϵ-constrained
6	0.067 (± 0.129)	**0.111** (± 0.136)	0.043 (± 0.0701)	**0.132** (± 0.13)
8	0.097 (± 0.096)	**0.116** (± 0.098)	0.057 (± 0.0978)	**0.116** (± 0.093)
10	0.063 (± 0.054)	**0.092** (± 0.122)	0.053 (± 0.0611)	**0.090** (± 0.112)
12	**0.062** (± 0.065)	0.053 (± 0.069)	0.061 (± 0.0384)	**0.076** (± 0.096)
15	**0.125** (± 0.088)	0.027 (± 0.057)	**0.084 (± 0.0638)**	0.045 (± 0.081)

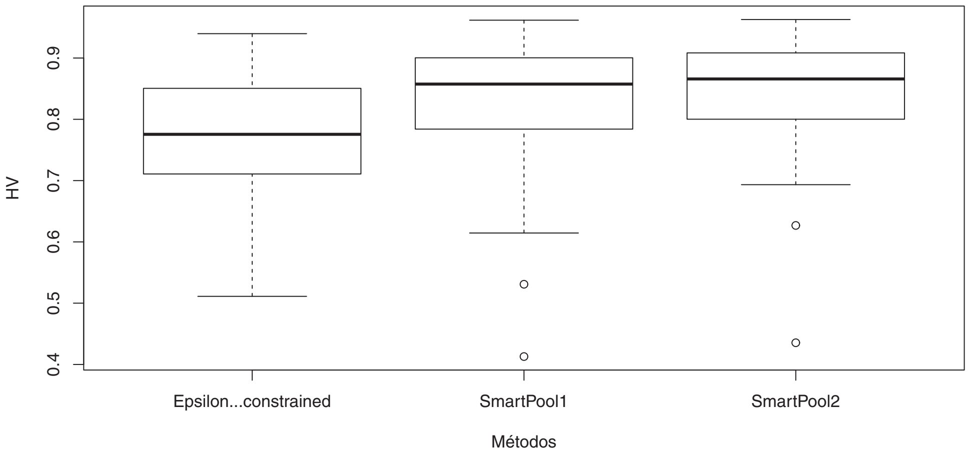

Fig. 5. Box plot of results for the HV indicator.

methods presented good convergence to the true Pareto front of the problems. This is expected for the ϵ-constrained method, since it relies on the exact solution provided by the solver. In smart pool method, the solver does not necessarily run until optimality, because of the time limit parameter. Nevertheless, for the instances considered, the convergence is quite good provided the values of HV metric. Regarding the CS indicator, the smart pool method also got the best results in most cases with the two configurations. But it was observed that both methods achieved low coverage relative to each other, thus suggesting that both methods match high-quality solutions relative to their estimated Pareto sets. Figures 5 and 6 show box plot graphs.

Figure 7 provides a graphical analysis of the ϵ-constrained method and smart pool, using two instances randomly selected. One instance has 12 jobs and 3 machines and the other has 15 jobs and 2 machines. A new variant of smart pool was created, the $SmartPool_3$, with $w_{\max} = 100$ and $timeLim = 600$ (seconds). This proposed combination of parameters may be able to show the evolution of the method with more weights and long time limit running for each solved MILP. Figures 7 and 8 show the results of the methods. The results of ϵ-constrained method are optimal.

It can be seen that the smart pool search was able to find a very good approximation of the Pareto front with a much larger number of solutions.

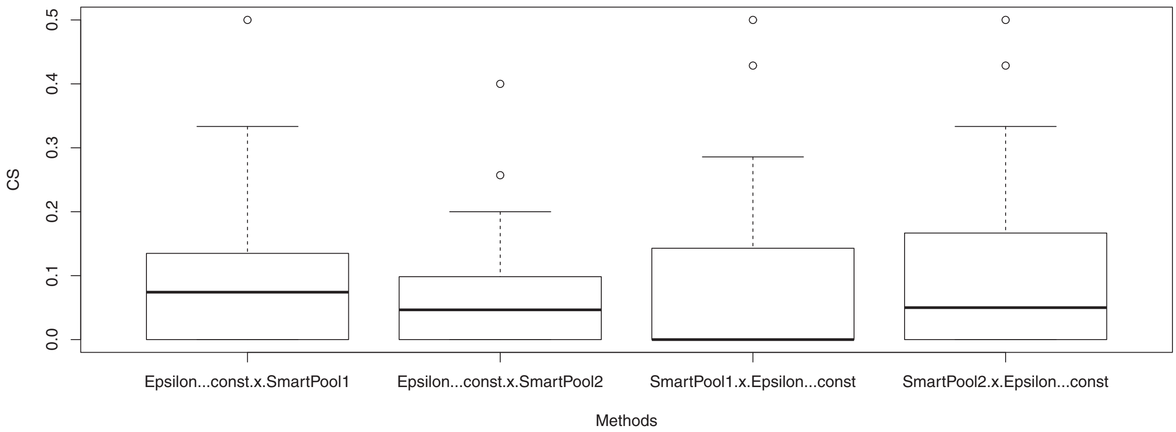

Fig. 6. Box plot of results for the CS indicator.

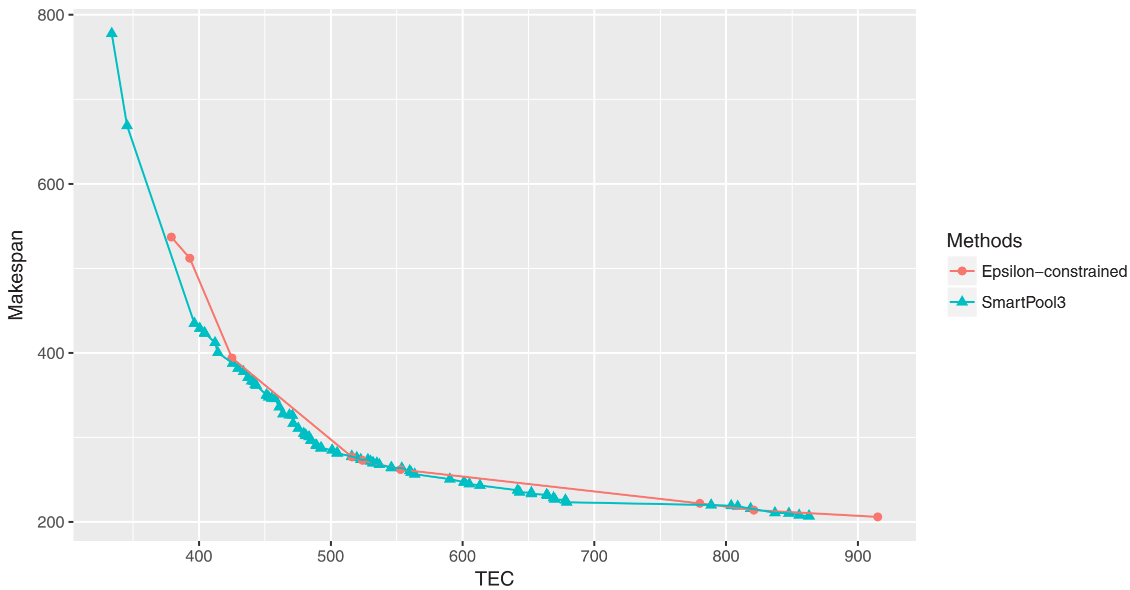

Fig. 7. Obtained Pareto fronts for the ϵ-constrained method and $SmartPool_3$ for an instance with 12 jobs and 3 machines.

6. Conclusions

This work presented a multi-objective approach for the unrelated parallel machine scheduling problem with sequence-dependent setup times, in the context of green scheduling, in which the main goals were to minimize makespan and *TEC*, formally defined as $R_M|S_{ijk}|(C_{\max}, TEC)$.

Motivated by the lack of similar approaches in the literature, we highlighted the possibility of providing good-quality schedules with low consumption of energy. For this purpose, a bi-objective variant of the problem was designed with a mathematical model. This model was inspired by recent

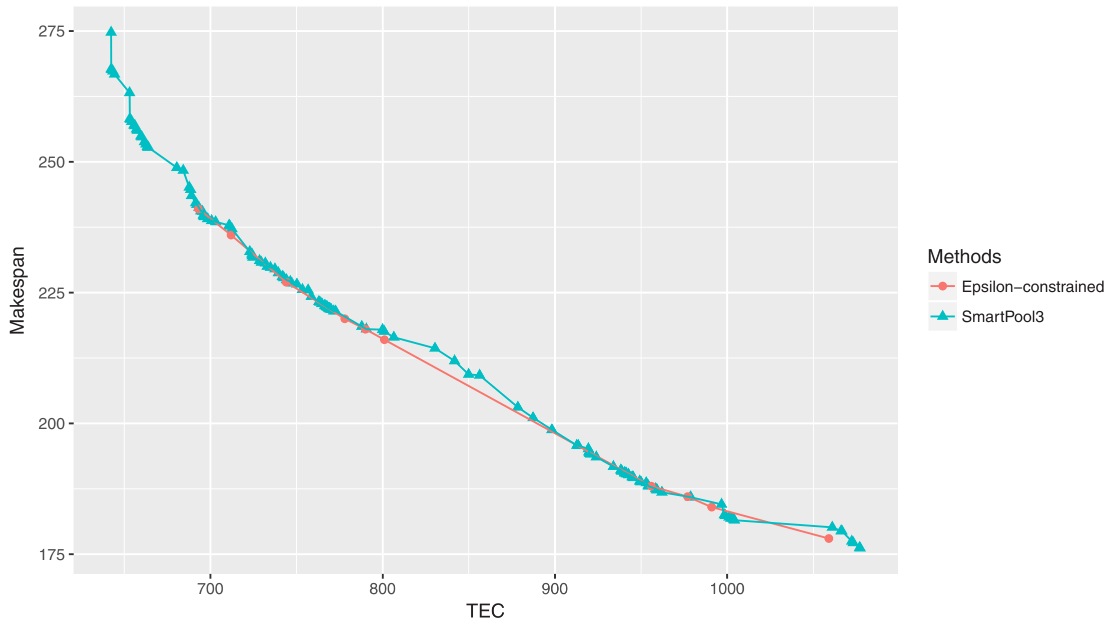

Fig. 8. Obtained Pareto fronts for the ϵ-constrained method and $SmartPool_3$ for an instance with 15 jobs and 2 machines.

works from the literature, which are moving towards the consideration of energy consumption in scheduling problems. The proposed mathematical model employs additional features that made the analysis of machines energy consumption possible, such as variable speed options for processing a job and corresponding machine power inputs.

For conducting our experiments, a new set of instances was created. The use of the multi-objective smart pool search matheuristic (or smart pool) was considered in order to find sets of non-dominated solutions in fixed and restricted computational budget, and compared with the classical ϵ-constrained method. The trade-off between both objectives, makespan and TEC, was verified and analysed. It was noted that the average time, per non-dominated solution found, taken by the smart pool was lower than the ϵ-constrained method, mainly for the larger problem instances (with 15 jobs). Thus, in real-life applications, this method would give more possibilities for decision makers in industry. The HV and coverage indicators also suggests a better performance of the smart pool. But it was observed that both methods achieved low coverage relative to each other.

In general, the contributions of this work can be applied to others scheduling problems. The mathematical model proposed and the smart pool search matheuristic can be easily adapted to various other machine scheduling problems. In this work, it is shown that the objectives makespan and TEC are conflicting; and that the TEC is very important because it represents a big cost to the industry. The efficiency of the smart pool search also is revealed, showing that this matheuristic can be applied to countless other problems in a simple way.

As future work, we intend to expand the proposed set of instances, as well as develop metaheuristic algorithms for solving the $R_M|S_{ijk}|(C_{\max}, TEC)$. Given the complexity of the problem, its use might be essential for solving large instances.

Acknowledgements

The authors would like to thank the Brazilian funding agencies CAPES, CNPq (grants 306850/2016-8, 307915/2016-6) and FAPEMIG (grant PPM CEX 676/17). Vitor N. Coelho would like to thank FAPERJ (grant E-26/202.868/2016).

References

Agnetis, A., Detti, P., Meloni, C., Pacciarelli, D., 2001. Set-up coordination between two stages of a supply chain. *Annals of Operations Research* 107, 15–32.

Ahilan, C., Kumanan, S., Sivakumaran, N., Dhas, J.E.R., 2013. Modeling and prediction of machining quality in CNC turning process using intelligent hybrid decision making tools. *Applied Soft Computing* 13, 3, 1543–1551.

Al-Salem, A., 2004. Scheduling to minimize makespan on unrelated parallel machines with sequence dependent setup times. *Engineering Journal of the University of Qatar* 17, 1, 177–187.

Arnaout, J.P., Musa, R., Rabadi, G., 2014. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines – part II: enhancements and experimentations. *Journal of Intelligent Manufacturing* 25, 43–53.

Arnaout, J. P., Rabadi, G., Musa, R., 2010. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. *Journal of Intelligent Manufacturing* 21, 6, 693–701.

Avalos-Rosales, O., Alvarez, A., Angel-Bello, F., 2013. A reformulation for the problem of scheduling unrelated parallel machines with sequence and machine dependent setup times. Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling, Rome, Italy, pp. 278–282.

Avalos-Rosales, O., Angel-Bello, F., Alvarez, A., 2015. Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times. *International Journal of Advanced Manufacturing Technology* 76, 9, 1705–1718.

Bampis, E., Letsios, D., Lucarelli, G., 2015. Green scheduling, flows and matchings. *Theoretical Computer Science* 579, 126–136.

Cheng, J., Chu, F., Liu, M., Wu, P., Xia, W., 2017. Bi-criteria single-machine batch scheduling with machine on/off switching under time-of-use tariffs. *Computers & Industrial Engineering* 112, 721–734.

Coelho, B.N., Coelho, V.N., Coelho, I.M., Ochi, L.S., Roozbeh Haghnazar, K., Lima, M.S., da Costa, A.R., 2017a. A multi-objective green UAV routing problem. *Computers & Operations Research* 88, Supplement C, 306–315.

Coelho, I.M., Munhoz, P.L.A., Haddad, M.N., Coelho, V.N., Silva, M.M., Souza, M.J.F., Ochi, L.S., 2011. A computational framework for combinatorial optimization problems. VII ALIO/EURO Workshop on Applied Combinatorial Optimization, Porto, pp. 51–54.

Coelho, V.N., Coelho, I.M., Coelho, B.N., Cohen, M.W., Reis, A.J.R., Silva, S.M., Souza, M.J.F., Fleming, P.J., Guimarães, F.G., 2016a. Multi-objective energy storage power dispatching using plug-in vehicles in a smart-microgrid. *Renewable Energy* 89, 730–742.

Coelho, V.N., Coelho, I.M., Coelho, B.N., de Oliveira, G.C., Barbosa, A.C., Pereira, L., de Freitas, A., Santos, H.G., Ochi, L.S., Guimarães, F.G., 2016b. A communitarian microgrid storage planning system inside the scope of a smart city. *Applied Energy* 201, 371–381.

Coelho, V.N., Cohen, M.W., Coelho, I.M., Liu, N., Guimarães, F.G., 2017b. Multi-agent systems applied for energy systems integration: state-of-the-art applications and trends in microgrids. *Applied Energy* 187, 820–832.

Cota, L.P., Guimarães, F.G., Oliveira, F.B., Souza, M.J.F., 2017. An adaptive large neighborhood search with learning automata for the unrelated parallel machine scheduling problem. Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC 2017), Donostia, San Sebastián, pp. 185–192.

Cota, L.P., Haddad, M.N., Souza, M.J.F., Coelho, V.N., 2014a. AIRP: a heuristic algorithm for solving the unrelated parallel machine scheduling problem. Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC 2014), Beijing, pp. 1855–1862.

Cota, L.P., Haddad, M.N., Souza, M.J.F., Martins, A.X., 2014b. A heuristic algorithm for solving the unrelated parallel machine scheduling problem with sequence dependent setup time (in Portuguese). Proceedings of the 35th Brazilian Congress of Applied and Computational Mathematics, Natal, Brazil. Available at http://proceedings.sbmac.org.br/sbmac/article/view/724/730 (accessed January 2017).

Deb, K., Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* 18, 4, 577–601.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 182–197.

Ding, J., Song, S., Wu, C., 2016. Carbon-efficient scheduling of flow shops by multi-objective optimization. *European Journal of Operational Research* 248, 758–771.

Franca, P.M., Gendreau, M., Laporte, G., Muller, F.M., 1996. A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times. *International Journal of Production Economics* 43, 79–89.

Garey, M., Johnson, D., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., San Francisco, CA.

Glass, A.C., Potts, C.N., Shade, P., 1994. Unrelated parallel machine scheduling using local search. *Mathematical and Computer Modeling* 20, 41–52.

Graham, R., Lawler, E., Lenstra, J., Kan, A., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* 5, 2, 287–326.

Haddad, M.N., Cota, L.P., Souza, M.J.F., Maculan, N., 2014. AIV: a heuristic algorithm based on iterated local search and variable neighborhood descent for solving the unrelated parallel machine scheduling problem with setup times. Proceedings of the 16th International Conference on Enterprise Information Systems (ICEIS 2014), Lisbon, Portugal, pp. 376–383. https://doi.org/10.5220/0004884603760383 (accessed January 2017).

Haddad, M.N., Cota, L.P., Souza, M.J.F., Maculan, N., 2015. Solving the unrelated parallel machine scheduling problem with setup times by efficient algorithms based on iterated local search. *Lecture Notes in Enterprise Information Systems* 227, 131–148.

Hansen, P., Mladenovic, N., Pérez, J.A.M., 2008. Variable neighborhood search: methods and applications. *4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies* 6, 319–360.

Karp, R.M., 1972. Reducibility among combinatorial problems. *Complexity of Computer Computations* 40, 4, 85–103.

Kim, D.W., Kim, K.H., Jang, W., Chen, F.F., 2002. Unrelated parallel machine scheduling with setup times using simulated annealing. *Robotics and Computer-Integrated Manufacturing* 18, 223–231.

Kim, D.W., Na, D.G., Chen, F.F., 2003. Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. *Robotics and Computer-Integrated Manufacturing* 19, 173–181.

Kurz, M.E., Askin, R.G., 2001. Heuristic scheduling of parallel machines with sequence-dependent set-up times. *International Journal of Production Research* 39, 3747–3769.

Lin, S.W., Ying, K.C., 2014. ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times. *Computers & Operations Research* 51, 172–181.

Liu, Y., Dong, H., Lohse, N., Petrovic, S., 2016. A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance. *International Journal of Production Economics* 179, 259–272.

Logendran, R., McDonell, B., Smucker, B., 2007. Scheduling unrelated parallel machines with sequence-dependent setups. *Computers & Operations Research* 34, 11, 3420–3438.

Lourenço, H.R., Martin, O., Stützle, T., 2003. Iterated local search. In Glover, F., Kochenberger, G. (eds), *Handbook of Metaheuristics, International Series in Operations Research & Management Science*, Vol. 57. Kluwer Academic, Norwell, MA, pp. 321–353.

Lust, T., Teghem, J., 2010. Two-phase pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics* 16, 475–510.

Mansouri, S.A., 2005. Coordination of set-ups between two stages of a supply chain using multi-objective genetic algorithms. *International Journal of Production Research* 43, 3163–3180.

Mansouri, S.A., Aktas, E., 2016. Minimizing energy consumption and makespan in a two-machine flow shop scheduling problem. *Journal of the Operational Research Society* 67, 11, 1382–1394.

Mansouri, S.A., Aktas, E., Besikci, U., 2016. Green scheduling of a two-machine flow shop: trade-off between makespan and energy consumption. *European Journal of Operation Research* 248, 772–788.

Meneghini, I.R., Guimarães, F.G., 2017. Evolutionary method for weight vector generation in multi-objective evolutionary algorithms based on decomposition and aggregation. 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, Spain, pp. 1900–1907.

Pereira Lopes, M.J., de Carvalho, J.M., 2007. A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times. *European Journal of Operational Research* 176, 1508–1527.

Pinedo, M.L., 2008. *Scheduling: Theory, Algorithms, and Systems* (3rd edn). Springer, New York, NY.

Rabadi, G., Moraga, R.J., Al-Salem, A., 2006. Heuristics for the unrelated parallel machine scheduling problem with setup times. *Journal of Intelligent Manufacturing* 17, 1, 85–97.

Radhakrishnan, S., Ventura, J.A., 2000. Simulated annealing for parallel machine scheduling with earliness/tardiness penalties and sequence-dependent set-up times. *International Journal of Production Research* 38, 2233–2252.

Randhawa, S.U., Kuo, C.H., 1997. Evaluating scheduling heuristics for non-identical parallel processors. *International Journal of Production Research* 35, 969–981.

Santos, H.G., Toffolo, T.A., Silva, C.L., Vanden Berghe, G., 2019. Analysis of stochastic local search methods for the unrelated parallel machine scheduling problem. *International Transactions in Operational Research* 26, 707–724.

Sauer, I.L., Tatizawa, H., Salotti, F.A., Mercedes, S.S., 2015. A comparative assessment of Brazilian electric motors performance with minimum efficiency standards. *Renewable and Sustainable Energy Reviews* 41, Suppl. C, 308–318.

Scheffé, H., 1958. Experiments with mixtures. *Journal of the Royal Statistical Society* 20, 2, 344–360.

Sen, P.C., 2013. *Principles of Electric Machines and Power Electronics*. Wiley, Hoboken, NJ.

SOA, 2011. Sistemas de Optimizacion Aplicada. A web site that includes benchmark problem data sets and solutions for scheduling problems. Available at http://soa.iti.es/problem-instances (accessed January 2017).

Srivastava, B., 1997. An effective heuristic for minimizing makespan on unrelated parallel machines. *Journal of the Operational Research Society* 49, 886–894.

Toro, E.M., Franco, J.F., Echeverri, M.G., Guimarães, F.G., 2017. A multi-objective model for the green capacitated location-routing problem considering environmental impact. *Computers & Industrial Engineering* 110, 114–125.

Tran, T.T., Araujo, A., Beck, J.C., 2016. Decomposition methods for the parallel machine scheduling problem with setups. *INFORMS Journal on Computing* 28, 1, 83–95.

Vallada, E., Ruiz, R., 2011. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research* 211, 3, 612–622.

Wang, S., Liu, M., Chu, F., Chu, C., 2016. Bi-objective optimization of a single machine batch scheduling problem with energy cost consideration. *Journal of Cleaner Production* 137, 1205–1215.

Weng, M.X., Lu, J., Ren, H., 2001. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *International Journal of Production Economics* 70, 215–226.

Wildi, T., 2013. *Electrical Machines, Drives and Power Systems*. Pearson, New York.

Ying, K.C., Lee, Z.J., Lin, S.W., 2012. Makespan minimisation for scheduling unrelated parallel machines with setup times. *Journal of Intelligent Manufacturing* 23, 5, 1795–1803.

Yu, J.J., 2010. Green scheduling and its solution. In Zhang, L., Zhang, C., Shi, T. (eds) *Advanced Materials Research*, Vol. 139–142. Trans Tech Publications, Zurich, pp. 1415–1418.

Zhang, H., Zhao, F., Fang, K., Sutherland, J.W., 2014. Energy-conscious flow shop scheduling under time-of-use electricity tariffs. *CIRP Annals* 63, 37–40.

Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q., 2011. Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm and Evolutionary Computation* 1, 1, 32–49.

Zhu, X., Wilhelm, W.E., 2006. Scheduling and lot sizing with sequence-dependent setup: a literature review. *IIE Transactions* 38, 987–1007.

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 257–271.