



# Bi-criteria parallel batch machine scheduling to minimize total weighted tardiness and electricity cost

Jens Rocholl<sup>1</sup> · Lars Mönch<sup>1</sup> · John Fowler<sup>2</sup>

Published online: 6 April 2020  
© The Author(s) 2020

## Abstract

A bi-criteria scheduling problem for parallel identical batch processing machines in semiconductor wafer fabrication facilities is studied. Only jobs belonging to the same family can be batched together. The performance measures are the total weighted tardiness and the electricity cost where a time-of-use (TOU) tariff is assumed. Unequal ready times of the jobs and non-identical job sizes are considered. A mixed integer linear program (MILP) is formulated. We analyze the special case where all jobs have the same size, all due dates are zero, and the jobs are available at time zero. Properties of Pareto-optimal schedules for this special case are stated. They lead to a more tractable MILP. We design three heuristics based on grouping genetic algorithms that are embedded into a non-dominated sorting genetic algorithm II framework. Three solution representations are studied that allow for choosing start times of the batches to take into account the energy consumption. We discuss a heuristic that improves a given near-to-optimal Pareto front. Computational experiments are conducted based on randomly generated problem instances. The  $\epsilon$ -constraint method is used for both MILP formulations to determine the true Pareto front. For large-sized problem instances, we apply the genetic algorithms (GAs). Some of the GAs provide high-quality solutions.

**Keywords** Scheduling · Batch processing · Semiconductor manufacturing · Energy consumption · Total weighted tardiness · Grouping genetic algorithm

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s11573-020-00970-6>) contains supplementary material, which is available to authorized users.

---

✉ Lars Mönch  
Lars.Moench@fernuni-hagen.de

<sup>1</sup> Department of Mathematics and Computer Science, University of Hagen, 58097 Hagen, Germany

<sup>2</sup> Department of Supply Chain Management, Arizona State University, Tempe, AZ 85287-5906, USA

**JEL Classification** C44 · C61 · L63 · M11

## 1 Introduction

Semiconductor manufacturing deals with producing integrated circuits (ICs) on wafers, thin discs made of silicon or gallium arsenide. The manufacturing process takes place in semiconductor wafer fabrication facilities (wafer fabs) by a process where ICs are built layer by layer on top of a raw wafer. Up to several thousands of ICs can be manufactured on a single wafer. The moving entities in a wafer fab are called lots, a set of wafers with ICs that belong to the same product. In this paper, we call lots jobs to align with the deterministic scheduling literature. Wafer fabs can be modeled as a job shop with a couple of unusual features such as a large number of machine groups with machines that offer the same functionality, reentrant process flows, (i.e., some of the machine groups are visited by the same job many times), and a mix of single wafer, lot, and batch processing (Mönch et al. 2013). A p-batch is a group of jobs that are processed at the same time, i.e. in parallel, on the same machine (Potts and Kovalyov 2000). Up to one-third of all operations in a wafer fab are performed on batch processing machines. The processing times on these machines generally are very long compared to times on other machines. Up to 24 h are possible for the longest processes. Since batch processing machines process several jobs at the same time, they tend to off-load multiple lots on machines that are able to process only single wafers or jobs. This leads to long queues in front of these serial machines. Therefore, scheduling batch processing machines in an appropriate manner is crucial for the overall performance of a wafer fab (Mönch et al. 2011).

The semiconductor industry is extremely energy-intensive due to the required clean room condition and the extremely complicated machinery. It consumes more electricity than other industries such as steel or petrochemical (Yu et al. 2017). Diffusion furnaces, typical batch processing machines in a wafer fab, are the machines with the largest energy consumption in wafer fabs (cf. Scholl 2017; Singapore Government 2019) due to the fact that the diffusion process is a high temperature process that disperses material on the wafer surface.

Therefore, in this paper, we study a model problem for scheduling jobs on parallel batch processing machines where we take two important, but contradicting performance measures into account, namely total weighted tardiness (TWT) of the jobs and electricity cost (EC) of the batch processing machines. We assume a TOU tariff, i.e., the energy price depends on the time when the energy is consumed. Although the effects of TOU tariffs are discussed controversially in the literature, there is some evidence that TOU tariffs can positively affect sustainability (Che et al. 2017). For instance, Finn et al. (2011) and Pina et al. (2012) demonstrate via use cases in Ireland and the Azores islands that demand response programs, among them TOU tariffs, lead to a higher penetration of renewable electricity. Stoll et al. (2014) demonstrate that the environmental impact of shifting consumption from on-peak to off-peak hours can be both positive and negative. A positive effect of the shifting is observed for Ontario and UK, but not for

Sweden. The impact depends on how dirty the generation of electricity during peak hours is. This observation is also confirmed by Zhang et al. (2014) who discuss a situation where electricity providers use gas-fired generation plants to support the coal-fired generation plants during peak hours. Albadi and El-Saadany (2008) argue that demand response programs allow for a better utilization of the current infrastructure. Wang and Li (2013) indicate that building costly backup infrastructure for peak hours can be avoided by demand response programs. Although a TOU tariff setting is still not a reality in many wafer fabs, the expected penetration of renewable energy sources in the high-tech industry (Ziarnetzky et al. 2017; Taiwan Semiconductor Manufacturing Company Limited (TSMC) Annual Report 2019) makes such a setting likely in the future. In addition to the EC performance measure, the TWT measure is also extremely important since it is related to on-time delivery performance which is important in the fierce competition of the semiconductor industry (Chien et al. 2011).

In the present paper, we propose heuristic approaches for the bi-criteria scheduling problem that are based on grouping genetic algorithms (GGAs) that are embedded into a non-dominated sorting genetic algorithm (NSGA) II-type framework. We also show that the  $\epsilon$ -constraint method can be used to compute the entire set of Pareto-optimal schedules for medium-sized problem instances of a special case of the general scheduling problem. To the best of our knowledge, the batch scheduling problem addressed in the present paper is not studied so far in the literature. This paper generalizes the problem studied in Rocholl et al. (2018) by changing from the total weighted completion time (TWC) measure to the TWT measure.

The paper is organized as follows. The bi-criteria scheduling problem is described in Sect. 2 and a mixed integer linear program (MILP) formulation is presented. Moreover, structural properties of Pareto-optimal solutions for a special case are stated. Related work is discussed in Sect. 3. Different heuristics are designed in Sect. 4. The results of computational experiments are presented in Sect. 5. Finally, conclusions and future research directions are discussed in Sect. 6.

## 2 Problem formulation and related work

### 2.1 Problem setting

The bi-criteria batch scheduling problem is based on the following assumptions:

1. There are  $F$  incompatible job families. Only jobs of the same family can be batched together due to the different chemical nature of the processes.
2. All jobs that belong to family  $1 \leq f \leq F$  have the same processing time  $p_{s(j)} \equiv p_f$  where  $s(j) := f$  is a mapping that assigns the family to a given job.
3. There are  $n_f$  jobs in family  $f$ . In total,  $n = \sum_{f=1}^F n_f$  jobs must be scheduled. Jobs are labeled by  $j = 1, \dots, n$ .
4. Job  $j$  has a weight  $w_j$  to model the importance of the job.
5. The size of job  $j$ , measured in number of wafers, is  $s_j$ .

6. Each job  $j$  has a ready time  $r_j \geq 0$ .
7. Each job  $j$  has a due date  $d_j$ .
8. There are  $m$  identical parallel machines, labeled by  $k = 1, \dots, m$ . All the machines are available at time  $t = 0$ .
9. All the machines have the same maximum batch size  $B$ , measured in number of wafers.
10. Preemption of the batch machines is not allowed, i.e., once a batch is started, it cannot be interrupted.
11. We assume that a finite scheduling horizon is divided into periods of equal size. The periods are labeled by  $t = 1, \dots, T$ . The EC is modeled as a piecewise constant function over the scheduling horizon.

We show an example for three furnace machines and jobs, each with five wafers, belonging to three incompatible families in Fig. 1. The different families are indicated by different colors of the wafers and the maximum batch size is  $B = 10$  wafers, i.e. two jobs. Note that the batch on the second machine only has a single job of family A. Forming such batches is reasonable in certain situations due to the ready times of the jobs and tight due dates. Seven additional jobs wait for processing in front of the three furnaces. They belong to the families A, B, and C. The time axis indicates that period-dependent energy costs are assumed.

The TWT measure of a schedule  $S$  is defined as follows:

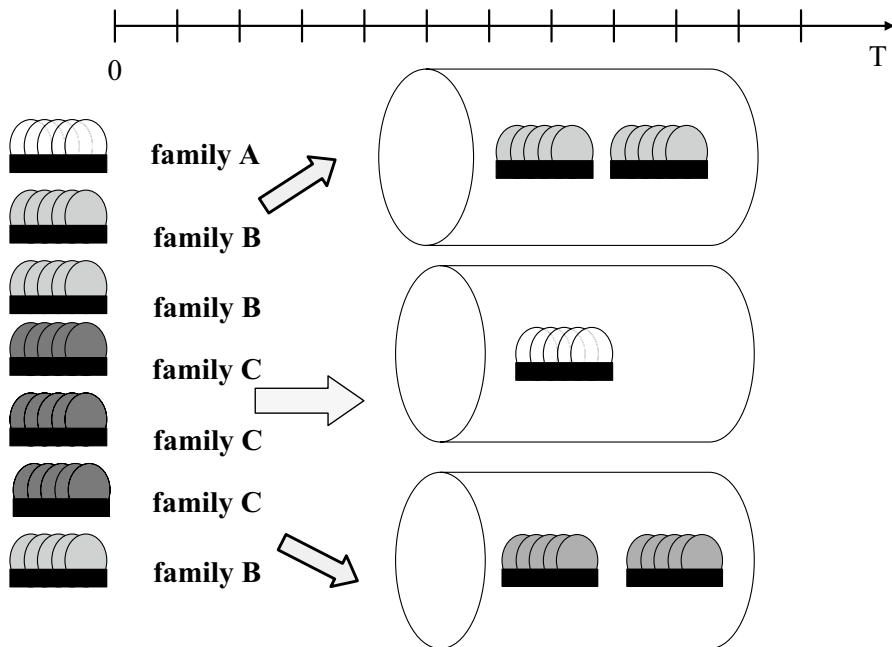


Fig. 1 Overall scheduling setting

$$TWT(S) = \sum_{j=1}^n w_j T_j, \quad (1)$$

where  $T_j := (C_j - d_j)^+$  is the tardiness,  $C_j$  the completion time of job  $j$  in  $S$ , and the abbreviation  $\chi^+ := \max(\chi, 0)$  is used for an arbitrary real number  $\chi$ . If we have  $d_j \equiv 0$ ,  $j = 1, \dots, n$  then we obtain  $TWT = \sum_{j=1}^n w_j C_j = TWC$ , i.e., we get the TWC measure. It is a surrogate measure for the weighted cycle time, an important key performance indicator in wafer fabs (Mönch et al. 2013). The second performance measure of interest is the EC. We assume that the EC values are constant in each period, i.e., we have a mapping  $e : [0, T] \rightarrow \mathbb{R}^+$  that is piecewise constant in each period, to model the TOU tariff. If  $e(t)$  is the corresponding EC value in period  $t$  then the EC value of schedule  $S$  can be expressed as follows:

$$EC(S) := \sum_{t=1}^T \sum_{k=1}^m e(t) z_t^k, \quad (2)$$

where  $z_t^k$  is 1 if a batch is processed in period  $t$  on machine  $k$  in  $S$  and zero otherwise. Note that the two objectives are in conflict. Minimizing TWT leads typically to non-delay schedules, while minimizing EC requires determining periods for processing within the scheduling horizon where the  $e(t)$  values are low. The EC measure is a surrogate measure for sustainability efforts since low  $e(t)$  values offered by TOU tariffs are often considered as a prerequisite to increase the penetration of renewable energy (Finn et al. 2011; Pina et al. 2012; IRENA 2019). Note that minimizing the EC measure in the case of a constant EC for the entire scheduling horizon is equivalent to minimizing the number of batches which clearly leads to a reduction of the energy consumption. But even in the case of a general TOU tariff a small EC value often leads to a small number of formed batches.

Using the three-field notation from deterministic scheduling theory (Graham et al. 1979), the scheduling problem at hand can be represented as follows:

$$P|p - \text{batch}, \text{incompatible}, r_j, s_j|ND(TWT, EC), \quad (3)$$

where  $P$  indicates identical parallel machines,  $p$ -batch, incompatible refers to batch processing machines with incompatible families,  $r_j$  to unequal ready times, and  $s_j$  to non-identical job sizes. The notation  $ND(TWT, EC)$  refers to the set of all Pareto-optimal solutions, i.e., a schedule  $S$  is called non-dominated when no other feasible schedule  $S'$  exists with  $TWT(S') \leq TWT(S)$  and  $EC(S') \leq EC(S)$ , and at least one of the two inequalities is strict. The entire set of all non-dominated solutions for a problem instance is called the Pareto frontier.

Note that scheduling problem (3) is NP-hard since the scheduling problem  $P|p - \text{batch}, \text{incompatible}|TWC$ , a special case of problem (3), is NP-hard due to Uzsoy (1995). Hence, we have to look for efficient heuristics if we want to tackle large-sized problem instances in a reasonable amount of computing time.

### 2.2 MILP formulation

A MILP formulation for scheduling problem (3) is presented in this section. A time-indexed formulation for the scheduling horizon  $t = 1, \dots, T$  is used. The following indices and sets are used in the model:

- $j = 1, \dots, n$ : job indices
- $i = 1, \dots, b$ : batch indices
- $k = 1, \dots, m$ : machine indices
- $t = 1, \dots, T$ : period indices.

The following parameters are included in the model:

- $w_j$ : weight of job  $j$
- $r_j$ : ready time of job  $j$
- $s_j$ : size of job  $j$  (in wafers)
- $B$ : maximum batch size  $B$  (in wafers)
- $s(j)$ : family of job  $j$
- $f(i)$ : family of batch  $i$
- $p_{s(j)}$ : processing time of job  $j$
- $P_{f(i)}$ : processing time of batch  $i$
- $e(t)$ : electricity cost in period  $t$ .

The following decision variables belong to the model:

$$x_{ij} = \begin{cases} 1, & \text{if job } j \text{ is assigned to batch } i \\ 0, & \text{otherwise} \end{cases}$$

$$y_{itk} = x_{ij} = \begin{cases} 1, & \text{if batch } i \text{ is started in period } t \text{ on machine } k \\ 0, & \text{otherwise} \end{cases}$$

- $C_j$ : completion time of job  $j$
- $T_j$ : tardiness of job  $j$
- $Cb_i$ : completion time of batch  $i$ .

The model can be formulated as follows:

$$\min \left( \sum_{j=1}^n w_j T_j, \sum_{i=1}^b \sum_{t=1}^T \sum_{k=1}^m \sum_{\tau=t}^{t+p_{f(i)}-1} e(\tau) y_{itk} \right) \tag{4}$$

subject to

$$\sum_{i=1}^b x_{ij} = 1 \quad j = 1, \dots, n \tag{5}$$

$$\sum_{j=1}^n s_j x_{ij} \leq B \quad i = 1, \dots, b \tag{6}$$

$$x_{ij}(f(i) - s(j)) = 0 \quad j = 1, \dots, n, i = 1, \dots, b \tag{7}$$

$$x_{ij} \leq \sum_{t=1}^T \sum_{k=1}^m y_{itk} \quad j = 1, \dots, n, i = 1, \dots, b \quad (8)$$

$$Cb_i \leq T \quad i = 1, \dots, b \quad (9)$$

$$\sum_{i=1}^b \sum_{\tau=(t-P_{f(i)})^++1}^t y_{i\tau k} \leq 1 \quad k = 1, \dots, m, t = 1, \dots, T \quad (10)$$

$$r_j x_{ij} + P_{f(i)} \leq Cb_i \quad j = 1, \dots, n, i = 1, \dots, b \quad (11)$$

$$Cb_i = \sum_{k=1}^m \sum_{t=1}^T t y_{itk} + P_{f(i)} \quad i = 1, \dots, b \quad (12)$$

$$Cb_i \leq C_j + T(1 - x_{ij}) \quad j = 1, \dots, n, i = 1, \dots, b \quad (13)$$

$$C_j \leq Cb_i + T(1 - x_{ij}) \quad j = 1, \dots, n, i = 1, \dots, b \quad (14)$$

$$C_j - d_j \leq T_j \quad j = 1, \dots, n \quad (15)$$

$$Cb_i, C_j, T_j \geq 0, x_{ij}, y_{itk} \in \{0, 1\} \quad j = 1, \dots, n, i = 1, \dots, b, t = 1, \dots, T, k = 1, \dots, m. \quad (16)$$

We are interested in determining all Pareto-optimal solutions with respect to the TWT and the EC measures. This is modeled by (4). The constraint set (5) ensures that each job is assigned to exactly one batch. The maximum batch size is modeled by constraint set (6). The constraints (7) model that only jobs belonging to the same family can be used to form a batch. The family of a batch is determined by its jobs. When at least one job is assigned to a batch then this batch has to start in some period on some machine. This is modeled by constraint set (8). Constraint set (9) makes sure that the completion time of a batch is not larger than the end of the scheduling horizon. The constraints (10) model the fact that each batch starts at most once before the end of the scheduling horizon and no overlapping occurs in the processing of batches on a machine. The constraints (11) enforce that the ready time of the jobs are respected, i.e., a batch can only start if all jobs that belong to the batch are ready. The completion of a batch is calculated by Eq. (12). The constraint sets (13) and (14) ensure that the completion time of a batch and the completion time of the jobs that belong to this batch are the same. The tardiness of a job is linearized by the constraints (15). The domains of the decision variables are respected by constraints (16).

Determining the Pareto front for instances of problem (3) can be based on the (equidistant)  $\varepsilon$ -constraint method (cf. Ehrgott 2010) if the quantities  $p_j$ ,  $w_j$ ,  $d_j$  and the  $e(t)$  values are integers. A single objective is optimized within each iteration while the remaining objectives are transformed into constraints. The application of the  $\varepsilon$ -constraint method to the MILP (4)–(16) is shown in the Appendix.

### 2.3 Analysis of a special case

For the sake of completeness, we recall important results for the special case  $r_j \equiv 0$ ,  $d_j \equiv 0$  from Rocholl et al. (2018). In addition, all jobs have the same size, i.e., without loss of generality we assume  $s_j \equiv 1$  and  $B \in \mathbb{N}$ . This leads to the scheduling problem

$$P|p - \text{batch, incompatible}|TWC. \quad (17)$$

Due to Uzsoy (1995), problem (17) is still NP-hard. The following property holds that generalizes a result from Uzsoy (1995) for single regular performance measures.

**Property 1** *There is a Pareto-optimal schedule for each point of the Pareto front of an instance of problem (17) where all batches except maybe the last scheduled batch of a family contain  $B$  jobs.*

We refer to (Rocholl et al. 2018) for a proof of this property. Note that Property 1 is not necessarily true for problem (3). It follows from Property 1 that a Pareto-optimal schedule exists for each point of the Pareto front where the number of batches in family  $f$  is  $\lceil n_f/B \rceil$ . A second property holds where the structure of batches in Pareto-optimal schedules for instances of problem (17) is considered.

**Property 2** *For each point of the Pareto front of instances of problem (17) there is a Pareto-optimal schedule where for each pair of batches  $\rho_1$  and  $\rho_2$  of the same family with completion times  $Cb_1 \leq Cb_2$  the weight of each job belonging to  $\rho_1$  is not smaller than any weight of a job from  $\rho_2$ .*

A proof of this property can be again found in Rocholl et al. (2018). The jobs in each family are sorted with respect to non-increasing values of the job weights. These job sequences are then used to form batches for each family according to Property 1. The total number of batches is known and denoted by  $b$ . The following compact MILP formulation is possible for problem (17). Only additional notation compared to the MILP formulation (4)–(16) is introduced:

$wb_i$  sum of the weights of the jobs that form batch  $i$ .

The MILP model can be formulated as follows:



$$\min \left( \sum_{i=1}^b \sum_{t=1}^T \sum_{k=1}^m w b_i (t + P_{f(i)}) y_{itk}, \sum_{i=1}^b \sum_{t=1}^T \sum_{k=1}^m \sum_{\tau=t}^{t+P_{f(i)}-1} e(\tau) y_{itk} \right) \quad (18)$$

subject to

$$\sum_{t=1}^{T+1-P_{f(i)}} \sum_{k=1}^m y_{itk} = 1 \quad i = 1, \dots, b \quad (19)$$

$$\sum_{i=1}^b \sum_{\tau=(t-P_{f(i)})^++1}^t y_{i\tau k} \leq 1 \quad k = 1, \dots, m, t = 1, \dots, T \quad (20)$$

$$y_{itk} \in \{0, 1\} \quad i = 1, \dots, b, t = 1, \dots, T, k = 1, \dots, m. \quad (21)$$

Constraint set (19) makes sure that each batch is used only once, whereas the constraint set (20) ensures that at most one batch is processed in each period on a single machine. The formulation (18)–(21) is easier to solve than (4)–(16) since the batches are already formed in the former formulation. This leads to less binary decision variables. The (equidistant)  $\varepsilon$ -constraint method can be applied to the MILP (18)–(21) in the same way as shown in the Appendix for the MILP (4)–(16). The  $\varepsilon$ -constraint method is able to determine all Pareto-optimal solutions for medium-sized problem instances with up to 90 jobs in a short amount of computing time. Therefore, we can assess the quality of the proposed heuristics based on these instances.

### 3 Discussion of related work

Next, we discuss the literature with respect to multi-criteria batch scheduling in semiconductor manufacturing and energy-aware scheduling for parallel batching and serial machines, especially for TOU tariffs. There are many papers that deal with single-criterion parallel batch scheduling in semiconductor manufacturing. For related surveys we refer to Mathirajan and Sivakumar (2006) and Mönch et al. (2011).

It turns out that multi-criteria scheduling approaches with parallel batching machines for semiconductor manufacturing are rarely discussed in the literature. We are only aware of Reichelt and Mönch (2006), Mason et al. (2007) and Li et al. (2009). In the first paper, the scheduling problem  $P|r_j, p - \text{batch, incompatible}|ND(C_{max}, TWT)$  is solved using an NSGA-II approach that is hybridized by a list scheduling approach, while the second paper proposes a NSGA-II scheme for a similar parallel batch machine scheduling problem for TWT, cycle time variation, and time constraint violation. Here,  $C_{max}$  is the makespan.

The third paper uses ant colony optimization to deal with a similar problem which requires additional consideration of qualification runs and sequence-dependent setup times. Problem (3) is different from these settings since it might be reasonable to consider schedules that are not necessarily non-delayed due to the TOU tariffs.

Energy-aware scheduling has recently attracted a number of researchers from academia and industry. Recent surveys of this topic are (Giret et al. 2015, Merkert et al. 2015, Gahm et al. 2016, Akbar and Irohara 2018, and Gao et al. 2019). They conclude that scheduling with energy-related objectives and constraints is an important research direction.

We start by discussing related work for single and parallel batch processing machines with EC-related objectives. The problem  $1|p - batch, p_j \equiv p|ND(C_{max}, EC)$  is discussed by Cheng et al. (2014). All jobs have the same processing time, i.e., there is only a single job family. A TOU tariff is assumed. The  $\epsilon$ -constraint method is used. Improved MILP formulations for this problem are discussed by Cheng et al. (2016b). Cheng et al. (2017) propose a heuristic variant of the  $\epsilon$ -constraint method for the same scheduling problem, but on/off switching of the machines is allowed. A similar single-machine batch scheduling problem is considered by Wang et al. (2016). The energy consumption depends on the selected temperature of the machine. The  $\epsilon$ -constraint method is applied for small-sized instances whereas constructive heuristics are designed to solve large-sized instances in a reasonable amount of computing time. Cheng (2017) and Cheng et al. (2016a) study the  $1|p - batch|ND(C_{max}, EC)$  problem where  $p$ -batch means that the processing time of a batch is determined by the longest processing time of the jobs that belong to the batch. The  $\epsilon$ -constraint method is used to solve this problem. The resulting single-objective problems are solved by considering a series of successive knapsack problems or multiple knapsack problems. Fuzzy logic is used to recommend a preferred solution to decision-makers. The problem  $1|p - batch|ND(L_{max}, NB)$  is studied by Cabo et al. (2018). Here,  $L_{max}$  and  $NB$  are the maximum lateness and the number of formed batches, respectively. Note that a small number of batches lead to a small energy consumption too. The  $\epsilon$ -constraint method is used to solve small-sized instances, while a biased random-key genetic algorithm is used to tackle large-sized instances. Since problem (3) contains parallel machines, the scheduling techniques from these single-machine papers cannot directly be applied to the present problem.

The problem  $P|r_j, p - batch|ND(TWT, CO_2)$  is studied by Liu (2014). The  $CO_2$  performance measure is considered. Its value is obtained from multiplying the total EC by a constant factor. A NSGA-II approach is used to solve large-sized instances. The problem  $P|r_j, s_j, p - batch|ND(C_{max}, EC)$  is studied by Jia et al. (2017) using a bi-criteria ant colony optimization (ACO) approach. Jia et al. (2019) consider the same problem as in (Jia et al. 2017), but the energy consumption of the machines in parallel can be different. Moreover, the ACO approach and the local search (LS) scheme are improved compared to the previous work. Cheng (2017) consider the scheduling problem  $Q|s_j, p - batch, |ND(EC, NEM)$ , where  $Q$  refers to uniform parallel machines and  $NEM$  is the number of enabled machines. A two-stage heuristic scheme is proposed. Batches are formed on the first stage, whereas the batches are

assigned and sequenced on the parallel machines in the second stage. None of these papers except the paper by Liu (2014) considers the TWT performance measure. The scheduling problem discussed by Liu (2014), however, does not have incompatible families and a TOU tariff. Hence, the proposed techniques in these papers cannot be directly applied to problem (3). Problem (17), a special case of problem (3), is studied by Rocholl et al. (2018). A MILP formulation is presented. The  $\varepsilon$ -constraint method is used in preliminary computational experiments for medium-sized problem instances. No heuristic approaches are considered. In the present paper, we will use the  $\varepsilon$ -constraint method for the problem studied by Rocholl et al. (2018) to assess the proposed GA-type approaches. The main contributions of this paper are twofold:

1. We consider a bi-criteria energy-aware scheduling problem for parallel batching machines and incompatible job families with a due date-related objective function, namely the TWT performance measure. It is well-known that this class of scheduling problems even for a single machine is strongly NP-hard (Brucker et al. 1998). Although it has applications in wafer fabs, to the best of our knowledge, this problem is not considered in the literature so far.
2. From a methodological point of view, we design three representations for GA-type algorithms that can be used in more general situations where non-delay schedules are not of interest. It is well-known that determining starting points given the sequence of activities is a nontrivial task (cf. Vidal et al. 2015). However, we are only aware of the papers by Goncalves et al. (2008) and Moon et al. (2013) where delay times are systematically incorporated in GAs for scheduling problems.

## 4 Heuristic solution approaches

### 4.1 Basic design ideas

Various metaheuristics are designed for multi-objective optimization problems. Among them, the NSGA-II approach proposed by Deb et al. (2002) is widely used (Landa Silva and Burke 2002). That is why we use a NSGA-II scheme in this research. More details of the major NSGA-II principles are discussed in Sect. 4.2. Scheduling problem (3) requires the formation of groups of jobs, i.e., a single batch is represented by a group. Grouping problems can be solved efficiently by GGAs (cf. Falkenauer 1998). Successful applications of GGA to bin packing problems (Falkenauer 1996), single-machine batch scheduling (Sobeyko and Mönch 2011), and single-machine multiple orders per job (MOJ) scheduling problems (Sobeyko and Mönch 2015) are known (cf. Sect. 4.2 for a more detailed discussion of GGAs). Therefore, we use a GGA-based representation within the NSGA-II scheme. We have to design more problem-specific representations to refine the GGA standard representation. Therefore, three representations will be proposed in Sect. 4.3. LS is crucial to improve the solutions determined by NSGA-II-type algorithms (Deb

and Goel 2001). Moreover, it is an important ingredient in metaheuristics for batch scheduling problems. This includes memetic algorithms, GAs where each chromosome is improved by LS (Chiang et al. 2010). LS procedures proposed for batch scheduling problems by Sobeyko and Mönch (2011) and Jia et al. (2017) are adapted to problem (3) in Sect. 4.4. Due to considering non-delayed schedules, it is likely that different non-dominated solutions can be found in the proximity of an already known Pareto-optimal solution. In the case of large Pareto frontiers, due to the resulting huge search space it may be inefficient or even impossible to fully explore the neighborhoods of all already determined solutions using a metaheuristic approach like NSGA-II. Therefore, an improvement heuristic is proposed in Sect. 4.5 to enlarge the set of non-dominated solutions with adjacent solutions within the same front.

## 4.2 NSGA-II and GGA principles

### 4.2.1 NSGA-II

The NSGA-II approach is designed to ensure diversity by exploiting information of solutions from the entire population. The set of solutions that corresponds to a population is sorted into distinct fronts of different domination levels in each iteration. The first front contains all solutions which are not dominated by any other solution. The second front contains those which are only dominated by solutions from the first front, and so on. The fitness value of an individual is determined by the front its solution belongs to. For solutions belonging to the same front a crowded-comparison operator is used that assigns a higher fitness value to solutions in less crowded regions of the solution space. Binary tournament selection is used, i.e., two individuals of a population are randomly chosen and the one with higher fitness is selected for crossover (Goldberg 1989; Michalewicz 1996). Offspring are generated by recombination until the population size is doubled. An elitist strategy (cf. Michalewicz 1996) is applied. Individuals are inserted into the new population by non-increasing fitness values, i.e., solutions are accepted starting from the first front until the original population size is reached. Solutions from the least crowded regions of the solution space are preferred from the last front to be accepted. In the present paper, we apply a NSGA-II procedure to deal with the two criteria of problem (3). We will describe the encoding and decoding schemes in the next subsections.

### 4.2.2 GGA

GGAs are introduced by Falkenauer (1996, 1998) since conventional encoding schemes for grouping problems often do not work well. This is caused by the fact that a direct encoding of grouping decisions often leads to highly redundant representations. Furthermore, recombination operations usually disrupt the formed group and require sophisticated repair actions. A GGA is a GA with a representation in which the genetic operations are not applied to the jobs but rather to the formed groups. One gene encodes one or more jobs of the same family and represents a group, i.e. a batch,

formed by these jobs. The number of formed groups may vary for different solutions. As the batch formation is a major decision in solving problem (3), we choose a grouping representation as the base for the encoding and decoding schemes described in the next subsection. Each genome consists of a given set of groups (batches). An example for the grouping representation is provided in Fig. 2. Four batches are formed that belong to three different families.

Recombination of two individuals can be carried out by a two-point crossover (cf. Goldberg 1989; Michalewicz 1996). First, the selected genes from both parents are copied to the offspring, only a single chromosome due to the NSGA-II principles (see Sect. 4.2.1). Possible duplicate jobs are then deleted from the genes of the second parent. To ensure feasibility, jobs missing in the offspring must be reinserted. It is well known (cf. Brown and Sumichrast 2003) that the chosen reinsertion strategy highly influences the performance of a GGA for a given problem. The reinsertion operation is designed in such a way that a missing job  $j$  is inserted in the first batch  $i$  of family  $s(j)$  for which the  $r_j$  value is less or equal to the ready time of at least one job that already belongs to  $i$  and  $\sum_{k \in i} s_k + s_j \leq B$ . In contrast to Falkenauer (1998), we do not use any mutation for the GGA-type part of the encoding scheme.

### 4.3 Additional encoding and decoding schemes

#### 4.3.1 First representation

The LIST representation is based on the idea to schedule each batch in such a way that its impact on only one of the two objectives is small. Therefore, we use a random key  $\theta \in [0, 1]$  in addition to the groups in each individual to encode which batches are scheduled with respect to TWT and which ones with respect to EC. If we have  $b$  groups in a chromosome (see Fig. 2) then the batches labeled by  $1, \dots, \lfloor \theta b \rfloor$  are scheduled in such a way that the corresponding partial schedule has a small TWT value, while the batches  $\lfloor \theta b \rfloor + 1, \dots, b$  are scheduled such that their EC value is small. During crossover, the value  $\theta$  is inherited from the first parent chromosome to the child chromosome. If mutation is applied to an individual, its  $\theta$  value is randomly reinitialized.

Next, we describe how we decode the batches in the two subsets. The decoder is given by two list scheduling algorithms that assign batches to machines and to time slots. The first algorithm, applied to the batches  $1, \dots, \lfloor \theta b \rfloor$  and abbreviated by LIST-ASAP, can be summarized as followed. Here, LIST refers to list scheduling and ASAP indicates that batches are started as soon as possible.

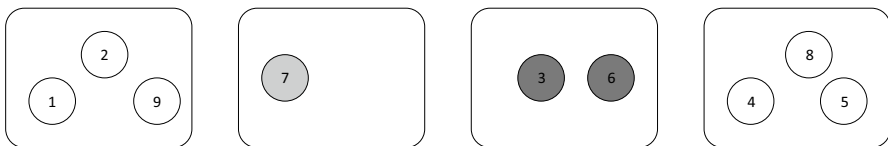


Fig. 2 Example for the grouping representation

### LIST-ASAP procedure

1. Whenever a machine becomes available, the first batch from the list is assigned to it and set to be started as soon as possible respecting the ready times of the jobs that belong to the batch.
2. Update the availability time of the machine.
3. Remove the batch from the list of unscheduled batches.
4. Go to Step 1 if the remaining list is non-empty.

The LIST-ASAP procedure produces non-delay schedules with rather small completion times, while the EC value is completely neglected.

The second list scheduling algorithm for the batches  $[\theta b] + 1, \dots, b$ , abbreviated by LIST-MEC, determines a partial schedule with a small EC value. In the abbreviation, MEC is used to indicate minimum EC values. The procedure can be summarized as follows:

### LIST-MEC procedure

1. Consider the first batch from the list.
2. Calculate the EC value for the batch for all feasible starting times of the batch by respecting machine availability, ready times of jobs that belong to the batch, and its processing time.
3. The batch is then scheduled on a machine at a starting time that leads to the smallest EC value. Ties are broken by considering the smallest starting time.
4. Update the machine availability of the chosen machine.
5. Remove the batch from the list of unscheduled batches.
6. Go to Step 1 when the remaining list is non-empty.

When electricity tariffs are used which are not increasing over the entire scheduling horizon, the EC measure is not regular, i.e., the objective value does not always increase with larger completion times. As a consequence, the search for Pareto-optimal solutions cannot be limited to the class of non-delay schedules. Although especially LIST-MEC may lead to schedules with idle times between batches, the completion times of batches will be limited to only a subset of all periods. The first batch to be assigned, for instance, will either be started immediately after its ready time or in the least expensive periods with respect to the EC value. Therefore, it is likely that certain parts of the search space are not explored by the LIST representation.

### 4.3.2 Second representation

This limitation of the LIST representation motivates a second encoding scheme which allows for inserting idle times between the processed batches. This representation is abbreviated by batch delayed (BD) to indicate that the processing of batches can be delayed. Given a fixed scheduling horizon, determining intervals of possible delays for the processing of a batch is non-trivial due to the strong interdependency

of the involved scheduling decisions. To allow a feasible schedule, the inserted idle times have to be smaller or equal to the total idle time of a machine which depends on the formed batches and their sequence on the machine. Moreover, the interval of possible delay for a single batch depends not only on the total idle time but also on the delays inserted before all other batches on the machine.

The assignment of batches to machines is encoded by a random value  $\mu_i \sim DU[1, m]$  for each batch  $i$ . Here,  $DU[a, b]$  is a discrete uniform distribution over the set of integers  $\{a, \dots, b\}$ . The presence and length of idle times before the processing of a batch  $i$  is encoded by a random key  $\theta_i \in [0, 1)$ . Additional random keys  $\tilde{\theta}_k \in [0, 1)$ ,  $k = 1, \dots, m$ , are used to represent the amount of idle time that is inserted after the last batch on machine  $k$ . Let  $B_k$  be the sequence of batches assigned to machine  $k$ , and  $C_{max}(k)$  the completion of the last batch on machine  $k$  if the first batch of  $B_k$  can start at  $t = 0$ . The amount of idle time inserted before batch  $i$  assigned to machine  $k$  is then calculated as follows:

$$\tilde{p}_i = (T - C_{max}(k))\theta_i / \left( \sum_{h \in B_k} \theta_h + \tilde{\theta}_k \right). \quad (22)$$

On the one hand, as the amount of idle time is relatively encoded, this information is context-sensitive to a large extent. Thus, we expect that crossover and LS operations will have a disruptive effect on the timing information. Therefore, this information cannot be passed along to later generations well. On the other hand, virtually all solutions from the solution space can be encoded and recombination always yields feasible schedules, provided that the encoded batch formation and assignment allow for one.

We expect that a random choice of the  $\theta_i$  values will only rarely lead to non-delay schedules. But since non-delay schedules are favorable with respect to small completion times, we generate a certain amount  $\tau$  of individuals in each generation with  $\theta_i = 0$ ,  $i \in \{1, \dots, b\}$  and  $\tilde{\theta}_k > 0$ ,  $k = 1, \dots, m$ . The crossover operator passes on the genes with the  $\mu_i$  and  $\theta_i$  values since each  $\mu_i$  and  $\theta_i$  value belongs to a group. The set of values  $\tilde{\theta}_k$ ,  $k = 1, \dots, m$  is inherited from the first parent chromosome. The mutation operator randomly reinitializes the  $\theta_i$  values for  $i = 1, \dots, b$ .

### 4.3.3 Third representation

A third representation, abbreviated by hybrid (HYB), is designed as a hybrid of LIST and BD. A chromosome of HYB contains all the features of the two encoding schemes. Moreover, a random key  $\gamma \in [0, 1]$  determines how the chromosome is decoded. The decoding scheme of LIST is applied if  $\gamma < 0.5$  holds, otherwise the BD decoding scheme is used. Only the components relevant for that particular encoding scheme are interpreted during decoding. Still, the other components are passed on to offspring because the scheme to be applied might be changed by recombination or mutation. Given the  $\gamma_1, \gamma_2$  values from two chromosomes chosen for recombination,  $\gamma := (\gamma_1 + \gamma_2)/2$  is chosen for the single child chromosome by crossover. The mutation operator sets the  $\gamma$  value as follows:

$$\gamma := \begin{cases} \gamma + 0.5, & \text{if } \gamma < 0.5 \\ \gamma - 0.5, & \text{otherwise.} \end{cases} \quad (23)$$

#### 4.4 Local search schemes

We propose a two-phase LS approach. In a first phase, the work load of the machines is balanced. The batch sequenced last on the machine with the largest completion time of a batch is reassigned to the machine with the lowest completion time if the objective values do not increase. This operation is repeated until no further improvement can be achieved. The described workload balancing procedure is also used to repair chromosomes with the BD representation if certain batch assignments lead to infeasible schedules.

We start by describing a LS scheme which is applied in the second phase. The LS scheme can be summarized as follows:

##### LS procedure

1. *Job insert* All jobs in a schedule are considered for being removed from one batch and inserted into another batch of the same family with enough available capacity and a starting time that is not smaller than the ready time of the job to be inserted. A job is reassigned to another batch if that leads to a reduced total completion time (TC) value. The TC measure is considered in this situation because even if reassigning a job does not lead to a reduced TWT value, it may allow that other delayed jobs are processed earlier. A single pass through all jobs in the order of non-decreasing completion times in the original schedule is performed.
2. *Job swap* The batch formation can be altered by exchanging two jobs of the same family between two different batches. All jobs are considered for a swap operation if the batches are not started before the ready times of the jobs. A swap is performed if the TWT value is reduced. All jobs are considered in a single pass through the list of jobs ordered with respect to non-decreasing completion times in the original schedule.
3. *Batch postpone* If a batch is completed before the smallest due date of the jobs that form the batch then the batch can be postponed without affecting the TWT value. Postponing a batch might even reduce the EC value if it is moved to a less expensive time slot. Each batch is considered in a single pass from the right to the left starting with the last batch on each machine. A batch is postponed as much as possible as long as both objective function values are not increased.
4. *Batch pull* It might be possible to shift batches to the left to reduce the TWT value without increasing the EC value. Therefore, a single pass from the left to the right through the list of batches processed on each machine is made. Each batch is shifted to the left as much as possible as long as the EC value is not increased.



A second LS procedure, applied within the NSGA-II scheme, is composed of batch postpone and pull operations. An integrated objective function is used for a given schedule  $S$  as follows:

$$F(S) := (w_1^S TWT(S) + w_2^S EC(S)) / (w_1^S + w_2^S), \quad (24)$$

where the weights are given by

$$w_1^S := \frac{TWT^{\max} - TWT(S)}{TWT^{\max} - TWT^{\min}}, \quad w_2^S := \frac{EC^{\max} - EC(S)}{EC^{\max} - EC^{\min}}. \quad (25)$$

Here,  $f^{\max}$  and  $f^{\min}$  are the maximum and minimum value of the objective function values  $f$  for the entire population. The batch postpone and pull moves are applied with respect to the integrated objective function  $F$  instead of using the  $TWT$  and the  $EC$  measures separately. We abbreviate this procedure by LS-NSGA-II.

#### 4.5 Improvement heuristic

The main principle of this algorithm is to iteratively compute modified schedules by postponing, i.e. right-shifting, batches. Batches are only postponed when EC reductions occur and only a minimum impact on the TWT value can be observed. This is important in order to obtain new schedules for which it is likely that they are Pareto-optimal. Because of these features we abbreviate the algorithm by MIN-POSP where MIN indicates the minimum impact on the TWT measure whereas POSP refers to a postponement of batches. The batch formation, batch assignment, and sequencing decisions remain unchanged.

Let  $k[n]$  be the batch in the  $n$ -th position on machine  $k$  and  $C_{k[n]}$  the completion time of this batch. In addition,  $\omega_k$  is the last scheduled batch on machine  $k$ . The mapping  $t(k, n)$  provides the number of periods the batch  $k[n]$  can be postponed without interfering with the batch  $k[n + 1]$  or the scheduling horizon if  $k[n] = \omega_k$ . We obtain:

$$t(k, n) := \begin{cases} C_{k[n+1]} - C_{k[n]} - p_{f(k[n+1])}, & \text{if } \omega_k \neq k[n] \\ T - C_{k[n]}, & \text{otherwise} \end{cases}. \quad (26)$$

The algorithm starts from a given schedule  $S$ . The notation  $\Delta f := f(S') - f(S)$  is introduced for a schedule  $S^*$  that is obtained from  $S$  by modifying it. A modified schedule  $S^*$  is returned if an EC value reduction is achieved by postponing at least one batch. The heuristic can be summarized as follows:

**MIN-POSP Procedure**

1. Initialize  $\Delta TWT_{\min} := M$  where  $M$  is a large value.
2. For each machine  $k = 1, \dots, m$  do
3.     For each batch  $i = \omega_k, \dots, k[1]$  do
4.         For each number of predecessors  $h = 0, \dots, i-1$  do
5.             For each number of periods  $p = 1, \dots, t(k, i)$  do
6.                 Calculate the  $\Delta EC$  and  $\Delta TWT$  values which result from
  - a) postponing batch  $i$  for  $p$  periods
  - b) if  $h > 0$  postponing all batches  $j = i-h, \dots, i-1$  each for  $t(k, j)$  periods.
7.                 If  $\Delta EC \geq 0$ , go to Step 4, otherwise, continue.
8.                 If  $\Delta TWT \geq \Delta TWT_{\min}$ , go to Step 3, otherwise, continue.
9.                 Set  $\Delta TWT_{\min} := \Delta TWT$ ,  $k^* := k$ ,  $i^* := i$ ,  $p^* := p$ ,  $h^* := h$ . Go to Step 3.
10.             End for
11.         End for
12.     End for
13.     If we have at least one  $\Delta EC < 0$  value in Step 6 then
  - a) postpone batch  $i^*$  on machine  $k^*$  for  $p^*$  periods and
  - b) if  $h^* > 0$  postpone all batches  $j = i^* - h^*, \dots, i^* - 1$  each for  $t(k^*, j)$  periods
  - c) return the modified schedule  $S^*$ .

Steps 6–8 ensure that only those EC reductions are considered which lead to the smallest TWT increase. The batches for moves to be implemented in the input schedule  $S$  are stored in Step 9. The computational effort for the MIN-POSP procedure is  $O(b^2T)$ .

Given an initial non-dominated set, the MIN-POSP is applied to each element of this set. The procedure is then applied in an iterative manner to the schedules obtained in the last iteration. Each schedule obtained in this way is inserted into the enlarged set of all solutions. Dominated solutions must be eventually removed

from this set after the application of the MIN-POSP procedure is completed. Therefore, improving a frontier by applying MIN-POSP is time-consuming. However, we expect it can significantly increase the quality of the Pareto-optimal set obtained by the NSGA-II approach.

#### 4.6 Overall approach

Three approaches based on the NSGA-II metaheuristic are considered. Each approach is based on the GGA encoding scheme described in Sect. 4.2.2. The basic GGA encoding scheme is extended by the encoding of additional information for the decoding process, namely the LIST, BD, and HYB representations. The different encoding and decoding schemes are summarized in Table 1.

The three approaches are called GGA-LIST, GGA-BD, and GGA-HYB, respectively where the abbreviation after GGA is inherited from the corresponding representation. They are depicted in Fig. 3.

The LS approach from Sect. 4.4 is applied within the three GAs each time a chromosome is changed by a crossover or mutation operation. Moreover, it is applied to each chromosome of the initial population after the chromosome is randomly generated. The LS-NSGA-II scheme is applied to each individual of a generation of the NSGA-II approach as soon as a population is completed. The

**Table 1** Summary of the encoding and decoding schemes

Representation	Basic encoding	Additional encoding	Decoding scheme
GGA-LIST	Groups of jobs (batches)	Random key $\theta$	LIST-ASAP for batches 1, ..., $[\theta b]$ , LIST-MEC for the remaining batches
GGA-BD	Groups of jobs (batches)	Random keys $\mu_i$ for batch-machine assignment, $\theta_i$ for the amount of idle time before starting batch $i$ , $\tilde{\theta}_k$ for inserting idle time before the end of the scheduling horizon	Based on the random keys
GGA-HYB	Groups of jobs (batches)	Additional encoding from LIST and BD, random key $\gamma$ which determines whether GGA-LIST or GGA-BD is used	Depends on the $\gamma$ value

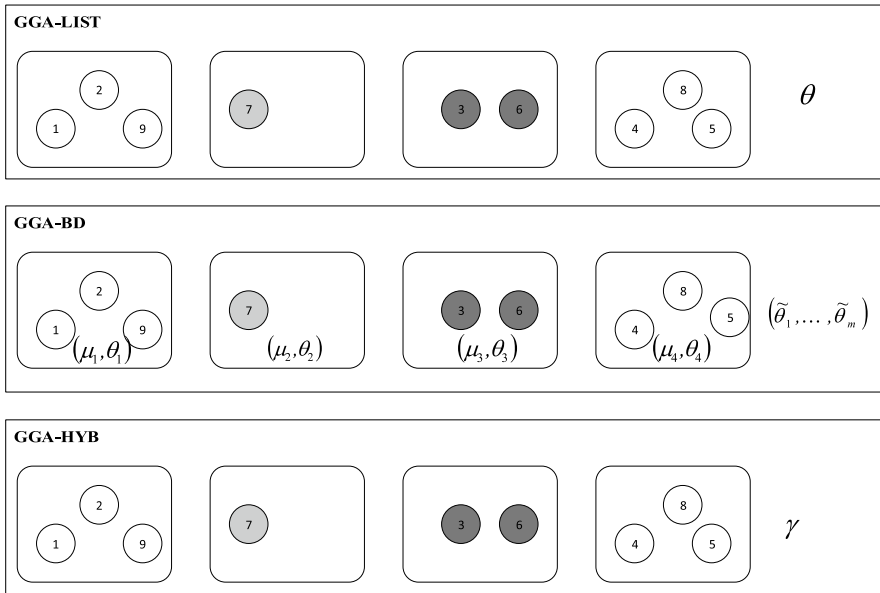


Fig. 3 Proposed GGA-type representations

MIN-POSP procedure is optionally used to improve the Pareto front obtained by the NSGA-II approach.

### 5 Computational experiments

#### 5.1 Design of experiments

We expect that the quality of the proposed heuristics depends on the number of jobs, the number of families, the maximum batch size, the job sizes, and the ready time and due date setting. Since problem (3) is NP-hard, the  $\epsilon$ -constraint method can only be applied to small-sized problem instances. Therefore, in a first experiment, a set of small-sized problem instances is generated according to the design of experiments (DOE) shown in Table 2.

The ready time and due date setting is similar to Mönch et al. (2005). We use the average batch size  $\bar{B} = B/\bar{s} = 2$  where  $\bar{s}$  is the average job size. The winter  $e_W$  and the summer electricity tariffs  $e_s$  are defined as follows:

$$e_W(t) := \begin{cases} 10, & 1 \leq t < \frac{1}{2}T \\ 8, & \text{otherwise} \end{cases}, \quad e_s(t) := \begin{cases} 10, & 1 \leq t < \frac{1}{3}T \\ 9, & \frac{1}{3}T \leq t < \frac{1}{2}T \\ 8, & \frac{1}{2}T \leq t < \frac{2}{3}T \\ 9, & \text{otherwise} \end{cases}. \quad (27)$$

**Table 2** DOE for small-sized problem instances

Factor	Level	Count
Number of families $F$	2, 3	2
Number of jobs per family $n_f$	5	1
Maximum batch size $B$ (in wafers)	4	1
Number of machines $m$	2	1
Job size	$s_j \sim DU[1, 3]$	1
Family processing times $p_f$	2 with probability 0.2 4 with probability 0.2 10 with probability 0.3 16 with probability 0.2 20 with probability 0.1	1
Job weights	$w_j \sim DU[1, 5]$	1
Job ready time	$r_j \sim DU\left[0, (\alpha/\bar{B}) \sum_{j=1}^n p_{s(j)}\right], \alpha \in \{0.25, 0.50\}$	2
Job due dates	$d_j - r_j \sim DU\left[0, (\beta/\bar{B}) \sum_{j=1}^n p_{s(j)}\right], \beta \in \{0.25, 0.50\}$	2
Electricity tariff	Winter rate, summer rate	2
Number of independent replications	5	5
Total number of problem instances		80

These winter and summer tariffs mimic to some extent the TOU rate plans offered by the Pacific Gas and Electric Company (TOU Rates-Pacific Gas and Electric Company 2019) with respect to the time windows over the course of 1 day and the relative difference between price levels. Semiconductor manufacturers consider information on their manufacturing processes as highly confidential. Hence, actual values of the energy consumption caused by processing batches in a furnace are not available. Instead of striving for decision support for a particular wafer fab, we provide general guidelines for managers to uncover savings potential. The  $e_W(t)$  and  $e_S(t)$  functions can easily be adapted for a real wafer fab.

Overall, we have 80 small-sized instances. These instances can be used to determine the Pareto frontier. This allows us to check whether the three heuristics are correctly implemented. Note that for the small-sized instances all the problem data is integer. Hence, the TWT and EC values are also integer and the  $\varepsilon$ -constraint method with a step size of 1 provides the entire Pareto front.

In addition, we consider large-sized instances. The corresponding design of experiments is shown in Table 3. Note that the largest instances contain 240 jobs. Such instances can be found in medium-sized wafer fabs (Mönch et al. 2013). The factors are the same as used for the small-sized instances.

We know from (Rocholl et al. 2018) that medium-sized instances for problem (17) can be solved to optimality using the  $\varepsilon$ -constraint method. Therefore, it is possible to compare the Pareto frontiers determined by the three heuristics with the true Pareto frontier. We consider problem instances with  $F = 2$ ,  $n_f = 20$ ,  $B \in \{2, 4, 8\}$  and  $F = 3$ ,  $n_f = 30$ ,  $B \in \{4, 8\}$ . The number of machines, processing times, and

**Table 3** DOE for large-sized problem instances

Factor	Level	Count
Number of families $F$	3, 6	2
Number of jobs per family $n_f$	20, 30, 40	3
Maximum batch size $B$ (in wafers)	25	1
Number of machines $m$	2, 5	2
Job size	$s_j \sim DU[1, 5], s_j \sim DU[1, 12], s_j \sim DU[1, 24]$	3
Family processing times $p_f$	2 with probability 0.2 4 with probability 0.2 10 with probability 0.3 16 with probability 0.2 20 with probability 0.1	1
Job weights	$w_j \sim DU[1, 5]$	1
Job ready times	$r_j \sim DU\left[0, (\alpha/\bar{B}) \sum_{j=1}^n p_{s(j)}\right], \alpha \in \{0.25, 0.50\}$	2
Job due dates	$d_j - r_j \sim DU\left[0, (\beta/\bar{B}) \sum_{j=1}^n p_{s(j)}\right], \beta \in \{0.25, 0.50\}$	2
Electricity tariff	Winter rate, summer rate	2
Number of independent replications	2	2
Total number of problem instances		576

weights are taken from Table 3. The maximum batch size  $B$  is measured in number of jobs. This leads to 40 problem instances for problem (17).

The large-sized problem instances from Table 3 are solved in another experiment. A fractional design is considered to assess the solution quality and the required computing time for the MIN-POSP procedure in an additional experiment. Therefore, we consider the design from Table 3 for  $n_f = 20$ . This leads to 192 problem instances with up to 120 jobs.

To examine the influence of the LS and LS-NSGA-II schemes described in Sect. 4.4, we conduct additional experiments with 20 instances randomly chosen from the set obtained by the design of experiments summarized in Table 3. Additional variants of each of the three heuristics are implemented where at least one of the LS and the LS-NSGA-II schemes are included. This leads to four variants of each heuristic to be compared.

The proposed bi-criteria heuristics determine an approximation of the Pareto frontier. Instead of comparing the heuristics based on objective function values, quality measures have to be derived from the approximation sets (Van Veldhuizen 1999; Zitzler et al. 2003; Coello Coello and Lamont 2004). Let  $Y_H$  be the set of solutions obtained by the heuristic  $H$  and  $Y_{true}$  be the corresponding set of Pareto-optimal solutions. The overall non-dominated vector generation (ONVG) measure is given by

$$ONVG(Y_H) := |Y_H|, \quad (28)$$

i.e., it is the number of non-dominated solutions found by  $H$ . The overall non-dominated vector generation ratio (ONVGR) measure is calculated as follows:

$$ONVGR(Y_H, Y_{true}) := ONVG(Y_H) / ONVG(Y_{true}). \tag{29}$$

Moreover, a distance measure is used that computes the mean distance of solutions provided by  $H$  to the nearest solution of the true Pareto front. Therefore, we consider:

$$d(y, \hat{y}) := ((TWT(y) - TWT(\hat{y}))^2 / (TWT^{max} - TWT^{min})^2 + (EC(y) - EC(\hat{y}))^2 / (EC^{max} - EC^{min})^2)^{1/2} \tag{30}$$

for  $y \in Y_H$  and  $\hat{y} \in Y_{true}$  where  $f_k^{max}$  and  $f_k^{min}$  are the maximum and minimum of the  $k$ -th objective function component found among the solutions from  $Y_H$  and  $Y_{true}$ . The average distance of a solution from  $Y_H$  to the closest solution in  $Y_{true}$  is then given by

$$dist(Y_H, Y_{true}) := \sqrt{\sum_{y \in Y_H} \left( \min_{\hat{y} \in Y_{true}} d(y, \hat{y}) \right)^2} / ONVG(Y_H). \tag{31}$$

The set coverage (CS) indicator determines the percentage of individuals in one set dominated by the individuals of the other set, i.e., we have

$$CS(Y_{true}, Y_H) := \left| \{y | y \in Y_H, \text{there is a } \hat{y} \in Y_{true}, \hat{y} \text{ dominates } y\} \right| / |Y_H|. \tag{32}$$

Moreover, the hypervolume (HV) indicator proposed by Zitzler and Thiele (1998) is used. It is a measure for the volume of the objective space dominated by a solution set. The HV indicator value for a solution set  $Y$  is given by

$$HV(Y) := (TWT^{max} - TWT(y_{(1)}))(EC^{max} - EC(y_{(1)})) + \sum_{i=1}^{|Y|-1} (TWT(y_{(i)}) - TWT(y_{(i+1)}))(EC^{max} - EC(y_{(i)})). \tag{33}$$

Here,  $y_{(i)}$  refers to the  $i$ -th solution in  $Y$  in descending order of  $TWT$  and  $TWT^{max}$ ,  $EC^{max}$  denote the maximum value found for the  $TWT$  and the  $EC$  values, respectively and is used as reference point. The relative HV index, abbreviated by HVR and defined by  $HVR(Y_H, Y_{true}) := HV(Y_H) / HV(Y_{true})$  is applied.

We are also interested in looking at the EC savings potential in relation to the TWT impairment. Therefore, we define the two measures

$$\lambda_{TWT}(I) := TWT^{max}(I) / TWT^{min}(I) \tag{34}$$

and

$$\lambda_{EC}(I) := EC^{min}(I) / EC^{max}(I) \tag{35}$$

for the Pareto front of an instance  $I$ . The pair  $(\lambda_{TWT}(I), \lambda_{EC}(I))$  indicates the increase of the TWT value relative to the smallest possible TWT value and the EC reduction relative to the largest occurring EC value.

When the  $\varepsilon$ -constraint method can be applied,  $Y_{true}$  is the set of Pareto-optimal solutions. For problem instances too large to be exactly solved in reasonable amount of computing time, the set  $Y_{true}$  is formed by all known solutions for this problem instance where dominated solutions have been removed. All heuristics are performed five times with different seeds for each instance since the NSGA-II approach has stochastic elements. The set of solutions is then formed by these five replications where dominated solutions are removed.

Instead of using a prescribed number of generations for the different GAs we allow a maximum computing time of 180 s for instances of Table 3, while the maximum of the remaining problem instances is only 60 s.

## 5.2 Parameter settings and implementation issues

Preliminary computational experiments with a limited set of problem instances are conducted to find appropriate parameter values of the different heuristics. We randomly choose 20 problem instances from the set presented in Table 3. The population size is taken from  $\{150, 300, 450\}$ , while the mutation probability  $p_M \in \{0.01, 0.05, 0.1\}$  is considered. For the sake of simplicity, the overall average of the HVR values of the solution sets found by the three heuristics are chosen to determine the parameter settings. As a result, we use 300 as population size and  $p_M = 0.01$  within the experiments. We do not use a specific crossover probability in our NSGA-II approach. The parameter  $\tau$  in the GGA-BD is set to 10% of the individuals of a generation.

The C++ programming language is used to code all the algorithms. The CPLEX 12.8 libraries are used to implement the  $\varepsilon$ -constraint method. The NSGA-II approach is implemented using the MOMHLib++ framework by Jaszkiwicz (2019). All computational experiments are performed on an Intel Core i7-2600 CPU 3.40 GHz PC with 16 GB RAM.

## 5.3 Computational results

### 5.3.1 Overview

We start by discussing computational results for small-sized instances of problem (3). This allows us to compare the heuristics with the true Pareto frontier obtained by the  $\varepsilon$ -constraint method. In a next step, results for experiments with the larger instances of problem (17) are presented. Again, the true Pareto frontier is obtained by the  $\varepsilon$ -constraint method. Since the  $\varepsilon$ -constraint method does not work for the



large-sized instances of problem (3), we next present results for large-sized instances where the set  $Y_{true}$  is formed by all known solutions as described above. We provide results that allow us to study the impact of the improvement heuristic and the two LS schemes. Finally, we discuss some managerial implications of the computational results.

### 5.3.2 Results for small-sized problem instances

Instead of comparing all instances individually, the instances are grouped according to the factor level values in all the tables found in this subsection. Results for the winter tariff,  $n_f = 20$ , and  $F = 3$  imply, for instance, that all other factors have been varied, but the tariff, the number of jobs per family, and the number of families have been constant at winter, 20, and 3, respectively. Table 4 shows the ONVGR and dist values for small-sized instances. The corresponding CS and HVR measure values are stated in Table 5. Best results are always marked in bold in the rest of the paper.

Figure 4 exemplifies sets of non-dominated solutions obtained by the NSGA-II approach with the different encoding schemes, namely GGA-LIST, GGA-BD, and GGA-HYB and the  $\epsilon$ -constraint method, abbreviated by OPT for a small-sized instance of problem (3).

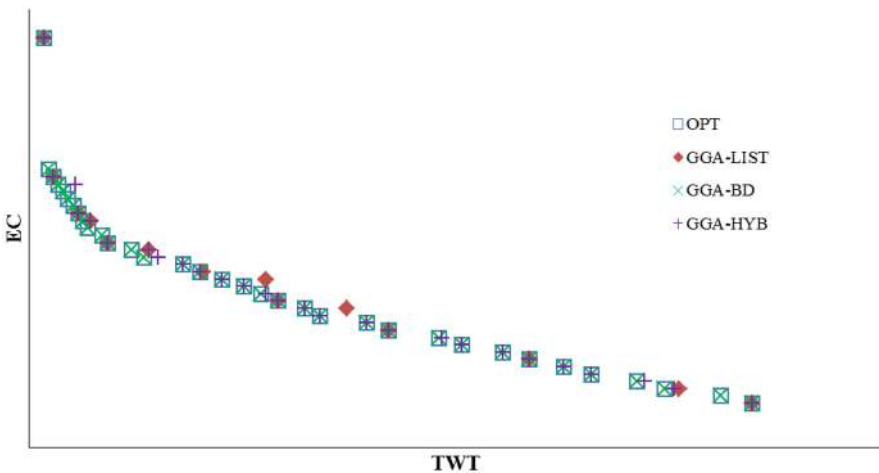
The computational results from Tables 4 and 5 demonstrate that the NSGA-II approach with the GGA-BD representation is able to find near-to-optimal solutions for small-sized instances. The GGA-BD representation outperforms the

**Table 4** ONVGR and dist values for small-sized problem instances

Tariff	F	$\alpha$	$\beta$	ONVGR			dist		
				GGA-LIST	GGA-BD	GGA-HYB	GGA-LIST	GGA-BD	GGA-HYB
Winter	2	0.25	0.25	0.216	<b>0.910</b>	0.696	0.011	<b>0.001</b>	0.002
			0.5	0.231	<b>0.941</b>	0.712	0.018	<b>0.002</b>	0.004
	0.5	0.25	0.283	<b>0.890</b>	0.663	0.036	<b>0.002</b>	0.008	
		0.5	0.316	<b>0.901</b>	0.689	0.008	<b>0.002</b>	<b>0.002</b>	
	3	0.25	0.25	0.339	<b>0.882</b>	0.607	<b>0.005</b>	0.061	0.070
		0.5	0.263	<b>0.831</b>	0.405	0.013	<b>0.006</b>	0.016	
Summer	2	0.25	0.25	0.285	<b>0.916</b>	0.691	0.024	<b>0.003</b>	0.006
			0.5	0.259	<b>0.980</b>	0.820	0.014	0.003	<b>0.002</b>
	0.5	0.25	0.264	<b>0.909</b>	0.739	0.022	<b>0.001</b>	0.007	
		0.5	0.418	<b>0.961</b>	0.693	0.011	<b>0.001</b>	0.004	
	3	0.25	0.25	0.422	<b>0.969</b>	0.703	<b>0.006</b>	0.036	0.070
		0.5	0.293	<b>0.860</b>	0.433	0.012	<b>0.003</b>	0.005	
Overall			0.25	0.304	<b>0.792</b>	0.480	0.007	<b>0.001</b>	0.004
			0.5	0.310	<b>0.809</b>	0.526	0.009	<b>0.001</b>	0.004
Overall				0.301	<b>0.885</b>	0.598	0.013	<b>0.008</b>	0.013

**Table 5** CS and HVR values for small-sized problem instances

Tariff	$F$	$\alpha$	$\beta$	CS			HVR		
				GGA-LIST	GGA-BD	GGA-HYB	GGA-LIST	GGA-BD	GGA-HYB
Winter	2	0.25	0.25	0.258	<b>0.129</b>	0.266	0.881	<b>0.980</b>	0.961
				0.470	<b>0.272</b>	0.310	0.882	0.981	<b>0.985</b>
		0.5	0.25	0.355	<b>0.265</b>	0.445	0.907	<b>0.981</b>	0.967
				0.256	<b>0.144</b>	0.340	0.936	<b>0.984</b>	0.977
	3	0.25	0.25	<b>0.420</b>	0.429	0.507	<b>0.952</b>	0.922	0.917
				0.457	<b>0.313</b>	0.682	0.929	<b>0.977</b>	0.948
		0.5	0.25	0.366	<b>0.361</b>	0.567	0.976	<b>0.982</b>	0.977
				0.522	<b>0.469</b>	0.613	0.970	<b>0.978</b>	0.974
				0.751	<b>0.154</b>	0.522	0.858	<b>0.987</b>	0.970
Summer	2	0.25	0.25	0.751	<b>0.154</b>	0.522	0.858	<b>0.987</b>	0.970
				0.732	<b>0.357</b>	0.633	0.892	<b>0.992</b>	0.987
		0.5	0.25	0.683	<b>0.194</b>	0.623	0.903	<b>0.994</b>	0.980
				0.621	<b>0.144</b>	0.454	0.934	<b>0.998</b>	0.986
	3	0.25	0.25	0.677	<b>0.478</b>	0.754	0.943	<b>0.947</b>	0.916
				0.784	<b>0.410</b>	0.852	0.921	<b>0.981</b>	0.955
		0.5	0.25	0.705	<b>0.496</b>	0.843	0.964	<b>0.990</b>	0.979
				0.832	<b>0.541</b>	0.804	0.936	<b>0.982</b>	0.972
				0.556	<b>0.322</b>	0.576	0.924	<b>0.979</b>	0.966



**Fig. 4** Example of the non-dominated sets of a small-sized problem instance

GGA-LIST representation for all performance measures under almost all experimental conditions. The largest difference can be observed for the ONVGR measure. This indicates that the GGA-BD representation is able to encode a much larger portion of the solution space than the remaining representations. The average CS measure value reveals that on average around 68% of the solutions found

with the GGA-BD representation are Pareto-optimal. The GGA-BD representation again leads to the best average HVR value. However, for two sets of factor combinations, namely both tariffs,  $F = 3, \alpha = 0.25, \beta = 0.25$ , the GGA-LIST representation shows an almost equal or even better HVR value despite providing a much smaller set of non-dominated solutions. Apparently, in some situations it can be reasonable to search for a limited set of high-quality solutions. The GGA-HYB representation shows average values close to the ones of the GGA-LIST for both the dist and the CS measures. Compared to the GGA-BD representation, fewer solutions are found but still the average HVR value is close to the one found by the GGA-BD representation. The non-dominated sets exemplified in Fig. 4 support these observations.

### 5.3.3 Results for problem (17)

Results of the computational experiments for problem (17) are presented in Tables 6 and 7.

**Table 6** ONVGR and dist values for medium-sized problem instances (special case)

Tariff	$n_f$	$F$	$M$	$B$	ONVGR			Dist			
					GGA-LIST	GGA-BD	GGA-HYB	GGA-LIST	GGA-BD	GGA-HYB	
Winter	20	2	2	2	0.190	<b>0.910</b>	0.715	<b>0.002</b>	<b>0.002</b>	0.004	
				4	0.403	<b>0.896</b>	0.779	0.008	0.007	<b>0.004</b>	
		8	0.359	<b>0.881</b>	0.840	0.009	<b>0.001</b>	<b>0.001</b>			
			5	2	0.303	<b>0.757</b>	0.320	0.003	<b>0.001</b>	0.003	
		4	0.167	<b>0.851</b>	0.578	0.004	<b>0.001</b>	<b>0.001</b>			
			8	0.114	<b>0.897</b>	0.774	0.002	<b>0.001</b>	<b>0.001</b>		
	30	3	2	4	0.410	<b>0.875</b>	0.576	<b>0.001</b>	0.002	<b>0.001</b>	
				8	0.227	<b>0.928</b>	0.564	0.006	<b>0.001</b>	0.002	
		5	4	0.322	<b>0.617</b>	0.332	0.008	<b>0.003</b>	<b>0.003</b>		
			8	0.309	<b>0.810</b>	0.565	0.005	<b>0.001</b>	0.002		
		20	2	2	2	0.218	<b>0.910</b>	0.759	0.005	<b>0.001</b>	0.005
					4	0.451	<b>0.946</b>	0.838	0.005	<b>0.000</b>	0.001
Summer	8	0.346	<b>0.903</b>	0.821	0.015	<b>0.003</b>	<b>0.003</b>	<b>0.003</b>			
									5	2	0.369
	4	0.155	<b>0.924</b>	0.418	0.005	<b>0.001</b>	0.002				
		8	0.080	<b>0.976</b>	0.747	0.008	<b>0.000</b>	0.002			
	30	3	2	4	0.390	<b>0.825</b>	0.558	0.006	<b>0.001</b>	0.004	
				8	0.295	<b>0.945</b>	0.698	0.010	<b>0.001</b>	0.002	
5		4	0.351	<b>0.509</b>	0.378	0.007	<b>0.002</b>	0.005			
		8	0.269	<b>0.914</b>	0.551	0.004	<b>0.002</b>	<b>0.002</b>			
Overall				0.286	<b>0.847</b>	0.611	0.006	<b>0.002</b>	0.003		

**Table 7** CS and HVR values for medium-sized problem instances (special case)

Tariff	$n_f$	$F$	$M$	$B$	CS			HVR		
					GGA-LIST	GGA-BD	GGA-HYB	GGA-LIST	GGA-BD	GGA-HYB
Winter	20	2	2	2	0.200	<b>0.186</b>	0.375	0.822	<b>0.977</b>	0.940
					0.133	<b>0.073</b>	0.167	0.928	<b>0.961</b>	0.938
					0.188	<b>0.068</b>	0.120	0.890	<b>0.960</b>	0.938
		5	2	<b>0.358</b>	0.803	0.413	0.948	<b>0.963</b>	0.952	
			4	0.325	<b>0.196</b>	0.562	0.913	<b>0.948</b>	0.924	
			8	<b>0.111</b>	0.128	0.320	0.936	0.976	<b>0.980</b>	
	30	3	2	4	<b>0.339</b>	0.391	0.477	0.972	0.968	<b>0.977</b>
					0.202	<b>0.177</b>	0.242	0.950	<b>0.983</b>	0.975
					0.630	0.968	<b>0.628</b>	0.916	<b>0.937</b>	0.932
		5	4	0.513	<b>0.308</b>	0.541	<b>0.981</b>	0.919	<b>0.981</b>	
			8	0.627	<b>0.188</b>	0.639	0.812	<b>0.977</b>	0.918	
			4	0.724	<b>0.066</b>	0.214	0.957	<b>0.992</b>	0.976	
Summer	20	2	2	2	0.627	<b>0.188</b>	0.639	0.812	<b>0.977</b>	0.918
					0.724	<b>0.066</b>	0.214	0.957	<b>0.992</b>	0.976
					0.558	0.104	<b>0.084</b>	0.858	<b>0.975</b>	0.947
		5	2	<b>0.902</b>	0.964	0.920	0.920	<b>0.974</b>	0.941	
			4	0.744	<b>0.480</b>	0.860	0.948	<b>0.987</b>	0.968	
			8	0.200	<b>0.147</b>	0.600	0.905	<b>0.998</b>	0.978	
	30	3	2	4	0.867	<b>0.471</b>	0.850	0.929	<b>0.959</b>	0.948
					0.851	<b>0.118</b>	0.570	0.922	<b>0.989</b>	0.961
					0.923	0.990	0.930	0.904	<b>0.953</b>	0.918
		5	4	<b>0.923</b>	0.990	0.930	0.904	<b>0.953</b>	0.918	
			8	0.747	<b>0.426</b>	0.807	0.970	<b>0.980</b>	0.965	
			Overall		0.507	<b>0.363</b>	0.516	0.919	<b>0.969</b>	0.953

The results in Tables 6 and 7 largely confirm the findings of the small-sized instances of problem (3) for experiments conducted for problem (17). The GGA-BD representation outperforms the remaining representations with respect to the ONVGR and dist measures. Applying the GGA-LIST representation can provide a large fraction of Pareto-optimal solutions under some experimental conditions as can be seen from the average CS value.

### 5.3.4 Results for large-sized problem instances

Computational results for the large-sized instances from Table 3 are shown in Tables 8 and 9. The interested reader is referred to the electronic supplement for detailed computational results for all factor combinations.

No clear superiority of one representation over the others can be found. Apparently, the advantage of the GGA-BD representation cannot be confirmed here. In particular, a decline of the ONVGR values with a larger total number of jobs  $n = Fn_f$  can be observed. This indicates that the criteria space is too large to be searched efficiently by the GGA-BD approach.

**Table 8** ONVGR and Dist values for large-sized problem instances

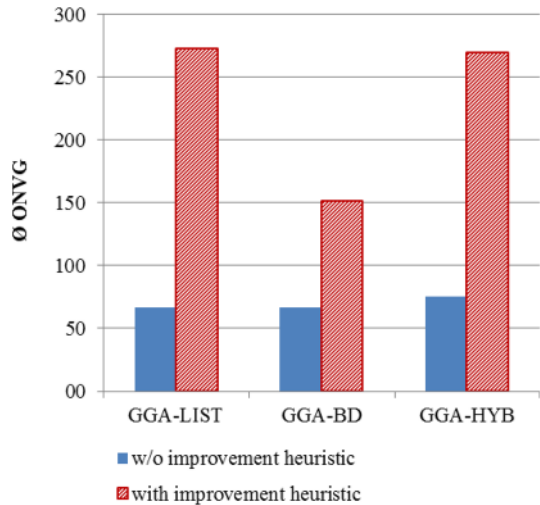
Tariff	$n_f$	$F$	ONVGR			Dist		
			GGA-LIST	GGA-BD	GGA-HYB	GGA-LIST	GGA-BD	GGA-HYB
Winter	20	3	0.669	<b>1.039</b>	0.804	0.006	0.004	<b>0.003</b>
		6	0.828	0.837	<b>0.890</b>	0.004	0.006	<b>0.002</b>
	30	3	0.758	<b>0.929</b>	0.583	0.004	0.005	<b>0.003</b>
		6	<b>0.883</b>	0.719	0.528	0.003	0.008	<b>0.002</b>
	40	3	0.862	<b>0.895</b>	0.543	0.004	0.007	<b>0.003</b>
		6	<b>0.916</b>	0.620	0.588	0.003	0.009	<b>0.002</b>
Summer	20	3	0.710	0.845	<b>0.880</b>	0.005	<b>0.002</b>	0.003
		6	0.843	0.618	<b>0.902</b>	0.003	0.004	<b>0.002</b>
	30	3	0.812	0.748	<b>0.848</b>	0.004	0.003	<b>0.002</b>
		6	<b>0.902</b>	0.579	0.850	<b>0.002</b>	0.009	0.003
	40	3	0.875	0.675	<b>0.891</b>	0.003	0.006	<b>0.003</b>
		6	<b>0.922</b>	0.480	0.870	<b>0.003</b>	0.015	0.003
Overall			<b>0.832</b>	0.749	0.765	0.004	0.007	<b>0.003</b>

**Table 9** CS and HVR values for large-sized problem instances

Tariff	$n_f$	$F$	CS			HVR		
			GGA-LIST	GGA-BD	GGA-HYB	GGA-LIST	GGA-BD	GGA-HYB
Winter	20	3	0.740	0.576	<b>0.470</b>	0.972	0.941	<b>0.991</b>
		6	0.581	0.757	<b>0.469</b>	0.979	0.844	<b>0.991</b>
	30	3	0.601	0.611	<b>0.583</b>	0.978	0.908	<b>0.985</b>
		6	<b>0.492</b>	0.786	0.528	0.978	0.784	<b>0.987</b>
	40	3	0.543	0.731	<b>0.543</b>	0.975	0.857	<b>0.986</b>
		6	<b>0.438</b>	0.805	0.588	<b>0.988</b>	0.727	0.987
Summer	20	3	0.730	<b>0.423</b>	0.606	0.977	0.936	<b>0.988</b>
		6	0.559	<b>0.556</b>	0.626	0.977	0.816	<b>0.986</b>
	30	3	0.612	<b>0.471</b>	0.673	0.977	0.889	<b>0.984</b>
		6	<b>0.375</b>	0.679	0.725	<b>0.985</b>	0.745	0.980
	40	3	<b>0.508</b>	0.640	0.647	0.980	0.832	<b>0.982</b>
		6	<b>0.352</b>	0.743	0.703	<b>0.989</b>	0.665	0.982
Overall			<b>0.544</b>	0.648	0.597	0.980	0.829	<b>0.986</b>

Considering the dist and the HVR measures, the GGA-HYB representation outperforms the remaining representations. We observe that the GGA-LIST representation leads to higher-quality schedules for all performance measures when applied to instances with six job families compared to instances with only three families. On the contrary, the GGA-BD representation can produce better solutions if the jobs are distributed over three families only. The number of families

**Fig. 5** Size of the Pareto frontiers depending on the MIN-POSP procedure



does not influence the performance of the GGA-HYB representation which can be expected as it combines features of the two remaining representations.

### 5.3.5 Impact of the improvement heuristic and the local search schemes

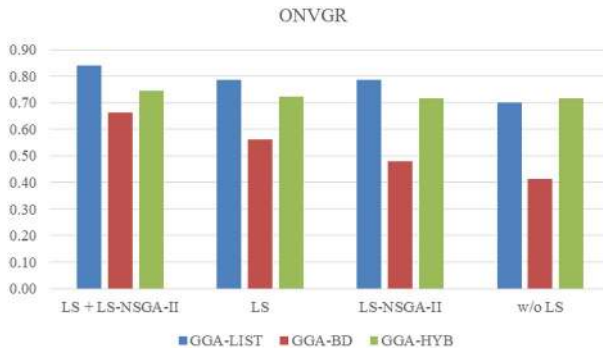
The impact of applying the MIN-POSP procedure from Sect. 4.4 on the number of solutions in the corresponding Pareto frontier is shown in Fig. 5 as the average ONVG value over all problem instances described in Sect. 5.1. The MIN-POSP procedure is called improvement heuristic in this figure.

On the one hand, the procedure is able to find a large number of additional non-dominated schedules for all assessed representations. On the other hand, the computing time can become very long. It depends on the original number of solutions being the input and the time horizon and number of batches of the corresponding schedule. The additional computational burden per instance is between less than one second and one hour with an average value of 87 s. From the experiments conducted it appears that the additional time requirement can be limited to around 2 min if the MIN-POSP procedure is applied to instances with not more than 60 batches.

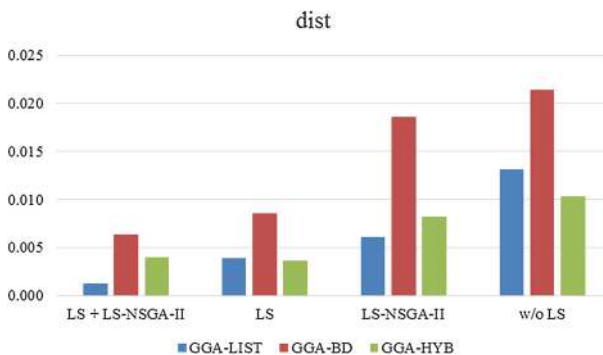
Figures 6 and 7 depict the ONVGR and dist measure values for the heuristics with different levels of LS consideration. The average values obtained from the experiments with all the 20 instances are shown.

The leftmost group of bars presents the values of the variant of the heuristics where both the LS and the LS-NSGA-II scheme are applied. The second and third groups from the left show the values for the variants where only the scheme shown under the group is used while “w/o LS” indicates that no LS at all is applied.

Figures 6 and 7 confirm that LS positively impacts the solution quality of NSGA-II-type heuristics. From Fig. 6 we see that the number of solutions found increases if



**Fig. 6** Average ONVGR values for different levels of local search consideration



**Fig. 7** Average dist values for different levels of local search consideration

either one of the proposed schemes is applied. The largest number of non-dominated solutions is found when both LS schemes are used. The ONVGR measure improves most for the GGA-BD representation whereas the GGA-HYB representation does not profit as much with respect to this measure. However, Fig. 7 shows a positive impact of LS on the distance measure for all three heuristics. Obviously the application of both LS schemes can move solutions closer to the true Pareto frontier.

### 5.3.6 Managerial insights from the computational experiments

Table 10 shows the EC savings in relation to the observed TWT increase. Note that we use only Pareto frontiers determined by the GGA-HYB since the differences between the different heuristics are fairly small.

The following managerial insights can be derived from the conducted experiments and Table 10:

**Table 10** EC savings potential in relation to the TWT impairment

$n_f$	$m$	$\alpha$	$\beta$	Winter			Summer				
				ONVG	$\lambda_{TWT}$	$\lambda_{EC}$	ONVG	$\lambda_{TWT}$	$\lambda_{EC}$		
20	2	0.25	0.25	47	3.051	0.868	82	2.812	0.906		
			0.50	49	4.381	0.848	78	3.928	0.891		
		0.50	0.25	87	3.899	0.785	117	2.999	0.837		
			0.50	71	4.713	0.793	93	3.338	0.841		
	5	0.25	0.25	89	3.782	0.710	127	3.415	0.779		
			0.50	73	4.130	0.779	86	3.720	0.803		
		0.50	0.25	85	2.479	0.603	120	2.499	0.578		
			0.50	79	2.657	0.628	102	2.652	0.644		
			2	0.25	0.25	46	3.295	0.849	81	3.373	0.891
					0.50	56	4.680	0.842	74	4.147	0.880
5	0.25	0.25	79	3.858	0.782	98	2.908	0.834			
		0.50	74	4.555	0.792	92	3.263	0.839			
	0.50	0.25	97	3.910	0.763	117	3.503	0.831			
		0.50	82	3.942	0.766	92	3.573	0.799			
		0.25	85	2.587	0.573	106	2.597	0.570			
		0.50	79	2.579	0.606	97	2.558	0.609			
40	2	0.25	0.25	52	3.225	0.850	73	3.232	0.892		
			0.50	49	3.994	0.850	70	3.537	0.889		
		0.50	0.25	74	4.409	0.799	77	3.135	0.834		
			0.50	80	4.554	0.795	79	3.463	0.835		
	5	0.25	0.25	93	4.110	0.754	112	3.600	0.807		
			0.50	78	4.119	0.785	83	3.557	0.816		
		0.50	0.25	84	2.744	0.580	100	2.663	0.589		
			0.50	74	2.440	0.613	91	2.419	0.632		
			Overall			73	3.671	0.751	93	3.204	0.784

1. Schedules with a small EC value tend to schedule batches in periods with small  $e(t)$  values, i.e. into off-peak periods. These schedules contribute to better balancing demand and supply of electricity.
2. In a highly loaded wafer fab, the room for EC reduction is smaller than in a wafer fab with moderate load. Small  $\alpha$  values mimic a high-load. The highest TWT impairment can be observed for tight due dates, represented by  $\beta = 0.25$ , and wide-spread ready times dates given by  $\alpha = 0.50$ . In this situation, a formation of full batches, i.e. reaching small EC values, is often only possible if some jobs are delayed for a long time. This leads to large TWT values.
3. The smallest EC improvement potential can be observed for a small number of jobs within each family. In this situation, it is not very likely that most of the batches are fully loaded. Hence, the number of batches is high and the EC values



- are large. This insight might be taken into account when product mix decisions are made by management.
4. A larger processing flexibility represented by a larger number of parallel machines increases the ability for EC reductions.
  5. The EC improvement potential is slightly smaller for the winter tariff. However, at the same time the EC improvement is reached at the expense of a larger TWT impairment in comparison to the summer tariff. The summer tariff results in smaller EC improvement, however, the TWT impairment is also smaller. This is caused by the larger number of available options of the summer tariff. This is also reflected by a larger average number of non-dominated solutions in the case of the summer tariff. Therefore, TOU tariffs with more segments seem to be beneficial.

## 6 Conclusions and future research

In this paper, we discussed a bi-criteria scheduling problem for identical parallel batch processing machines. The TWT and EC performance measures were used. A MILP formulation is proposed. In addition, we provided a second, much simpler MILP for a special case where structural properties of Pareto-optimal solutions were provided. Both MILP models were solved using the  $\epsilon$ -constraint method. To tackle large-sized problem instances within a reasonable amount of computing time, we proposed three GGAs that are embedded into a NSGA-II-type framework. They differ in representations and decoding schemes. The performance of the GA variants was assessed using randomly generated problem instances. Moreover, the  $\epsilon$ -constraint method was used to assess the correct implementation of the heuristic approaches and to check the performance of the GA-type heuristics for medium-sized instances of the special case with the TWC performance measure instead of the TWT measure. Overall, we were able to demonstrate by the computational experiments that some of the proposed heuristics perform very well. Senior operators in wafer fabs can apply the proposed algorithms to choose furnace schedules that find a compromise between tardiness and sustainability goals. However, large EC reductions in highly loaded wafer fabs are only possible at the expense of large TWT values. A larger number of incompatible job families and of parallel machines are preferred from an EC reduction point of view.

This paper contributes to sustainable manufacturing from two points of view. First, it allows for computing schedules with small EC values, which might result in a small CO<sub>2</sub> emission. Second, schedules with small EC values tend to have many batches in the off-peak periods, i.e. they balance the demand and supply of electricity. As a result, less backup infrastructure (which often generates electricity in a more dirty way) is required. Again, reduced CO<sub>2</sub> emissions are likely to be the result.

There are several directions for future research. First of all, although the  $\epsilon$ -constraint method used in this paper is able to compute the exact Pareto front for the used integer-valued problem instances, it is interesting to look at several recent efficient implementations of this method for the specific instances and general

instances, for example the one provided by Mavrotas and Florios (2013). Secondly, the problem setting can be generalized by considering unrelated parallel machines instead of identical ones as in the present paper. Moreover, the energy consumption can be made family-dependent. We believe that it is also desirable to model standby times and sequence-dependent warm-up and cool-down processes since they lead to a different amount of energy consumption. It is also interesting to explore in addition to TOU tariffs different types of real-time energy pricing in the context of the scheduling problem at hand. As a third direction, we are interested in designing sampling algorithms for the present problem to obtain robust schedules since the ready times are typically uncertain in a real-world wafer fab. Integrating the proposed scheduling technique in a global scheduling approach for an entire wafer fab as, for instance, the one proposed by Mönch et al. (2007), is also desirable.

**Acknowledgements** The research is funded in parts by a research grant from the University of Hagen within the MaXFab project. The authors gratefully acknowledge this financial support. Open Access funding provided by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### Appendix

$$\min \left[ E_{TWT} \sum_{j=1}^n w_j T_j + E_{EC} \left( \sum_{i=1}^b \sum_{t=1}^T \sum_{k=1}^m \sum_{\tau=t}^{t+P_{f(i)}-1} e(\tau) y_{ik} \right) \right] \tag{36}$$

subject to (5)–(16)

$$(1 - E_{TWT}) \sum_{j=1}^n w_j T_j \leq \varepsilon_{TWT} \tag{37}$$

$$(1 - E_{EC}) \sum_{i=1}^b \sum_{t=1}^T \sum_{k=1}^m \sum_{\tau=t}^{t+P_{f(i)}-1} e(\tau) y_{ik} \leq \varepsilon_{EC}. \tag{38}$$

The quantities  $E_{TWT}, E_{EC} \in \{0, 1\}$ ,  $E_{TWT} + E_{EC} = 1$ , and  $\varepsilon_{TWT}, \varepsilon_{EC} \in \mathbb{R}$  are parameters of the model. For  $E_{TWT} = 1$  and  $E_{EC} = 0$  the model minimizes the TWT value whereas the EC value is restricted to  $\varepsilon_{EC}$ . In the case of  $E_{TWT} = 0$  and  $E_{EC} = 1$ , the model aims for a EC minimization where the TWT value of the schedule is restricted to  $\varepsilon_{TWT}$ . The model (36)–(38) and (5)–(16) is iteratively solved. The first iteration starts with  $E_{TWT} = 1$ ,  $E_{EC} = 0$ ,  $\varepsilon_{TWT} = 0$ , and

$\varepsilon_{EC} = M$  where  $M$  is determined using the piecewise constant EC function. The solution is a schedule  $S$  with objective function value  $TWT(S)$  where the EC value is restricted to  $M$ . The MILP is then solved a second time with the settings  $E_{TWT} = 0, E_{EC} = 1, \varepsilon_{TWT} = TWT(S)$ . The result is a Pareto optimal schedule. The next iteration starts from  $E_{TWT} = 0, E_{EC} = 1, \varepsilon_{EC} = 0$ , and  $\varepsilon_{TWT} = TWT(S) - 1$ . The setting  $\varepsilon_{TWT} = TWT(S) - 1$  is reasonable since the TWT and EC values are integers due to the integer parameters (see Sect. 5.1). This procedure is repeated until the MILP becomes infeasible for the parameters  $\varepsilon_{EC}$  and  $\varepsilon_{TWT}$ .

## References

- Akbar M, Irohara T (2018) Scheduling for sustainable manufacturing: a review. *J Clean Prod* 2005:866–883
- Albadi MH, El-Saadany EF (2008) A summary of demand response in electricity markets. *Electr Power Syst Res* 78(11):1989–1996
- Brown E, Sumichrast R (2003) Impact of the replacement heuristic in a grouping genetic algorithm. *Comput Oper Res* 30:1575–1593
- Brucker P, Gladky A, Hoogeveen JA, Kovalyov MY, Potts CN, Tautenhahn T, van den Velde SL (1998) Scheduling a batching machine. *J Sched* 1(1):31–54
- Cabo M, González-Velarde JL, Possani E, Ríos Solís YA (2018) Bi-objective scheduling on a restricted batching machine. *Comput Oper Res* 100:201–210
- Che A, Zhang S, Wu X (2017) Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. *J Clean Prod* 156:688–697
- Cheng J (2017) Multi-criteria batch scheduling under time-of-use tariffs. Ph.D. thesis. University of Evry-Val d'Essonne, Evry, Northwestern Polytechnical University, Xi'an
- Cheng J, Chu F, Xia W, Ding J, Ling X (2014) Bi-objective optimization for single-machine batch scheduling considering energy cost. In: Proceedings of the 2014 International Conference on Control, Decision and Information Technologies (CoDIT), Metz, pp 236–241
- Cheng J, Chu F, Chu C, Xia W (2016a) Bi-objective optimization of single-machine batch scheduling under time-of-use electricity prices. *RAIRO Oper Res* 50(4–5):715–732
- Cheng J, Chu F, Liu M, Xia W (2016b) Single-machine batch scheduling under time-of-use tariffs: new mixed-integer programming approaches. In: Proceedings of the 2016 IEEE international conference on systems, man, and cybernetics (SMC), pp 3498–3503. <https://doi.org/10.1109/SMC.2016.7844775>
- 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. *Comput Ind Eng* 112:721–734
- Chiang TC, Cheng HC, Fu LC (2010) A memetic algorithm for minimizing total weighted tardiness on parallel batch machines with incompatible job families and dynamic job arrival. *Comput Oper Res* 37(12):2257–2269
- Chien C, Dauzère-Pérès S, Ehm H, Fowler J, Jiang Z, Krishnaswamy S, Mönch L, Uzsoy R (2011) Modeling and analysis of semiconductor manufacturing in a shrinking world: challenges and successes. *Eur J Ind Eng* 5(3):254–271
- Coello Coello CA, Lamont GB (2004) An introduction to multi-objective evolutionary algorithms and their applications. In: Coello Coello CA, Lamont GB (eds) Applications of multi-objective evolutionary algorithms. World Scientific, Singapore, pp 1–28
- Deb K, Goel T (2001) A hybrid multi-objective evolutionary approach to engineering shape design. In: Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization, LNCS 1993, Zurich, pp 385–399
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Ehrgott M (2010) Multicriteria optimization, 2nd edn. Springer, New York
- Falkenauer E (1996) A hybrid grouping genetic algorithm for bin packing. *J Heuristics* 2:5–30
- Falkenauer E (1998) Genetic algorithms and grouping problems. Wiley, Chichester

- Finn P, Fitzpatrick C, Connolly D, Leahy M, Relihan L (2011) Facilitation of renewable electricity using price based appliance control in Ireland's electricity market. *Energy* 36(5):2952–2960
- Gahm C, Denz F, Dirr M, Tuma A (2016) Energy-efficient scheduling in manufacturing companies: a review and research framework. *Eur J Oper Res* 248:744–757
- Gao K, Huang Y, Sadollah A, Wang L (2019) A review of energy-efficient scheduling in intelligent production systems. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-019-00122-6>
- Giret A, Trentesaux D, Prabh V (2015) Sustainability in manufacturing operations scheduling: a state of the art review. *J Manuf Syst* 37(1):126–140
- Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading
- Goncalves JF, Mendes JJM, Resende MGC (2008) A genetic algorithm for the resource constrained multi-project scheduling problem. *Eur J Oper Res* 189:1171–1190
- Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann Discrete Math* 5:287–326
- IRENA (2019) Innovation landscape brief: time-of-use tariffs. International Renewable Energy Agency, Abu Dhabi
- Jaskiewicz A (2019) MOMHLIB++: multiple objective metaheuristics library in C++. <https://github.com/derino/maponoc/tree/master/libs/libmomh-1.91.3>. Accessed 30 Apr 2019
- Jia Z-H, Zhang Y-I, Leung JY-T, Li K (2017) Bi-criteria ant colony optimization algorithm for minimizing makespan and energy consumption on parallel batch machines. *Appl Soft Comput* 55:226–237
- Jia Z-H, Wang Y, Wu C, Yang Y, Zhang X-Y, Chen H-P (2019) Multi-objective energy-aware batch scheduling using ant colony optimization algorithm. *Comput Ind Eng* 131:41–56
- Landa Silva JD, Burke EK (2002) A tutorial on multiobjective metaheuristics for scheduling and time-tabling. [http://uuhost.uantwerpen.be/eume/workshops/momhjdls\\_momh2002.pdf](http://uuhost.uantwerpen.be/eume/workshops/momhjdls_momh2002.pdf). Accessed 20 Apr 2019
- Li L, Qiao F, Wu Q (2009) ACO-based multi-objective scheduling of parallel batch processing machines with advanced process control constraints. *Int J Adv Manuf Technol* 44(9):985–994
- Liu C-H (2014) Approximate trade-off between minimisation of total weighted tardiness and minimisation of carbon dioxide (CO<sub>2</sub>) emissions in bi-criteria batch scheduling problem. *Int J Comput Integr Manuf* 27(8):579–771
- Mason SJ, Kurz M, Pohl LM, Fowler JW, Pfund ME (2007) Random keys implementation of NSGA-II for semiconductor manufacturing scheduling. *Int J Inf Technol Intell Comput* 2(3)
- Mathirajan M, Sivakumar AI (2006) A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *Int J Adv Manuf Technol* 29(9–10):990–1001
- Mavrotas G, Florios K (2013) An improved version of the augmented e-constraint method (AUGMECON2) for Finding the exact pareto set in multi-objective integer programming problems. *Appl Math Comput* 219(18):9652–9669
- Merkert L, Harjunkoski I, Isaksson A, Säynevirta S, Saarela A, Sand G (2015) Scheduling and energy—industrial challenges and opportunities. *Comput Chem Eng* 72(2):183–198
- Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs, 3rd edn. Springer, Berlin
- Mönch L, Balasubramanian H, Fowler JW, Pfund ME (2005) Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. *Comput Oper Res* 32:2731–2750
- Mönch L, Schabacker R, Pabst D, Fowler JW (2007) Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. *Eur J Oper Res* 177(3):2100–2118
- Mönch L, Fowler JW, Dauzère-Pérès S, Mason SJ, Rose O (2011) A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *J Sched* 14(6):583–595
- Mönch L, Fowler JW, Mason SJ (2013) Production planning and control for wafer fabrication facilities: modeling, analysis, and systems. Springer, New York
- Moon J-Y, Shin K, Park J (2013) Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. *Int J Adv Manuf Technol* 68(1–4):523–535
- Pina A, Silva C, Ferrão P (2012) The impact of demand side management strategies in the penetration of renewable electricity. *Energy* 41:128–137
- Potts CN, Kovalyov MY (2000) Scheduling with batching: a review. *Eur J Oper Res* 120:228–249

- Reichelt D, Mönch L (2006) Multiobjective scheduling of jobs with incompatible families on parallel batch machines. In: Gottlieb J, Raidl GR (eds) *Evolutionary computation in combinatorial optimization*. EvoCOP 2006. Lecture Notes in Computer Science, vol 3906. Springer, Berlin, Heidelberg, pp 209–221
- Rocholl J, Mönch L, Fowler JW (2018) Electricity power cost-aware scheduling of jobs on parallel batch processing machines. In: *Proceedings of the 2018 Winter Simulation Conference*, Gothenburg, pp 3420–3431
- Scholl W (2017) Private communication
- Singapore Government (2019) Energy efficiency in the microelectronics industry. <https://www.e2singapore.gov.sg/DATA/0/docs/NewsFiles/Energy%20efficiency%20in%20the%20microelectronics%20industry%20v2.pdf>. Accessed 15 Apr 2019
- Sobeyko O, Mönch L (2011) A comparison of heuristics to solve a single machine batching problem with unequal ready times of the jobs. In: *Proceedings of the 2011 Winter Simulation Conference*, Phoenix, AZ, pp 2011–2020
- Sobeyko O, Mönch L (2015) Grouping genetic algorithms for solving single machine multiple orders per job scheduling problems. *Ann Oper Res* 235(1):709–739
- Stoll P, Brandt N, Nordström L (2014) Including dynamic CO<sub>2</sub> intensity with demand response. *Energy Policy* 65:490–500
- Taiwan Semiconductor Manufacturing Company Limited (TSMC) Annual Report (2019) [http://www.tsmc.com/download/ir/annualReports/2016/english/e\\_7\\_2.html](http://www.tsmc.com/download/ir/annualReports/2016/english/e_7_2.html). Accessed 15 Apr 2019
- TOU Rates—Pacific Gas & Electric Company (2019) [https://www.pge.com/en\\_US/small-medium-business/your-account/rates-and-rate-options/time-of-use-rates.page](https://www.pge.com/en_US/small-medium-business/your-account/rates-and-rate-options/time-of-use-rates.page). Accessed 15 Oct 2019
- Uzsoy R (1995) Scheduling batch processing machines with incompatible families. *Int J Prod Res* 33(10):2685–2708
- Van Veldhuizen DA (1999) *Multiobjective evolutionary algorithms: classifications, analysis, and new innovations*. Air Force Institute of Technology, Department of Electrical and Computer Engineering, Ohio
- Vidal T, Crainic TG, Gendreau M, Prins C (2015) Timing problems and algorithms: time decisions for sequences of activities. *Networks* 65(2):102–128
- Wang Y, Li L (2013) Time-of-use based electricity demand response for sustainable manufacturing systems. *Energy* 63(15):233–244
- Wang S, Liu M, Chu F, Chu C (2016) Bi-objective optimization of a single machine batch scheduling problem with energy cost consideration. *J Clean Prod* 137:1205–1215
- Yu C-M, Chien C-F, Kuo C-J (2017) Exploit the value of production data to discover opportunities for saving power consumption of production tools. *IEEE Trans Semicond Manuf* 30(4):345–350
- Zhang H, Zhao F, Fang K, Sutherland J (2014) Energy-conscious flow shop scheduling under time-of-use electricity tariffs. *CIRP Ann Manuf Technol* 63:37–40
- Ziarnetzky T, Kannaian T, Jimenez J, Mönch L (2017) Incorporating elements of a sustainable and distributed generation system into a production planning model for a wafer fab. In: *Proceedings of the 2017 Winter Simulation Conference*, Las Vegas, NV, pp 3519–3530
- Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: *Proceedings 5th International Conference Parallel Problem Solving From Nature (PPSN V)*, Amsterdam, pp 292–301
- Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG (2003) Performance assessment of multi-objective optimizers: an analysis and review. *IEEE Trans Evol Comput* 7(1):117–132

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.