Energy-oriented scheduling based on Evolutionary Algorithms

Markus Rager, Christian Gahm, Florian Denz

Angaben zur Veröffentlichung / Publication details:

Rager, Markus, Christian Gahm, and Florian Denz. 2015. "Energy-oriented scheduling based on Evolutionary Algorithms." *Computers & Operations Research* 54: 218–31. https://doi.org/10.1016/j.cor.2014.05.002.

Energy-oriented scheduling based on Evolutionary Algorithms

Markus Rager [a], Christian Gahm [b,*], Florian Denz [b]

[a] Faculty of Electrical Engineering & Computer Science, University of Applied Sciences Ravensburg-Weingarten, PO Box 1261, D-88241 Weingarten, Germany
[b] Chair of Business Administration, Production & Supply Chain Management, Augsburg University, D-86135 Augsburg, Germany

Keywords:
Parallel machine scheduling
Energy efficiency
Resource leveling
Genetic Algorithm
Memetic Algorithm

A B S T R A C T

Energy efficiency has become more and more critical for the success of manufacturing companies because of rising energy prices and increasing public perception of environmentally conscious operations. One way to increase energy efficiency in production is to explicitly consider energy consumption during short-term production planning. In many cases, final energy sources (FES) are not directly consumed by production resources and thus have to be transformed by conversion units into applied energy sources (AES), such as steam or pressure, so the relationship between AES and FES has to be considered. Therefore, we present an energy-oriented scheduling approach for a parallel machine environment. These parallel machines require production order and process time specific amounts for AES and the objective is to minimize the demand of FES. This minimization can be achieved by smoothing the cumulated demand of AES to avoid the frequent load alternations that are responsible for the inefficient operation of conversion units. Therefore, resource leveling is used as a surrogate objective for optimization. To solve the resource leveling problem for large problems, a Genetic Algorithm and two Memetic Algorithms are developed. The evaluation of the proposed Evolutionary Algorithms is based on small test instances and several real-world instances. These latter instances are based on an application case from the textile industry, and promising results concerning energy costs and carbon dioxide emissions are reported.

1. Introduction

An analysis of recent trends in manufacturing industries shows that energy efficiency is one of the most important challenges that companies face, which is due to the economic fact of rising energy costs (e.g., for electricity, natural gas, or fuel oil). Managing these costs is crucial – especially in energy-intensive industries such as chemicals, textiles, or food – for corporate success (e.g., [1–3]). An additional reason for manufacturing companies to address energy efficiency is the increased public perception of business operations' ecological impact (cf. [1–3]). With regard to the manufacturing of goods, environmentally conscious operations depend, on the one hand, on an efficient use of resources (e.g., energy or raw materials) and, on the other hand, on minimizing the environmental impact (e.g., carbon dioxide or greenhouse gas emissions in general) during the entire product life cycle. For these reasons, it is necessary to use energy in a more efficient way to stay (or become) competitive.

Energy use during production, and thus energy efficiency, can be influenced by the following two types of measures [4]: technological and organizational. Technological measures focus on efficiency improvements through technical innovations (e.g., new machines or the production process [2]). The main drawbacks of these technological measures are the high costs of development and/or investment. Here, we focus on the organizational measures used to improve energy efficiency, particularly on short-term production planning, i.e., energy-oriented scheduling.

Because (machine) scheduling is the final planning task regarding production processes within a hierarchical supply chain planning system (cf. [5]), the preceding planning tasks (e.g., master planning) determine the decision limits and restrictions (e.g., the available capacity) for energy-oriented scheduling. For that reason, the presented energy-oriented scheduling approach is subject to these limits and restrictions, as is the underlying production system. In terms of the production system, we investigate identical parallel machines, which for example can be found in the energy-intensive textile industry (cf. [4]). We further assume a given order portfolio consisting of production orders that have individual processing times and energy requirements and that have to be processed without interruptions (i.e., preemption is not allowed). Based on this order portfolio and a predetermined (fixed) planning horizon, the preceding planning task determines the number of

* Corresponding author. Tel.: +49 821 5984359; fax: +49 821 5984353.
E-mail addresses: ragerma@hs-weingarten.de (M. Rager), christian.gahm@wiwi.uni-augsburg.de (C. Gahm), florian.denz@wiwi.uni-augsburg.de (F. Denz).

parallel machines and ensures the feasibility of the temporal and energy conditions. Consequently, the basic scheduling task considered in this contribution is the allocation and sequencing of production orders on identical parallel machines within a fixed planning horizon.

The structure of the paper is as follows. To be able to schedule for energy efficiency, a detailed knowledge of the energy demand and supply characteristics in production environments is indispensable and is therefore analyzed in the following section. Afterward, the surrogate objective of resource leveling is introduced to achieve the desired energy efficiency, and a literature review concerning energy-oriented scheduling and resource leveling is presented. A binary program of the planning problem at hand is formulated in Section 3. The developed solution methods, a Genetic Algorithm and two Memetic Algorithms, are described in Section 4. An evaluation of these solution methods with small academic test instances and a real-world application case is presented in Section 5. The last section provides a brief conclusion and describes future research topics.

2. Energy-oriented scheduling

2.1. Energy demand and supply in production systems

To consider energy efficiency during scheduling in a suitable manner, an analysis regarding the characteristics of energy supply and the energy demand of a production system has to be carried out. This production-oriented analysis reveals two major cases for the provisioning of energy in production systems [4]. Either there is a direct use of final energy sources (FESs) in the production process (cf. case 1 in Fig. 1) or there is the need to convert FESs before their designated application in a production process (cf. case 2 in Fig. 1). An example of the former case is a production system (PS) consisting of machining production units (PUs) that are solely run by electricity. The latter case is accomplished by so-called "conversion units" (CUs). Their duty is to provide the necessary applied energy sources (AESs), such as steam or pressure, to run PUs (PU_1, …, PU_m). A CU centrally provides AESs for several PUs. This case usually occurs when thermal energy is needed in any form to run the production process. Therefore, fossil energy sources are generally burned in CUs centrally to provide heat, which, in turn, ensures the right process conditions needed in industrial chemical processes, for example.

In summary, the initial situation for all considerations in the article at hand is a production environment in which energy is provided centrally. In addition, the energy supply system is built in such a way that AES is produced on demand, i.e., there is no possibility for recirculation or the like.

The consideration of energy is challenging because – in contrast to traditional scheduling approaches – the impact of an allocation is not direct or obvious, and thus, an in-depth analysis of the conversion unit operation's behavior is necessary in the first step. The situation is even more complicated because the energy supply system (including the CU) is often separated from the production system in both a spatial and an organizational context. Because of that separation, there is no adequate information exchange and no coordination of both systems takes place in most cases. In consequence, the CU has to fulfill the cumulated and uncoordinated demand of AES (AESD) of all production units. This often leads to inefficient operation of the CU because the production system only considers the maximum energy supply from the CU as a restriction during scheduling, if it is considered at all. Therefore, scheduling for energy efficiency in production has to consider the cumulated AESD and the economic and ecological effects of this demand on the operation of the CU. This is an additional requirement apart from the traditional task of scheduling that leads to "energy-oriented scheduling".

In general, scheduling seeks an efficient allocation and sequencing of production orders to production units (here, one of the parallel machines) with regard to a certain performance measure (e.g., makespan, maximum tardiness, etc.). To be able to generate an energy-oriented schedule, a new point of view has to be introduced first; production orders have to be distinguished with regard to their AESD, and thus, each order is specified by an individual AESD profile. The demand can either be constant or can vary over time. In contrast to the former case, which is quite easy to model, the latter case requires a more sophisticated approach, which is why production orders are artificially divided up into operations whereby each operation has a certain constant AESD (cf. [4]). In return, the non-preemptive execution of a production order has to be assured by time lags of zero between its operations (i.e., the operations have to be linked by the no-wait constraint; cf. [6,7]). This modeling approach offers the possibility of accounting for the energy characteristics of any type of order (e.g., start-up

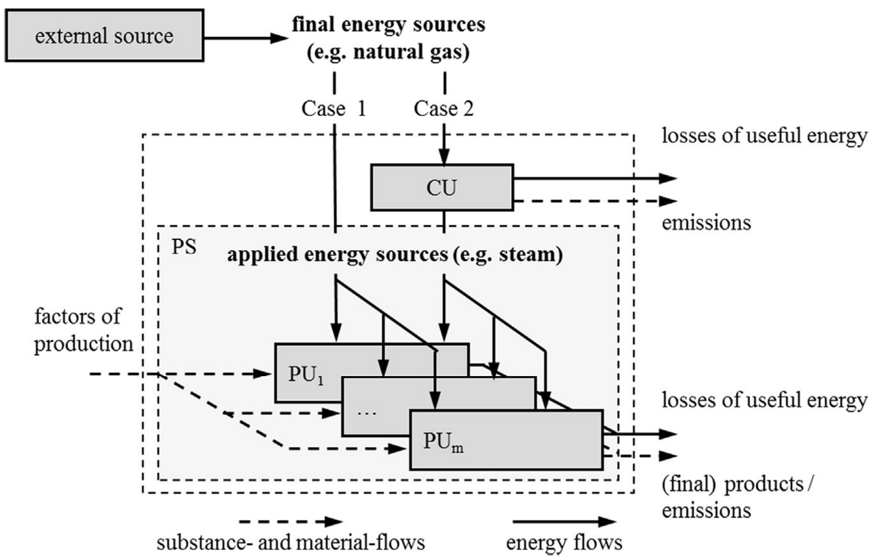

Fig. 1. Energy supply for production systems (following [4]).

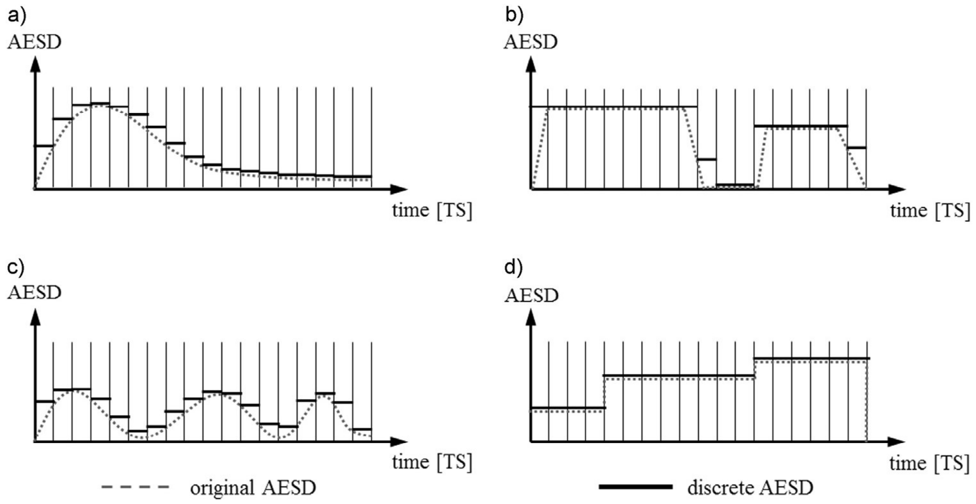

Fig. 2. Examples of AESD profile discretization.

phases, strongly varying demands, etc.). Obviously, this discretization of the energy demand has an inherent loss of accuracy, but it avoids the modeling of complex functions and thus reduces the problem's complexity (a similar approach is used in [8]). Fig. 2 illustrates this discretization based on a predefined time-slice model that represents a required level of detail (the temporal resolution) that depends on the specific scheduling problem. Each time slice (TS) aggregates a certain number of time units (e.g., 10 min to one TS). The number of operations per job varies significantly with regard to the level of detail and the original AESD profile (cf. Fig. 2a: 18 operations, Fig. 2b: 5 operations, Fig. 2c: 18 operations and Fig. 2d: 3 operations).

The proposed modeling approach therefore leads to a certain number of operations o_{ij} per production order J_j that have an individual constant demand for AES r_{ij} per time slice and a processing time p_{ij} that defines the number of time slices the operation requires on one of the parallel machines. R_{ij} denotes the total amount of AESD per operation ($R_{ij} = p_{ij} r_{ij}$).

In any production system in which production orders are executed concurrently on more than one production unit and demand the same type of AES, such as the parallel machine environment considered in this contribution, the individual AES demands accumulate per time slice. The lower part of Fig. 3 illustrates the allocation and sequencing of the two production orders whose individual AES demands lead to the cumulated AESD (cum. AESD) shown in the upper part of Fig. 3.

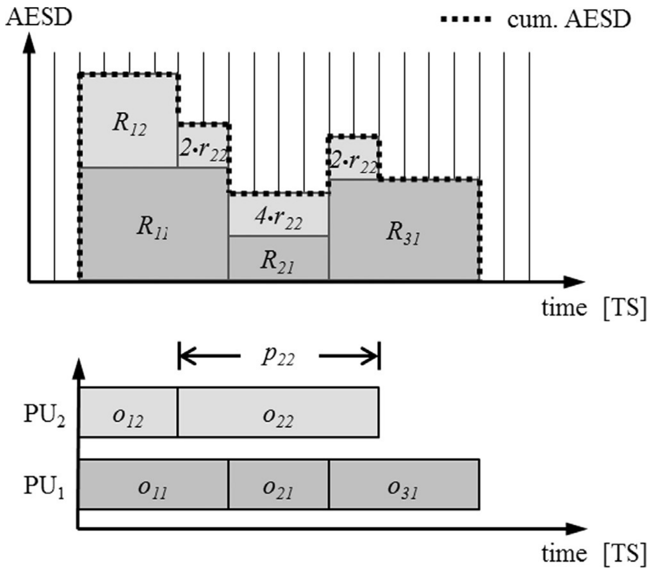

Fig. 3. Cumulated AES demands.

Here, it can be observed that the temporal overlapping of the individual AES demands result in a strongly varying cumulated AESD. The consequences of these short-term variations, economic and ecologic, strongly depend on the type and technical characteristics of the CU. This effect can be shown by the example of a steam boiler CU that converts FES (e.g., natural gas or fuel oil) and water into AES in the form of steam (detailed information about steam boiler technology, characteristics, and operation efficiency can be found, e.g., in [9–11]). Three general operating situations involving steam boilers can be distinguished. In the first situation, the cumulated AESD falls below a base load limit and additional throttling is no longer possible. The boiler then has to be shut down, the flue gas has to be aerated, and the steam boiler has to be restarted. This situation is obviously inefficient and causes additional emissions and losses of useful energy. In the second situation, the required amount of AES reaches peak loads. If a certain upper load limit is exceeded, either shutting down the boiler for technical security reasons (with the consequences explained above) or starting up additional peak-load-boilers will be unavoidable. Both scenarios would cause high additional losses of useful energy and emissions, particularly as peak-load boilers have to be held in warm redundancy. In the third situation, which is most common and therefore the most important, the cumulated AESD varies between the base load limit and the upper load limit. These variations are undesirable for two reasons; on the one hand, there is an enormous reduction in efficiency if a CU operates considerably far from its optimal efficiency level. On the other hand, steam boilers do not operate efficiently during the heating-up and cooling phases. These inefficient phases, caused by a strongly varying cumulated AESD, result in significant losses of useful energy and high emissions. The degree of inefficiency depends on the volume of the AESD increase (or decrease) and the technical characteristics of the CU, but the consequences of varying demands are generally observable. Therefore, the smooth operation of CUs bears significant optimization potential for energy efficiency in production processes.

Following this analysis, an adaption or extension of the classical scheduling problem by constraints and/or objective functions has to be made to improve energy efficiency in production by energy-oriented scheduling.

2.2. *Energy efficiency – achieved by resource leveling*

Based on the previous analysis, we formulate a surrogate objective function that addresses case 2 of Fig. 1. Because the total AES demand is defined by the AES demand of all jobs and their operations, it is not possible to influence this total demand. The only potential for optimization lies in the improvement of the provisioning of the AES by CUs. Because of this fact and because AES is provided by conversion units that convert FES to AES, energy-oriented optimization can only be achieved by efficient operation of the CU.

The main challenge here is the suitable modeling of the effects of strongly varying cumulated AESD. One problem is that it is not possible to exactly calculate the FESD based on the given AESD because the conversion process is not deterministic for technical reasons. Accordingly, it is not possible to formulate a suitable objective function based on the FESD. In consequence, a surrogate objective function reflecting the effects on costs and emissions based on the cumulated AESD is required.

The formulation of the surrogate objective has to account for the following two issues: the analysis in Section 2.1 has shown that CUs lose efficiency disproportionately the further away their operating point is from the optimal efficiency level and, because CUs do not operate efficiently during adjustment phases, the objective function also has to take load alternations into account. These load alternations can be distinguished with regard to the volume of increase or decrease and the number of times they occur. The volume of the alternation is mainly responsible for losses of useful energy (independent of the direction of the alternation). For these reasons, we identified the minimization of the deviations of the cumulated AESD from a desired level as an appropriate surrogate objective (cf. also [4]). Problems with such an objective are referred to as resource leveling problems in the literature and have been extensively studied by several authors (e.g., [12–14]).

The technical benefit of a leveled cumulated AESD is the smooth and even operation of the CU, without frequent and voluminous load alternations. From an organizational point of view, the resource leveling approach is advantageous because of its applicability to any type of CU. If the optimal efficiency level of the CU is known, it can be used as the desired level for the optimization; otherwise, the desired level can be easily derived from the theoretical optimum. Fig. 4 illustrates the influence of leveling the cumulated AESD on the estimated FESD.

In the left part of Fig. 4, a potential result of planning under the objective “minimize makespan” (chosen as a representative example of a time-related objective) is shown. The right part shows the impact of applying the resource leveling objective. In addition to the obvious improvements regarding the estimated FESD, and thus the CU's overall mode of operation, another effect of the leveling procedure is that the improvements can come at the expense of the objective to minimize the makespan. Nevertheless, the limit given by the total length of the planning horizon offers the potential to improve energy efficiency. Of course, this procedure is only advantageous for energy efficiency. As soon as the decision maker is interested in time-related objectives (e.g., minimize makespan) and also energy-efficiency, whether they are in conflict has to be investigated (cf. Fig. 4). If this is the case, the conflict has to be resolved, for example by changing one objective into a satisficing objective, in which an aspiration level is formulated as a constraint, or by using a multi-objective solution approach. For example, the latter is used in Gahm et al. [15], in which order lead-times and resource levels are simultaneously optimized, or in Mouzon et al. [16], in which energy consumption and total completion time are considered in a multi-objective mixed-integer linear program. However, in this study, we assume the other case and focus on single objective resource leveling to exploit the potential for improving energy efficiency.

2.3. *Literature review*

This section provides a literature review consisting of two parts: first, contributions regarding energy aspects within production planning are analyzed, and second, approaches addressing resource leveling are analyzed.

As stated in Section 1, there is an increasing awareness of sustainable and energy-efficient operations by managers and researchers. Energy-intensive industries in particular need tools and methods to optimize production processes regarding energy efficiency [17]. The need for research to integrate energy orientation in operations management, especially in short-term production planning, has been stated by a broad range of authors (e.g., [2,3,18,19]). Energy orientation in short-term planning is not a new research area, but in most cases, it is dedicated to specific problems in certain industry sectors or to utility systems, such as cogeneration systems (cf. [20] or [21] and [18] or [22], respectively)

To the best of our knowledge, energy aspects in terms of production scheduling were first addressed by Subaï et al. [23]. Instead of exclusively maximizing throughput, as is done in origin hoist scheduling problems, the authors expand the objective function by additional cost factors for energy (among others). Another

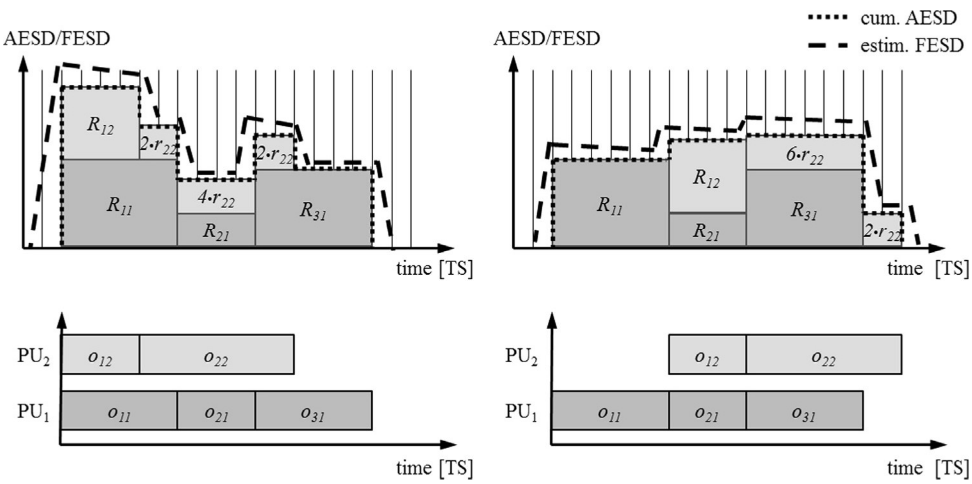

Fig. 4. Influence of the resource leveling objective on the estimated FESD.

report considering a scheduling problem from an energy-oriented point of view is derived by Mouzon et al. [16]. The authors investigate energy-saving potentials by switching machines from idle to offline mode if they are not bottleneck resources. This work has been the topic of further research in a subsequent article by Mouzon and Yildirim [24], which focuses on a single machine scheduling problem. A newer contribution by Nolde and Morari [25] presents a production scheduling approach for a steel plant. In this article, a schedule (and the resulting load curve) is planned a day in advance and sent to the electricity provider. This provider in turn adjusts the curve (e.g., to avoid peaks during energy intensive parts of the day), and the steel plant has to reschedule because any gaps between the updated and actual load curve results in penalty costs. Therefore, the objective is to minimize penalty costs by job scheduling. Fang et al. [2] develop a multi-objective mixed-integer programming model for a flow shop environment to minimize makespan, peak power loads, and the carbon footprint. Therefore, a decision has to be made regarding processing speed (among other considerations), which influences the competitive objectives (e.g., if the processing speed is increased, the makespan shortens at the expense of a higher power demand and vice versa). Because computational effort tends to be quite high, the authors conclude that there is a need to improve the developed solution method to make it applicable to real-world problems.

Insights into the impacts of the task-oriented consideration of energy consumption in a machining environment are provided by He et al. [26]. The authors develop a method to model energy characteristics through discrete event graphs. Furthermore, a method of investigating energetic behavior through simulation analysis is derived.

Artigues et al. [8] present an energy-scheduling problem, which is formulated as a generalization of the cumulative scheduling problem (which itself is an extension of the well-known parallel machine scheduling problem). The main characteristic of the derived problem formulation is the possibility to adjust an order's energy demand. Thus (similar to the processing speed adaptation of [2]), the completion of production orders can be accelerated at the expense of higher power consumption. The maximum available power is restricted to a certain level, and overruns are penalized. The authors propose a two-step constraint programming/MILP approach to determine a valid schedule with regard to the power limit. The production environment considered by Artigues et al. [8] is very similar to the one considered in this contribution, but the planning decisions and the objective are different.

The only planning approach that considers energy efficiency in a suitable manner concerning the planning problem at hand is the approach of [4]. In his doctoral thesis, the author presents fundamental aspects of energy-oriented scheduling that also form the basis of the article at hand. Here, we present a modified version of the planning problem and an enhanced solution method to achieve energy efficiency through resource leveling.

The objective of resource leveling originated in the field of project planning and scheduling. Comprehensive overviews can be found in Younis and Saad [27], Neumann and Zimmermann [13,28], Caramia and Dell'Olmo [29], Anagnostopoulos and Koulinas [30] and Gather et al. [14]. A small selection of current contributions is analyzed in the following paragraphs.

Drótos and Kis [31] describe a resource leveling problem in a general machine environment without precedence relationships between tasks and solve it by branch-and-bound. Here, up to 20 machines with up to 20 tasks are considered. A branch-and-bound-based solution method is also presented by Gather et al. [14]. The authors use a new tree-based enumeration scheme, a constructive lower bound, and preprocessing techniques and solve problem instances with up to 20 activities to optimality. Rieck et al. [32] propose a new mixed-integer linear programming formulation for the resource leveling problem and solve problems with 50 activities and tight deadlines to optimality for the first time. Analyzing these contributions to resource leveling, it must be concluded that none of the proposed exact solution methods are able to solve larger problem instances (with up to 1000 operations) within a reasonable amount of time.

A basic decision concerning resource leveling is the determination of a criterion for measuring the leveling that is adequate for the problem (cf. [13,33]). The most common objective criterion for resource leveling in the literature is the minimization of the sum of the squared deviations of the resource utilization from an optimal value over time. This criterion explicitly penalizes large deviations and therefore is most appropriate for the problem at hand.

Because none of the above-mentioned exact solution methods are able to solve problem sizes addressed in this contribution and due to the complexity of the problem (which can be assessed on the grounds that the general resource leveling problem with precedence constraints is NP-hard in the ordinary sense [13] and that even the single-machine resource leveling problem is NP-complete in the strong sense [31]), the use of meta-heuristics as solution method is indispensable ([30,33–35] also come to this conclusion).

Rager [4] describes the only meta-heuristics that adequately address the defined energy-oriented scheduling problem by considering the no-wait constraint: a Genetic Algorithm and a Memetic Algorithm. Therefore, we present and evaluate these meta-heuristics and compare their performance to a new variant of a Memetic Algorithm (MA; e.g., [36,37]).

3. Formulation of a binary program for energy-oriented scheduling

To substantiate the energy-oriented scheduling problem considered in this contribution, the following non-linear binary program is used (cf. [4]). As described above, the basic scheduling task is the allocation and sequencing of a set of production orders $J=\{J_j|j=1,\dots,n\in\mathbb{N}\}$ on a set of identical parallel machines $M=\{M_k|k=1,\dots,m\in\mathbb{N}\}$. According to Section 2.1, each production order J_j consists of a sequence of operations $O_j=\{o_{ij}|i=1,\dots,n_j\in\mathbb{N}\}$, whereby each operation o_{ij} has a processing time of p_{ij} time slices and requires a constant amount r_{ij} of AES per time slice. After the start of production order J_j, its operations have to be processed in the right order $o_{1j}\rightarrow o_{2j}\rightarrow\dots\rightarrow o_{n_jj}$ without interruptions (a consequence of the non-preemptive and the no-wait constraints) on the same machine M_k. Consequently, the overall duration (p_j) of production order J_j can be calculated by $\sum_{i=1}^{n_j}p_{ij}$. Without loss of generality, the planning horizon defined by $T>1$ ($t\in\{0,\dots,T\}$) determines a common release date rel ($rel=1$) and a common deadline $\tilde{d}$ ($\tilde{d}=T-1$) that are used for modeling purposes. Each machine is only capable of processing one production order at once at any given time point $t\in\{0,\dots,T\}$, and each production order may only be processed by one machine at any given time point t. A machine schedule S can therefore be expressed as the following function $S:j\mapsto(M_j^S,S_j^S)$, in which $M_j^S\in\{1,\dots,m\}$ denotes the machine to process production order J_j, and $S_j^S\in\{1,\dots,\tilde{d}\}$ denotes its starting time.

With regard to the selected objective criterion for resource leveling – the minimization of the sum of the squared deviations of the resource utilization from a desired level – the resulting schedule depends on the desired level $\hat{r}$. In certain cases, there can be a significant influence of this level on the schedule ([33] differentiates here in the context of the considered time interval between full range, dynamic range, and effective range values), but in this contribution, the theoretical optimum, the average resource utilization, is appropriate (cf. Section 2.2).

$$\hat{r} = \frac{1}{\tilde{d}} \sum_{j=1}^{n} \sum_{i=1}^{n_j} r_{ij} \quad (1)$$

These preliminary definitions lead to the following binary program:

Indices

$j = 1, ..., n \in \mathbb{N}$ index for production orders
$i = 1, ..., n_j \in \mathbb{N}$ index for operations (of production order J_j)
$k = 1, ..., m \in \mathbb{N}$ index for machines
$t = 1, ..., T \in \mathbb{N}$ index for time slices ($T > 1$: length of planning period)

Parameters

p_{ij} processing time of operation o_{ij} (of production order J_j)
r_{ij} AESD of operation o_{ij} (of production order J_j) per time slice
$\hat{r}$ desired level
$\tilde{d}$ common deadline ($\tilde{d} = T - 1$)

Variables

x_{ijkt} binary variable indicating if operation o_{ij} is processed on machine k at time slice t
X_{ijt} binary variable indicating if operation o_{ij} is completed at the end of time slice t

Objective function

$$Min \sum_{t=1}^{\tilde{d}} \left(\sum_{j=1}^{n} \sum_{i=1}^{n_j} \sum_{k=1}^{m} x_{ijkt} r_{ij} \right) - \hat{r} \Bigg)^2 \quad (2)$$

Constraints

$$\sum_{t=1}^{\tilde{d}} X_{ijt} = 1 \qquad \forall i, j \quad (3)$$

$$(x_{ijkt} - x_{ijkt+1}) \le X_{ijt} \qquad \forall i, j, k, t = 1, ..., \tilde{d} \quad (4)$$

$$\sum_{j=1}^{n} \sum_{i=1}^{n_j} x_{ijkt} \le 1 \qquad \forall k, t \quad (5)$$

$$\sum_{k=1}^{m} \sum_{t=1}^{\tilde{d}} x_{ijkt} = p_{ij} \qquad \forall i, j \quad (6)$$

$$p_{ij} x_{i'jkt} \le \sum_{\tilde{t}=1}^{t-1} x_{ijk\tilde{t}} \qquad \forall i < i' \le n_j, j, k, t > 0 \quad (7)$$

$$X_{ijt} = X_{(i-1)j(t-p_{ij})} \qquad \forall i = 2, ..., n_j, j, t > p_{ij} \quad (8)$$

$$x_{ijkT} = 0 \qquad \forall i, j, k \quad (9)$$

$$x_{ijkt}, X_{ijt} \in \{0, 1\} \qquad \forall i, j, k, t \quad (10)$$

The resource leveling objective is represented by the non-linear objective function (2) and minimizes the sum of the squared deviations of the resource utilization from the optimal value $\hat{r}$ (1). Constraints (3) and (4) define the completion time and prevent the preemption of operations. Eq. (5) assures that not more than one production order is allocated to a machine at any time point. Eq. (6) guarantees the complete processing of each operation, and Eq. (7) guarantees the compliance of the processing sequence of all operations of a production order on the same machine. The no-wait constraint is modeled by Eq. (8). Eq. (9) prevents the allocation of operations at the last time slice of the planning horizon. Eq. (10) limits the domain of the decision variables.

4. Solution method

Following the results of the literature review presented in Section 2.3, the energy-oriented scheduling problem considered in this study has to be solved by meta-heuristics because of its complexity and the praxis-relevant problem instances that have to be solved with regard to solution quality and computing time. The solution method has to make two decisions to calculate a schedule, namely, the allocation of production orders to machines and the determination of starting times for each production order. The sequence of orders can be seen as a by-product of the determination of the start time. The suggested solution approach decomposes the problem into two sequentially performed steps, one for each of these decisions (as proposed e.g. in [38]). In the first step, production orders are allocated to machines, and the objective is to distribute the workload as evenly as possible among machines to facilitate the following leveling of the cumulated AESD. In the second step, the allocation of the production orders remains unchanged, but the production orders' starting times are determined. This step focuses the objective of resource leveling. It has to be noted here that the heuristic solution method determines the starting times for the production orders itself and uses the operations only to calculate the corresponding cumulated AESD per time slice, which implicates the use of the time slice model here.

4.1. Opening procedure – allocation of production orders to machines

The general purpose of the first step, which can also be seen as an opening procedure (and is therefore called INIT), is to distribute the workload (defined by the processing times p_j of the production orders) as equally as possible among the parallel machines (cf. [4]). This is done to increase the flexibility for the determination of the starting time of the succeeding second step (which can be seen as an improvement procedure).

Because the objective of equal workload distribution in an identical parallel machine environment is similar to the objective of makespan minimization, the former can be replaced. This replacement leads to a well-known dual scheduling problem, the "bin packing problem" (cf. [39,40]), and established solution methods can be applied (overviews can be found in [41,40]). Because several authors (e.g., [42–44]) have shown that the "largest processing time first" rule performs well in solving this problem and because the resulting schedule will only be used as an initial solution, this priority rule is used to allocate production orders to machines and to distribute the workload equally. Here, production orders are first sorted in descending order according to their processing time p_j and then allocated to the machine with the lowest workload. The result (S^{INIT}) of this planning step is the allocation (indicated by a_{jk}) of production orders J_j to machines $k = 1, ..., m$ ($a_{jk} = 1$ if J_j is allocated to machine k, $a_{jk} = 0$ otherwise) and the corresponding schedule that contains the workload wl_k of each machine: $wl_k = \sum_{j=1}^{n} \sum_{i=1}^{n_j} p_{ij} a_{jk}$.

4.2. *Improvement procedure – determination of starting times*

Based on the discussion of solution methods for resource leveling problems presented in Section 2.3, we apply different Evolutionary Algorithms (EAs; e.g., [37,45]) to solve the described energy-oriented scheduling problem: the standard Genetic Algorithm (abbreviated GA in the following) and the MA with a peak-oriented local search mechanism (abbreviated MA-PLS) as proposed by Rager [4] and a new MA with a more sophisticated local search mechanism based on deviations (abbreviated MA-DLS). These EAs are used as there is no general advantageousness of any other meta-heuristics such as Tabu Search, Simulated Annealing, or Variable Neighborhood Search (cf. [46] for an overview of these heuristics) and as such meta-heuristics would require very sophisticated transformation rules (e.g., for swapping or shifting several production orders simultaneously). Another aspect is, that, if adequately designed, EAs provide the possibility of maintaining certain (promising) parts of a solution and changing only other (less promising) parts and, consequently, are able to generate and store information about promising solutions or solution parts. Moreover, the application of EAs and particular GAs is seen to be suitable if the search space is large and/or not well understood (cf. [47]).

To solve a particular problem using EAs, several components must be specified (cf. [37,48]; general and detailed information on EAs and their subclasses, procedures, etc. can be found in [46,47,49–52] and the references presented there):

(i) the representation of a solution by an individual – here of a schedule S;
(ii) the evaluation or fitness function – here the objective function (2);
(iii) a procedure for selecting parents and individuals for the next generation and the number of chromosomes in a generation (population size μ);
(iv) a procedure for recombination and its probability (Θ_r);
(v) a procedure for mutation and its probability (Θ_m);
(vi) a procedure to generate an initial population; and
(vii) one or more termination criteria.

The first decision to be made is the representation of the schedule. In this context, a solution is called a "phenotype", and the encoded representation is called the "genotype" or "chromosome". The elements of the string that represents a genotype are called "genes", and the values those genes can take are called "alleles". The encoding and decoding of a solution, also designated as "genotype–phenotype mapping", is required, on the one hand, to be able to use evolutionary procedures (e.g., parent selection, recombination, and mutation) and, on the other hand, to evaluate the solution using the corresponding fitness function. The type of genotype representation used for the EAs presented in this paper is called permutation representation (cf. [37]) and the sequence of orders also determines the starting time. The applied string representation of the genotype is founded on the time-slice model used to discretize the energy demand (cf. Section 2.1) and the schedule definition function $S: j \mapsto (M_j^S, S_j^S)$ that is based on the machine allocation ($M_j^S \in \{1, \ldots, m\}$) and the starting times of the production orders ($S_j^S \in \{1, \ldots, \tilde{d}\}$). In doing so, the remaining idle time id_k of a machine k, which is the difference in the length of the planning horizon and the workload wl_k (determined by the first optimization step), is divided into a set of dummy orders $D = \{D_{d,k} | d = 1, \ldots, x_k \in \mathbb{N},\ k = 1, \ldots, m \in \mathbb{N}\}$. The amount of dummy orders x_k per machine k depends on id_k and on the underlying time-slice model, where each dummy order has a duration of one time slice. These dummy orders enable the representation of a schedule by a string of genes in which the alleles are either production orders $J_{j,k}$ or dummy orders $D_{d,k}$, where both have dedicated machines k. An exemplary genotype–phenotype mapping is illustrated in Fig. 5.

With this permutation representation, the shifting of a gene would either change the sequence of orders on a machine (e.g., swapping gene 1 and gene 3) or not (e.g., swapping genes 1 and 2 or genes 2 and 4). Based on this information, it is possible to maintain certain parts of solutions and to change others. Moreover, this representation provides the advantage that from any permutation, along with the fixed machine allocation, an unambiguous and feasible schedule can be determined, and it is possible to use objective function (2) for evaluating a solution (cf. (ii)).

The use of a permutation representation for the problem at hand permits the selection of procedures and parameters (cf. (iii)–(vii)) described and used in the literature (see above). In this context, interactions between the procedures and their parameters that can hardly be estimated often exist. In addition, there is a lack of theoretical investigation of these interactions in the literature. A survey of contributions with regard to the theoretical behavior of EAs can be found in Kallel et al. [53]. These contributions often refer to single procedures and parameters for specific problems (e.g., mutation rates for problems with special objective functions) and thus cannot be generalized. For this reason, we investigated several well-known procedures that are adequate for the problem at hand (i.e., maintaining certain parts of a solution) for the recombination (e.g., partially mapped crossover and order crossover; cf. [54,37]) and mutation (e.g., swap mutation, insert mutation, scramble mutation, and inversion mutation; cf. [55,37]) of individuals. The choice of procedures and parameters should be made with regard to the balance of exploration (or search diversification) and exploitation (or search intensification).

Before describing the selected procedures and parameters, the main course of action of the applied EAs is outlined in the following pseudo code (following [56]):

```
GA/MA-PLS/MA-DLS (S^INIT, μ, Θ_r, Θ_m);

c := 0; t := now();
P_c := initializePopulation(S^INIT, μ);                    // (vi)
F_c := evaluatePopulation(P_c);                            // (ii)
do
    c := c+1:
    // iii. (two parents for two offspring)
    Par_c := selectParents(P_{c−1}, μ)
    Rec_c := recombineParents(Par_c, Θ_r,"PMX");           // (iv)
    Mut_c := mutateIndividuals(Rec_c, Θ_m, "INV");         // (v)
    // MA enhancement
    LS_c := performLocalSearch(Mut_c);
    F_c := evaluatePopulation(LS_c or Mut_c);              // (ii)
    P_c := selectSurvivors(P_{c−1}, F_c, μ);               // (iii)
loop while t <= t^Max;                                     // (vii)
```

The function to initialize the start population (P_0) randomly generates production and dummy order sequences (*initializePopulation*). This is done based on the allocation of production orders to machines from the opening procedure (S^{INIT}, cf. Section 4.1) and the population size (μ). After the evaluation of this population (*evaluatePopulation(P_0)*) with the objective function (2), the selection of parents (*selectParents*; also called mating selection) is executed by a fitness-proportional selection rule (cf. [57]) – scaled by the maximum objective value of all individuals of the corresponding generation (cf. [49]) – based on stochastic universal sampling (SUS; cf. [58]). This procedure keeps high-quality solutions and deepens the search in promising search regions and is also used for the selection of μ individuals (also called survivor selection or replacement; cf. [37]) for the next generation

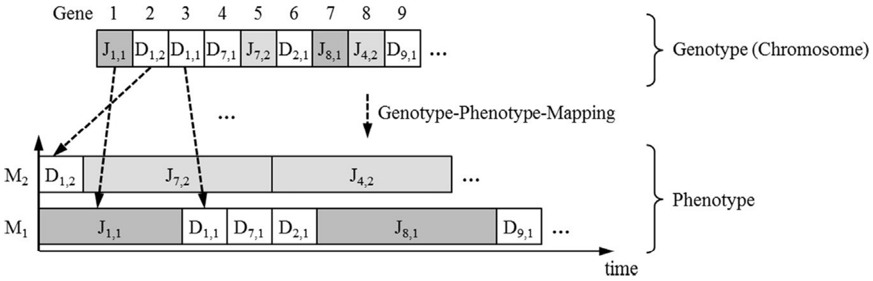

Fig. 5. Genotype–phenotype mapping of the permutation representation [4].

(*selectSurvivors*). Here, the individuals of the previous population (P_{c-1}) and the new individuals (F_c) are considered. The best-known solution from previous generations is always transferred to the next generation. For the recombination of selected parents (*recombineParents*), the partially mapped crossover (PMX) procedure is used because it has shown better performance (concerning the trade-off between computing time and objective value) than the order crossover procedure, although both procedures transfer sequences of genes to their offspring (cf. [37]). Regarding both procedures, Θ_r determines the crossover rate, which is the probability that two offspring are generated by recombination or asexually (copying the parents). The inversion mutation procedure for the mutation of individuals (*mutateIndividuals*), combined with a very low mutation rate Θ_m, has shown the best effects concerning the diversification of the populations. Regarding the EA termination, computing time is used as a criterion (t^{Max}) because the time for scheduling is generally limited in practice, and therefore, this time limit can be set in advance. Furthermore, the performance of the three solution methods, particularly the performance of the MAs, is directly comparable by the objective function value.

The first applied solution method, the GA uses these procedures and parameters. The two MAs enhance this GA through different local search mechanisms (*performLocalSearch*). This hybridization has been proposed by several authors (e.g., [36,37,51]). The basic idea behind the local search procedures developed for the resource leveling problem at hand is to support the GA by the elimination of single peaks (MA-PLS) or the elimination of deviations from the desired resource level in general (MA-DLS). Both local search mechanisms are executed before the evaluation of new individuals takes place (*evaluatePopulation*(LS_c or Mut_c)).

The local search mechanism of the first MA only considers peaks of the cumulated AESD and is therefore called MA-PLS (the PLS represents "peak local search"). This mechanism was introduced by Rager [4]. The first step of MA-PLS is the identification of the maximum peak that will be eliminated or at least softened (*getPeak* in the following pseudo code). This peak is defined by the point in time t^{PEAK} with the highest cumulated AESD. If there is more than one time point with the same maximum peak, the first one in the planning horizon is used (*getFirstPeakTimePoint*). Afterward, the set of all "active" production orders J^{PEAK} that are processed on any machine at time point t^{PEAK} is identified (*getActiveOrders*) as follows:

$$J^{PEAK} = \left\{ J_j \in J | S_j^S \le t^{PEAK} \le S_j^S + \sum_{i=1}^{n_j} p_{ij} \right\}$$

Because the local search mechanism does not change the sequence of orders on a machine but rather changes starting times, the earliest starting time est_i and the latest starting time lst_i of a production order $J_j \in J^{PEAK}$ are used as borders for starting time changes (*calculateTimeWindows*):

$est_j = S_{j'}^S + \sum_{i=1}^{n_{j'}} p_{ij'}$ where j' is the preceding production order on the corresponding machine

$lst_j = S_{j'}^S - \sum_{i=1}^{n_j} p_{ij}$ where j' is the succeeding production order on the corresponding machine

Based on this time window, the PLS first tries to completely shift the currently considered production order to est_j ($\Rightarrow S_j^S = est_j$) or lst_j($\Rightarrow S_j^S = lst_j$) if the peak cannot be reduced. This decision is repeated with all $J_i \in J^{PEAK}$ that have an AESD ($r_{ij} > 0$) at time point t^{PEAK} until the peak is reduced to the optimal value or all possible shifts are evaluated. The following pseudo code illustrates the main course of action of MA-PLS:

MA-PLS (S)

// determine the highest cumulated AESD
  $peak :=$ getPeak(S);
// first occurrence
  t^{PEAK} := getFirstPeakTimePoint($peak$);
// determine all active production orders
  J^{PEAK}:= getActiveOrders(t^{PEAK});
calculateTimeWindows(J^{PEAK});
for $l := 1$ **to** $|J^{PEAK}|$ **do**
    // index of (first) active order
    $x :=$ getNext(J^{PEAK})
    $r_{ix}(t^{PEAK}) :=$ getAESD(J_x, t^{PEAK});
    if $r_{ix}(t^{PEAK}) > 0$ **then**
        if $est_x + \sum_{i=1}^{n_x} p_{ix} < t^{PEAK}$ **then**   // left shift
            $S_x^S := est_x$;
            $peak := peak - r_{ix}(t^{PEAK})$;
            $J^{PEAK} := J^{PEAK} \setminus J_x$;
        elseif $lst_x > t^{PEAK}$ **then**   // right shift
            $S_x^S := lst_x$;
            $peak := peak - r_{ix}(t^{PEAK})$;
            $J^{PEAK} := J^{PEAK} \setminus J_x$;
        endif;
    endif;
    // termination if peak is eliminated
    if $peak <= \hat{r}$ **then** exit for;
next;

The newly developed second variant MA-DLS also does not change the sequence of orders but, in contrast to MA-PLS, considers not only peaks but all deviations from the optimal resource level and is therefore called MA-DLS (the DLS represents "deviation local search"). Thereby, the number of deviations that are iteratively investigated are defined by the parameter no_{ct}^{DEV}, which can be varied during the execution of MA-DLS. This is done to reduce the changes within a solution and to thus support the search intensification in the optimization process.

A second difference between the MAs is the procedure used to eliminate undesired cumulated AESD levels. DLS uses the same time borders as described above, but it iteratively swaps one production order with one dummy order. This is done in both directions as long as dummy orders are available or until the start or the end of the planning horizon is reached. As soon as a swap leads to a reduction of the deviation from the optimal value, the swapping of this production order stops. This is repeated until the deviation is eliminated or all possible swaps are evaluated. Because this detailed swapping procedure is very time consuming, we evaluate swaps only in one direction; the direction is determined with a certain probability based on the position of the deviation in the planning horizon (*getDirectionProbability*). Therefore, this second local search procedure can also be seen as a swap mutation procedure that uses problem-specific information to improve a current solution *S*. The main course of action of MA-DLS is presented in the following pseudo code:

```
MA-DLS (S, t, no_ct^DEV)
// determine the no_ct^DEV highest deviations (absolute values) in
  solution S
dev[] := getAbsoluteDeviations(S, no_ct^DEV)
for f =1 to no_ct^DEV do
      // flag that indicates the swapping direction
      leftOrRight := "right";
     // determine the highest deviation
     dev := getHighestDeviation(dev[]);
     t^DEV := getDeviationTimePoint(dev);
     // determine all active production orders
    ao[] := getActiveOrders(t^DEV);
    calculateTimeWindows(ao[]);
    for l :=1 to |ao[]| do
         // index of (first) active order
         x := getNext(ao[]);
         // machine of the active order
        k := getMachine(J_x);
        leftP=getDirectionProbability(J_x,t^DEV, "left");
        if (getRandomNumber() <= leftP)
             then leftOrRight="left";
        // get set of preceding/succeeding dummy orders on
  machine k
        do[] := getDummyOrders(J_x, k, leftOrRight)
        for g := 1 to |do[]| do
              // returns the resulting deviation
              newDev := swapOrders(J_x, do[g], leftOrRight)
              if (newDev < dev) then
                   acceptSwap(J_x, do[g]); dev := newDev; exit for;
            endif;
     next;
     // termination if deviation is smoothed
     if dev <= 0 then exit for;
  next;
  devS[] := devS[] \ dev;
next;
no_ct^DEV := updateCT(no_ct^DEV, t);
```

The function *updateCT* actualizes the number of considered deviations that are dependent on the actual optimization time *t*. After $t^{update} = t^{\max}/no_{ct}^{DEV}$ time periods (calculated before the optimization starts), no_{ct}^{DEV} is reduced by one.

5. Evaluation

All three developed solution methods are evaluated by a set of small test instances and a set of problem instances that are derived from a real-world application case. To assess the solution quality of the proposed EAs, we compare their objective values with a corresponding reference value (best known objective value, BKOV) calculated by Gurobi 5.5 within the General Algebraic Modeling System (GAMS).

Furthermore, to show the suitability of energy-oriented scheduling for improving energy efficiency in production, not only the results concerning the objective function are presented in the subsequent sections, but the resulting costs and carbon dioxide emission reductions are also estimated.

5.1. Application case

The underlying planning problem of this study originates from a company in the textile industry whose core competency is the refinement of yarns (cf. [4]). The relevant aspects (from an energy-oriented point of view) of the company are its dyeing house (PUs) and its boiler house (CU). All refinement steps are executed with 32 identical dyeing machines, which are heated with process steam centrally provided by the boiler house (consisting of a single three-pass boiler with an economizer). For the purpose of refinement, yarns are put into dyeing machines and brought into contact with different chemicals (i.e., dyes, acids, alkaline or leveling agents). Dyes are extracted by the application of thermal energy provided by the process steam and are then absorbed by the yarn. The filling quantity (fleet size) of a dyeing machine is almost independent of the production order and accounts for 2950 l.

For the presented problem, the AESD of a production order can be identified by the dyeing recipes, which predefine the entire refinement process. Each recipe determines the quantities of input materials (e.g., grey cloth, chemicals, or dyes) and the temporal course of the dyeing process through a time-temperature profile (cf. Fig. 6).

This profile determines the residence times and temperatures for the combination of chemicals and yarns. Thus, all recipes contain heating phases, which require that heating always has to be done at a constant factor of 1 K/min before the dyeing phases begin. Because of this, there is a process-time dependent demand for thermal energy provided by the AES steam. Overall, the individual AES demand profile of a production order is specified by the sequence of phases with varying durations (modeled by operation o_{ij} with a corresponding processing time p_{ij}, cf. Fig. 6), in which there either is a need for process steam during heating phases (marked gray in Fig. 6) or not (during all remaining process steps; note that the temperature during the flat periods in Fig. 6 is kept constant by a heat recovery system). In the application case, a time-slice model with a temporal resolution of five minutes is suitable, and therefore r_{ij} can be calculated as follows:

$$r_{ij} = \Delta Temp/TS\ M\ c = 5\ K/TS\ M\ c \quad [J/TS]$$

In this formula, *M* represents the mass of the fleet size, and *c* represents the specific heat capacity of water.

5.2. Data and parameter settings

To evaluate the solution quality of the proposed solution methods, a set of small test instances (TI) that are solvable within a reasonable computing time by the commercial solver Gurobi is used. These instances vary in the number of production orders n, the number of operations per production order n_j, the planning horizon T, and the number of available machines m (cf. Table 1; processing times can be found in Table A1). Purpose of these instances is to show the suitability of the proposed solution methods in different resource availability settings with comparable relative workloads. Here, the relative workload is calculated by $\sum_{j=1}^{n}\sum_{i=1}^{n_j} p_{ij}/mT$. Accordingly, the number of available machines and the length of the planning horizon are chosen to provide small but sufficient capacity for the investigated quantity of production orders.

For these small test instances, three different types of AESD profiles are specified. The first type contains alternating AESDs and r_{ij} is one for operations with an even index i and zero in all other cases (we indicate this type by an additional 0 and thus, the first instance is fully specified by TI1a0). For the next two types, the AESDs are derived from the processing times of the operations. Here, r_{ij} is either defined by a monotonously increasing (11) or by a monotonously decreasing (12) functional relation and calculated as follows:

$$r_{ij}^{1} = Min(0.8 + p_{ij}\,0.15,\ 1.6) \tag{11}$$

$$r_{ij}^{2} = Max(1.6 - p_{ij}\,0.15,\ 0.8) \tag{12}$$

Consequently, there are 33 small test instances in total (TI1a0, TI1a1, TI1a2, TI1b0, …, TI3e2).

In addition to these small test instances, the developed solution methods are evaluated by different production order scenarios (SC) derived from the aforementioned application case (cf. Section 5.1). Each of these scenarios is defined by the number of production orders (minimum 48 to maximum 128) and the type of these orders. The type specifies the time–temperature profile and thus the AES demand profile. The number of production orders of a certain type is random within a scenario. With respect to these order scenarios, the length of the planning horizon is equal for each of them (one production day with 288 time slices), and the number of available machines is again defined to provide small but sufficient capacity for the investigated quantity of production orders. The number of available machines varies between 12 and 32, and the relative workload for all scenarios is between 92% and 94%. For the application case, the decision on the number of available machines is made by a superior planning step.

For both types of instances, the following parameter settings (cf. Table 2) are used for evaluation and were derived from numerous experiments.

All of the following results concerning the EAs are mean values based on seven experiments per test instance or application case scenario, in which the optimization runtime is limited to 30 s or 30 min, respectively. All experiments are executed on a PC with an Intel Core i7-2600 and 8 GB RAM.

Table 1
Parameters of the small test instances.

	n	n_j	m	T	Relative workloads
TI1 a/b/c	4	4	2	20/24/28	85%/71%/61%
TI2 a/b/c	4	6	2	27/33/39	85%/70%/59%
TI3 a/b/c	6	4	3	20/24/28	85%/71%/61%
TI3 d/e	6	4	4	18/21	71%/61%

5.3. Performance analysis

The numerical results of the solution methods GA, MA-PLS, and MA-DLS concerning the small test instances are summarized in Table 3. The first two columns contain the (best known) objective values (OV^G) and the computing time (CT; in seconds) from Gurobi. Here, an asterisk marks the termination of Gurobi after the specified time limit (10 h) as it was not possible to solve the instance to optimality (in contrast to all other instances not marked by an asterisk). The column INIT presents the objective value (OV^I) achieved with the opening procedure. The following columns contain the mean objective value (OV^7), the coefficient of variation (CV^7), the relative improvement compared with the solution calculated with Gurobi ($Imp.^G$), and the relative improvement compared with the results from the opening procedure ($Imp.^I$) for each of the three EAs (values obtained by EAs are mean values of the seven experiments). The last row in Table 3 (Mean) summarizes the results

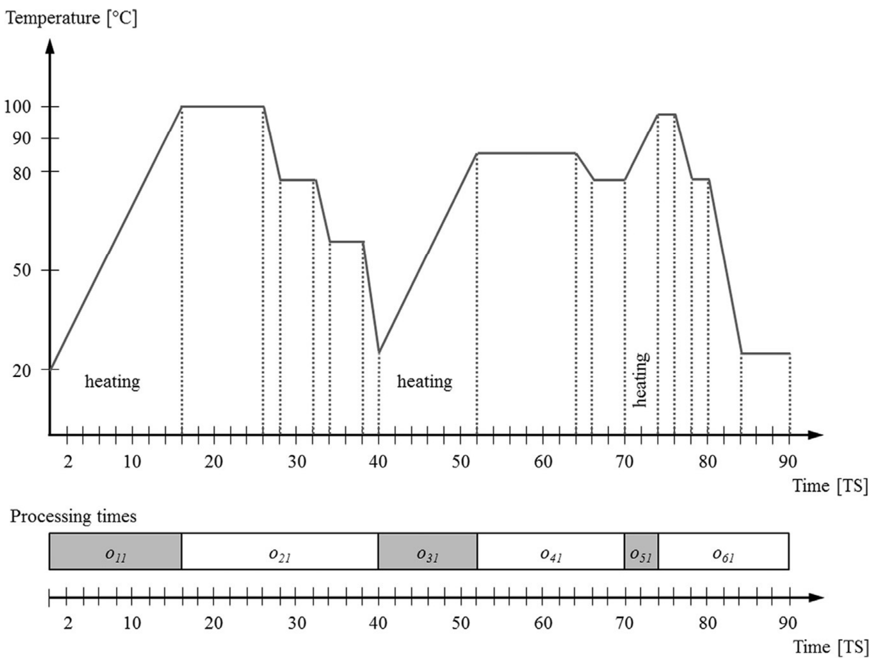

Fig. 6. Example of a time–temperature profile and modeling by operations (following [4]).

by reporting the corresponding mean values (based on all 33 small test instances).

Analyzing these results, it can be stated that all three EAs achieve quite good results concerning their deviation from the BKOV (6.38%, 6.02%, and 5.84% on average) and that all three EAs reach the BKOV for 13 instances. Furthermore, the EAs are able to calculate better objective values for the instances TI3c1, TI3d1, and TI3e1. The relative large deviations of the EAs from the BKOV with regard to certain instances (e.g., TI1a0, TI1c0, or TI3d0) can be traced back to the fixation of orders to machines after step one of the proposed solution methods (cf. Section 4). The results also show, that the MA with the more sophisticated local search mechanism DLS slightly outperforms the others.

To analyze the solution quality of the developed solution methods with regard to the order scenarios derived from the application case, Table 4 presents the achieved mean objective value (OV^7), the coefficient of variation (CV^7), and the mean relative improvement ($Imp.^I$) compared with the results obtained by the opening procedure INIT (all values obtained by EAs are the mean values of the seven experiments). The best objective values per scenario are in bold.

The results show that each of the three EAs achieves significant improvements (compared to the initial solution) concerning the objective of resource leveling by 67.29%, 70.17%, and 71.14% on average. Additionally, the MAs outperform the standard GA in all scenarios except for scenario SC-108. Comparing MA-PLS and MA-DLS, no clear dominating position can be detected. On the one hand, MA-DLS outperforms MA-PLS in terms of the confidence interval that is given by mean objective value and its coefficient of variation. On the other hand, the robustness, measured by the coefficient of variation CV^7, of MA-PLS is better than that of MA-DLS.

Because the effect of the resource leveling is difficult to trace solely based on numerical values, an exemplary comparison (based on SC-124, Experiment 1) between the results of INIT (OV = 3645) and MA-DLS (OV = 1034.71) is made. Fig. 7 shows the resulting course of the cumulated AESD for both solution methods to exemplify the effects of resource leveling.

The two AESD courses clearly show that the defined objective function is able to eliminate large variations in the cumulated

Table 2
Parameter settings of the solution methods.

Solver/solution method	Procedure/parameter	Value
Gurobi	optcr	0.01
	reslim	10 h
GA	Parent selection and survivor selection	Fitness proportional selection by SUS
	Recombination	PMX with probability $\Theta_r = 0.8$
	Mutation	Inversion with probability $\Theta_m = 0.1$
	Population size	$\mu = 300$
	t^{Max}	60 s or 30 min
MA-PLS	–	–
MA-DLS	no_{ct}^{DEV}	4 (initial value)

Table 3
Solution quality of the small test instances.

			Gurobi		INIT	GA				MA-PLS				MA-DLS			
			OV^G	CT [s]	OV^I	OV^7	CV^7	$Imp.^G$ [%]	$Imp.^I$ [%]	OV^7	CV^7	$Imp.^G$ [%]	$Imp.^I$ [%]	OV^7	CV^7	$Imp.^G$ [%]	$Imp.^I$ [%]
TI1	a	0	2.95	30	14.95	4.95	0.00	−67.80	66.89	4.95	0.00	−67.80	66.89	4.95	0.00	−67.80	66.89
		1	3.99	76	14.72	3.99	0.00	0.00	72.87	3.99	0.00	0.00	72.87	3.99	0.00	0.00	72.87
		2	5.40	85	16.16	5.40	0.00	0.00	66.56	5.40	0.00	0.00	66.56	5.40	0.00	0.00	66.56
	b	0	4.63	40	18.63	4.63	0.00	0.00	75.17	4.63	0.00	0.00	75.17	4.63	0.00	0.00	75.17
		1	5.25	366	27.92	6.04	0.00	−15.15	78.35	6.04	0.00	−15.15	78.35	6.04	0.00	−15.15	78.35
		2	6.88	255	30.72	6.88	0.00	0.00	77.61	6.88	0.00	0.00	77.61	6.88	0.00	0.00	77.61
	c	0	5.25	20	21.25	7.25	0.00	−38.10	65.88	7.25	0.00	−38.10	65.88	7.25	0.00	−38.10	65.88
		1	4.67	830	37.35	4.75	0.02	−1.84	87.27	4.75	0.02	−1.84	87.27	4.75	0.02	−1.84	87.27
		2	5.66	606	41.12	5.66	0.00	0.00	86.23	5.66	0.00	0.00	86.23	5.66	0.00	0.00	86.23
TI2	a	0	8.67	392	22.67	8.67	0.00	0.00	61.76	8.67	0.00	0.00	61.76	8.67	0.00	0.00	61.76
		1	5.29	96	19.14	5.29	0.00	0.00	72.37	5.29	0.00	0.00	72.37	5.29	0.00	0.00	72.37
		2	7.69	140	22.69	7.90	0.00	−2.73	65.20	7.90	0.00	−2.73	65.20	7.90	0.00	−2.73	65.20
	b	0	8.73	825	28.73	8.73	0.00	0.00	69.62	8.73	0.00	0.00	69.62	8.73	0.00	0.00	69.62
		1	7.40	25,155	37.84	7.40	0.00	0.00	80.44	7.40	0.00	0.00	80.44	7.40	0.00	0.00	80.44
		2	9.57	6778	45.11	9.78	0.00	−2.19	78.32	9.78	0.00	−2.19	78.32	9.78	0.00	−2.19	78.32
	c	0	8.92	542	32.92	8.92	0.00	0.00	72.90	8.92	0.00	0.00	72.90	8.92	0.00	0.00	72.90
		1	5.96	21,675	50.79	6.27	0.01	−5.10	87.66	6.27	0.01	−5.10	87.66	6.27	0.01	−5.10	87.66
		2	6.24	14,900	60.63	6.59	0.06	−5.63	89.13	6.59	0.06	−5.63	89.13	6.59	0.06	−5.63	89.13
TI3	a	0	6.95	8365	34.95	6.95	0.00	0.00	80.11	6.95	0.00	0.00	80.11	6.95	0.00	0.00	80.11
		1	4.07	4385	33.64	4.75	0.00	−16.58	85.89	4.75	0.00	−16.58	85.89	4.75	0.00	−16.58	85.89
		2	5.22	19,826	36.55	6.00	0.02	−15.07	83.57	5.95	0.00	−14.09	83.72	5.95	0.00	−14.09	83.72
	b	0	6.96	34,136	42.96	6.96	0.00	0.00	83.80	6.96	0.00	0.00	83.80	6.96	0.00	0.00	83.80
		1	2.50	*	63.49	2.92	0.07	−16.99	95.40	2.99	0.08	−19.66	95.29	2.99	0.08	−19.66	95.29
		2	3.22	*	69.16	3.58	0.04	−11.36	94.82	3.58	0.06	−11.23	94.83	3.58	0.06	−11.23	94.83
	c	0	4.68	*	48.68	6.68	0.00	−42.75	86.28	5.82	0.17	−24.43	88.04	5.54	0.18	−18.32	88.63
		1	5.02	*	84.81	4.64	0.00	7.42	94.53	4.64	0.00	7.42	94.53	4.64	0.00	7.42	94.53
		2	4.80	*	92.45	5.36	0.06	−11.64	94.20	5.44	0.07	−13.29	94.12	5.44	0.07	−13.29	94.12
	d	0	5.61	6540	35.61	7.61	0.00	−35.64	78.63	7.61	0.00	−35.64	78.63	7.61	0.00	−35.64	78.63
		1	4.15	*	49.92	2.74	0.05	34.01	94.51	2.74	0.05	34.01	94.51	2.74	0.05	34.01	94.51
		2	2.29	*	45.96	2.37	0.00	−3.15	94.85	2.37	0.00	−3.15	94.85	2.37	0.00	−3.15	94.85
	e	0	7.24	35,265	43.24	7.24	0.00	0.00	83.26	7.24	0.00	0.00	83.26	7.24	0.00	0.00	83.26
		1	7.64	*	78.35	3.74	0.27	51.05	95.23	3.74	0.27	51.05	95.23	3.74	0.27	51.05	95.23
		2	5.13	*	77.01	5.71	0.07	−11.25	92.59	5.88	0.10	−14.64	92.36	5.88	0.10	−14.64	92.36
Mean					0.02	−6.38	81.57		0.03	−6.02	81.62		0.03	−5.84	81.64		

Table 4
Solution quality of the order scenarios from the application case.

	INIT	GA			MA-PLS			MA-DLS		
	OV^{I}	OV^{7}	CV^{7}	$Imp.^{I}$ [%]	OV^{7}	CV^{7}	$Imp.^{I}$ [%]	OV^{7}	CV^{7}	$Imp.^{I}$ [%]
SC-48	810	273.14	0.04	66.28	245.57	0.07	69.68	**212.57**	0.11	73.76
SC-52	625	238.14	0.04	61.90	244.14	0.12	60.94	**229.14**	0.10	63.34
SC-56	879	287	0.11	67.35	304.71	0.12	65.33	**269.86**	0.08	69.30
SC-60	952	320	0.06	66.39	317.29	0.05	66.67	**289.14**	0.12	69.63
SC-64	989	354.57	0.04	64.15	341.86	0.12	65.43	**332.71**	0.13	66.36
SC-68	1313	396.57	0.09	69.80	388.71	0.04	70.39	**323.57**	0.08	75.36
SC-72	1314	481	0.07	63.39	395.71	0.06	69.88	**387.14**	0.10	70.54
SC-76	1228	499.57	0.08	59.32	413.57	0.11	66.32	**402.14**	0.10	67.25
SC-80	1622	516.57	0.02	68.15	**414.57**	0.05	74.44	438.14	0.07	72.99
SC-84	1871	555.43	0.06	70.31	**486.71**	0.05	73.99	497.86	0.05	73.39
SC-88	1840	506.29	0.07	72.48	615.71	0.10	66.54	**504.57**	0.07	72.58
SC-92	1979	735.29	0.12	62.85	**533.43**	0.07	73.05	554.43	0.14	71.98
SC-96	2247	748.43	0.11	66.69	**654.43**	0.05	70.88	655.57	0.09	70.82
SC-100	2243	720	0.04	67.90	**659.43**	0.09	70.60	728.57	0.08	67.52
SC-104	2672	807.29	0.07	69.79	729.57	0.06	72.70	**727.14**	0.11	72.79
SC-108	2922	**770.86**	0.07	73.62	772	0.08	73.58	779.71	0.14	73.32
SC-112	3309	948.86	0.10	71.32	820.57	0.12	75.20	**813**	0.10	75.43
SC-116	3736	988	0.07	73.55	**810**	0.09	78.32	948.43	0.09	74.61
SC-120	3492	1116.14	0.06	68.04	**880.14**	0.10	74.80	1019.71	0.06	70.80
SC-124	3645	1236.86	0.08	66.07	1152.29	0.04	68.39	**1034.71**	0.07	71.61
SC-128	3653	1321.71	0.04	63.82	1264	0.05	65.40	**1073.57**	0.06	70.61
Mean			0.07	67.29		0.08	70.12		0.09	71.14

AESD, and therefore, costs for FES and carbon dioxide emissions should be reduced significantly.

Because MA-DLS shows the highest potential for cost and emission reduction, it is used for their estimation in the next section.

5.4. Estimated cost and emission reductions

Although it is possible to determine the total quantity of requested AES (process heat in the application case) and the temporal course of AES from the machine schedule, there is no possibility for an exact quantification of the FESD because there is no deterministic functional correlation between AESD and FESD (cf. Section 2.2). Consequently, we estimate the FESD and, based on this value, costs and carbon dioxide emissions (CO_2). The following functions and parameters strongly depend on the specific (technical) characteristics of the production system in question, the production units, and the energy supply system. Therefore, in contrast to the universal applicable objective function, it is application-case specific.

For estimation purposes, we calculate the FESD for a given schedule S ($FESD^{EST,S}$) by the following function (13). This function incorporates the AESD per time slice ($AESD_{TS}$; cf. (14)), technical parameters representing radiation (rad) and leakage ($leak$) losses, a function for the actual boiler efficiency level (f_{Boiler}), and a function accounting for load alternations ($f_{LoadAlternation}$):

$$FESD^{EST,S} = f(AESD_{TS}, rad, leak, f_{Boiler}, f_{LoadAlternation}) \quad (13)$$

The calculation of $FESD^{EST,S}$ is based on the amount of $AESD_{TS}$ resulting from the (optimized) schedule (calculated per time slice) as follows:

$$AESD_{TS} = \sum_{j=1}^{n} \sum_{i=1}^{n_j} r_{ij}(TS) \quad (14)$$

As an intermediate step, the $AESD_{TS}$ has to be adjusted by the two technical parameters representing radiation and leakage losses during transport of AES from source (CU) to consumer (PU). In terms of the given steam boiler, the difference between AESD and FESD is mainly influenced by the following two factors: the efficiency of the steam-boiler operation mode and efficiency losses due to rapid changes in steam-boiler operations. Both are calculated by specific functions derived from the application case. The first function depends on the boiler utilization resulting from the cumulated AESD; the latter depends on the height of the load alternations.

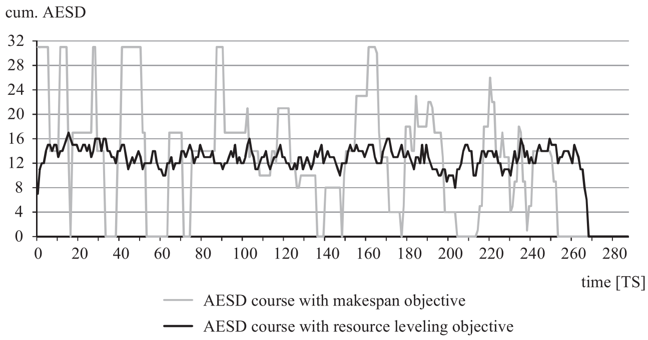

Fig. 7. Comparison of cumulated AES demands (SC124, experiment 1).

Based on this estimation of the total FESD, costs and emission reduction can be calculated with the following factors (that are also application-case specific): 0.04€ per unit FES and 0.18 kg CO_2 per unit FES. The presented results in Table 5 are again mean values based on seven experiments per scenario.

The results show that the energy-oriented scheduling approach is able to increase energy efficiency in an impressive manner. By calculating a schedule with MA-DLS, costs for FES can be reduced by 20.51% or 1.050€ (per day) on average. Because this savings can be realized every production day, a total savings of 231.000€ per year can be achieved (assuming 220 working days per year). Consequently, carbon dioxide emissions can be reduced by 20.51% or 4774.39 kg (per day) and 1.050.365.8 kg per year on average.

These savings could be further improved if the conversion unit were to be replaced by one that is dimensioned for the leveled AES demands. However, it must be noted that this technological measure is only possible if an energy-oriented scheduling approach is used.

Table 5
Estimated cost and carbon dioxide emission reductions.

	INIT	MA-DLS			INIT	MA-DLS		
	Cost [€]	Cost [€]	Imp. [%]	CV[7]	CO_2 [kg]	CO_2 [kg]	Imp. [%]	CV[7]
SC-48	3480.0	3185.8	8.45	0.002	15,823.7	14,486.2	8.45	0.002
SC-52	3559.0	3187.5	10.44	0.002	16,183.3	14,493.9	10.44	0.002
SC-56	3682.1	3182.0	13.58	0.001	16,742.8	14,469.0	13.58	0.001
SC-60	3779.9	3189.7	15.61	0.002	17,187.6	14,504.1	15.61	0.002
SC-64	3939.8	3193.4	18.94	0.002	17,914.6	14,520.8	18.94	0.002
SC-68	4047.9	3222.0	20.40	0.003	18,406.2	14,650.7	20.40	0.003
SC-72	4191.0	3290.4	21.49	0.005	19,056.7	14,961.6	21.49	0.005
SC-76	4396.1	3361.4	23.54	0.004	19,989.4	15,284.8	23.54	0.004
SC-80	4534.6	3461.8	23.66	0.004	20,619.3	15,741.3	23.66	0.004
SC-84	4742.6	3582.2	24.47	0.002	21,565.2	16,288.8	24.47	0.002
SC-88	4904.6	3735.0	23.85	0.003	22,301.9	16,983.5	23.85	0.003
SC-92	5078.9	3876.4	23.68	0.005	23,094.4	17,626.2	23.68	0.005
SC-96	5254.4	4030.6	23.29	0.004	23,892.2	18,327.4	23.29	0.004
SC-100	5423.5	4196.7	22.62	0.003	24,661.4	19,082.7	22.62	0.003
SC-104	5654.9	4370.3	22.72	0.003	25,713.3	19,872.1	22.72	0.003
SC-108	5792.4	4495.7	22.39	0.002	26,338.8	20,442.5	22.39	0.002
SC-112	6034.5	4645.7	23.01	0.003	27,439.5	21,124.5	23.01	0.003
SC-116	6202.2	4828.1	22.16	0.005	28,202.2	21,953.8	22.16	0.005
SC-120	6401.5	4978.0	22.24	0.003	29,108.2	22,635.4	22.24	0.003
SC-124	6602.2	5125.5	22.37	0.004	30,021.0	23,306.4	22.37	0.004
SC-128	6788.9	5303.1	21.89	0.003	30,870.0	24,113.5	21.89	0.003
Mean	4975.7	3925.7	20.51	0.003	22,625.3	17,850.91	20.51	0.003

6. Conclusion

Economic (e.g., increasing costs for energy) and ecological (e.g., greenhouse gas emissions) reasons require consideration of energy efficiency in production. Consequently, the purpose of the energy-oriented scheduling approach presented in this contribution is to increase energy efficiency in production to achieve environmentally conscious operations.

To increase energy efficiency in production, classical scheduling problems have to be extended to address aspects concerning energy supply and energy demand. In terms of energy supply, the necessary conversion of FES into AES by conversion units (e.g., steam boilers) plays a central role given that inefficiencies in this process are mainly responsible for losses of useful energy and unnecessary emissions. Because these inefficiencies primarily result from a strongly varying AES demand, the energy-oriented scheduling approach developed here minimizes these inefficiencies by leveling the cumulated AES demand. To be able to level this cumulated demand, detailed information about the AES demand of single production orders is required. Therefore, we propose splitting production orders into operations that have individual and constant AES demands so that their sequence reflects the AES demand profile of a production order. The resulting energy-oriented scheduling problem is defined by the underlying identical parallel machine environment and the resource leveling objective. To solve this complex scheduling problem, a two-step solution method based on problem decomposition is applied. The first step allocates production orders to machines, and the second step determines starting dates for each production order. For the second step, a GA and two MAs are developed. The evaluation of the proposed energy-oriented scheduling approach by problem instances based on an application case from the textile industry has shown its suitability for minimizing costs for FES and carbon dioxide emissions. Both can be reduced by approximately 20% on average.

With regard to the area of energy-oriented scheduling, further research should investigate other objective functions to represent energy-oriented objectives (e.g., by explicitly considering efficiency functions of CUs or the undesired FES/AES demand variations) and other (meta-) heuristics should also be taken into consideration. Regarding other solution methods, the effects of a flexible order to machine allocation should also be examined.

Furthermore, the transferability of the proposed generic energy-oriented scheduling approach (e.g., the modeling of different energy resource requirements according to operations) to other production systems seems to be worth investigating. The incorporation of technical subtleties such as cyclic energy demands, heat recovery systems, or different types of energy streams could also be a topic for further research.

Appendix A

See Table A1.

Table A1
Processing times of small test instances.

	n	n_j	p_{1j}	p_{2j}	p_{3j}	p_{4j}	p_{5j}	p_{6j}
TI-1	4	4	3	1	2	1		
			3	2	1	2		
			2	2	4	1		
			3	2	3	2		
TI-2	4	6	3	1	2	2	2	1
			4	1	1	1	2	2
			2	1	3	1	3	2
			2	1	3	1	3	2
TI-3	6	4	3	1	2	1		
			3	2	1	2		
			2	2	4	1		
			3	2	3	2		
			3	2	1	2		
			2	2	4	1		

References

[1] Kleindorfer PR, Singhal K, van Wassenhove LN. Sustainable operations management. Prod Oper Manag 2005;14(4):482–92.

[2] Fang K, Uhan N, Zhao F, Sutherland JW. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. J Manuf Syst 2011;30(4):234–40.

[3] Ngai E, To CK, Ching VS, Chan L, Lee MC, Choi Y, et al. Development of the conceptual model of energy and utility management in textile processing: a soft systems approach. Int J Prod Econ 2012;135(2):607–17.

[4] Rager M. Energieorientierte Produktionsplanung: Analyse, Konzeption und Umsetzung. Wiesbaden: Gabler; 2008.

[5] Fleischmann B, Meyr H. Planning hierarchy, modeling and advanced planning systems. In: Kok AG, de Graves SC, editors. Supply chain management: design, coordination and operation. Amsterdam: Elsevier; 2003. p. 457–523.

[6] Blazewicz J, Ecker KH, Pesch E, Schmidt G, Weglarz J. Handbook on scheduling: from theory to applications. Berlin, Heidelberg, New York: Springer; 2007.

[7] Pinedo ML. Planning and scheduling in manufacturing and services. 2nd ed.. Dordrecht, Heidelberg, London, New York: Springer; 2009.

[8] Artigues C, Lopez P, Haït A. The energy scheduling problem: industrial case-study and constraint propagation techniques. Int J Prod Econ 2011;143 (1):13–23.

[9] Payne FW, Thompson RE. Efficient boiler operations sourcebook. 4th ed.. Lilburn: Fairmont Press; 1996.

[10] Heselton KE. Boiler operator's handbook. Lilburn: Fairmont Press; 2005.

[11] Doty S, Turner WC. Energy management handbook. 7th ed.. Lilburn: Fairmont Press; 2009.

[12] Brucker P, Drexl A, Mohring R, Neumann K, Pesch E. Resource-constrained project scheduling: notation, classification, models, and methods.pdf. Eur J Oper Res 1999;112(1):3–41.

[13] Neumann K, Zimmermann J. Resource levelling for projects with schedule-dependent time windows. Eur J Oper Res 1999;117(3):591–605.

[14] Gather T, Zimmermann J, Bartels J. Exact methods for the resource levelling problem. J Sched 2011;14(6):557–69.

[15] Gahm C, Dünnwald B, Sahamie R. A multi-criteria master production scheduling approach for special purpose machinery. In: Grubbstrom RW, Hinterhuber HH, editors. Proceedings of the seventeenth international working seminar on production economics: Pre-Prints; 2012. p. 165–77.

[16] Mouzon G, Yildirim MB, Twomey J. Operational methods for minimization of energy consumption of manufacturing equipment. Int J Prod Res 2007;45(18–19):4247–71.

[17] Bunse K, Vodicka M, Schonsleben P, Brülhart M, Ernst FO. Integrating energy efficiency performance in production management – gap analysis between industrial needs and scientific literature. J Clean Prod 2011;19(6–7):667–79.

[18] Salgado F, Pedrero P. Short-term operation planning on cogeneration systems: a survey. Electr Power Syst Res 2008;78(5):835–48.

[19] Gunasekaran A, Ngai EW. The future of operations management: an outlook and analysis. Int J Prod Econ 2012;135:687–701.

[20] Papalexandri KP, Efstratios PN, Kalitventzeff B. Modelling and optimization aspects in energy management and plant operation with variable energy demands – application to industrial problems. Comput Chem Eng 1998;22 (9):1319–33.

[21] Ashok S. Peak-load management in steel plants. Appl Energy 2006;83 (5):413–24.

[22] Zhang B, Hua B. Effective MILP model for oil refinery-wide production planning and better energy utilization. J Clean Prod 2007;15(5):439–48.

[23] Subaï C, Baptiste P, Niel E. Scheduling issues for environmentally responsible manufacturing: the case of hoist scheduling in an electroplating line. Int J Prod Econ 2006;99(1–2):74–87.

[24] Mouzon G, Yildirim MB. A framework to minimize total energy consumption and total tardiness on a single machine. Int J Sustain Eng 2008;1:105–16.

[25] Nolde K, Morari M. Electrical load tracking scheduling of a steel plant. Comput Chem Eng 2010;34(11):1899–903.

[26] He Y, Liu B, Zhang X, Gao H, Liu X. A modeling method of task-oriented energy consumption for machining manufacturing system. J Clean Prod 2012;23 (1):167–74.

[27] Younis M, Saad B. Optimal resource leveling of multi-resource projects. Comput Ind Eng 1996;31(1–2):1–4.

[28] Neumann K, Zimmermann J. Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. Eur J Oper Res 2000;127:425–43.

[29] Caramia M, Dell'Olmo P. Effective resource management in manufacturing systems: optimization algorithms for production planning. London: Springer; 2006.

[30] Anagnostopoulos KP, Koulinas GK. A simulated annealing hyperheuristic for construction resource levelling. Constr Manag Econ 2010;28(2):163–75.

[31] Drótos M, Kis T. Resource leveling in a machine environment. Eur J Oper Res 2011;212(1):12–21.

[32] Rieck J, Zimmermann J, Gather T. Mixed-integer linear programming for resource leveling problems. Eur J Oper Res 2012;221(1):27–37.

[33] Roca J, Pugnaghi E, Libert G. Solving an extended resource leveling problem with multiobjective evolutionary algorithms. Int J Comput Intell 2008;4: 289–300.

[34] Ballestín F, Schwindt C, Zimmermann J. Resource leveling in make-to-order production: modeling and heuristic solution method. Int J Oper Res 2007;4 (1):50–62.

[35] Geng J, Weng L, Liu S. An improved ant colony optimization algorithm for nonlinear resource-leveling problems. Comput Math Appl 2011;61(8):2300–5.

[36] Moscato P, Cotta C. A gentle introduction to memetic algorithms. In: Glover F, Kochenberger GA, editors. Handbook of metaheuristics. Boston: Kluwer Academic Publishers; 2003. p. 105–44.

[37] Eiben AE, Smith JE. Introduction to evolutionary computing. New York: Springer; 2003.

[38] Silver EA. An overview of heuristic solution methods. J Oper Res Soc 2004;55 (9):936–56.

[39] Scholl A, Klein R, Jürgens C. BISON: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem. Comput Oper Res 1997;24 (7):627–45.

[40] Chen B. Parallel scheduling for early completion. In: Leung JY, editor. Handbook of scheduling: algorithms, models, and performance analysis. Boca Raton: Chapman & Hall/CRC; 2004. p. 175–84.

[41] Cheng TCE, Sin CCS. A state-of-the-art review of parallel-machine scheduling research. Eur J Oper Res 1990;47(3):271–92.

[42] Graham RL. Bounds for certain multiprocessing anomalies. Bell Syst Tech J 1966;XLV(9):1563–81.

[43] Coffman EJ, Garey M, Johnson D. An application of bin-packing to multiprocessor scheduling. SIAM J Comput 1978;7(1):1–17.

[44] Blazewicz J, Ecker KH, Pesch E, Schmidt G, Weglarz J. Scheduling computer and manufacturing processes. 2nd ed.. Berlin, Heidelberg: Springer; 2001.

[45] Reeves C. Genetic algorithms. In: Glover F, Kochenberger GA, editors. Handbook of metaheuristics. Boston: Kluwer Academic Publishers; 2003. p. 55–82.

[46] Glover F, Kochenberger GA, editors. Handbook of metaheuristics. Boston: Kluwer Academic Publishers; 2003.

[47] Mitchell M. An introduction to genetic algorithms. 8th ed.. Cambridge, MA: MIT Press; 2002.

[48] Eiben AE, Smith JE. Evolutionary algorithms. In: Neri F, Cotta C, Moscato P, editors. Handbook of memetic algorithms. Berlin, Heidelberg: Springer; 2012. p. 9–28.

[49] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Boston, San Francisco, New York: Addison-Wesley; 1989.

[50] Bäck T, Fogel DB, Michalewicz Z, editors. Evolutionary computation. New York, London: Taylor & Francis; 2000.

[51] Goldberg DE. The design of innovation: lessons from and for competent genetic algorithms. 2nd ed.. Boston: Kluwer Academic; 2002.

[52] Neri F, Cotta C, Moscato P, editors. Handbook of memetic algorithms. Berlin, Heidelberg: Springer; 2012.

[53] Kallel L, Naudts B, Rogers A, editors. Theoretical aspects of evolutionary computing. Berlin: Springer; 2001.

[54] Booker LB, Fogel DB, Whitley D, Angeline PJ, Eiben AE. Recombination. In: Bäck T, Fogel DB, Michalewicz Z, editors. Evolutionary computation. New York, London: Taylor & Francis; 2000. p. 256–307.

[55] Bäck T, Fogel DB, Whitley D, Angeline PJ. Mutation operators. In: Bäck T, Fogel DB, Michalewicz Z, editors. Evolutionary computation. New York, London: Taylor & Francis; 2000. p. 237–55.

[56] Bäck T. Introduction to evolutionary algorithms. In: Bäck T, Fogel DB, Michalewicz Z, editors. Evolutionary computation. New York, London: Taylor & Francis; 2000. p. 59–63.

[57] Grefenstette JJ. Proportional selection and sampling algorithms. In: Bäck T, Fogel DB, Michalewicz Z, editors. Evolutionary computation. New York, London: Taylor & Francis; 2000. p. 172–80.

[58] Baker JE. Reducing bias and inefficiency in the selection algorithm. In: Grefenstette JJ, editor. Genetic algorithms and their applications: proceedings of the second international conference on genetic algorithms: July 28–31, 1987. Massachusetts Institute of Technology, Cambridge, MA. Hillsdale: L. Erlbaum Associates; 1987. p. 14–21.