International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:11, No:5, 2017

A Memetic Algorithm for an Energy-Costs-Aware Flexible Job-Shop Scheduling Problem

Christian Böning, Henrik Prinzhorn, Eric C. Hund, Malte Stonis

Abstract**—**In this article, the flexible job-shop scheduling problem is extended by consideration of energy costs which arise owing to the power peak, and further decision variables such as work in process and throughput time are incorporated into the objective function. This enables a production plan to be simultaneously optimized in respect of the real arising energy and logistics costs. The energy-costs-aware flexible job-shop scheduling problem (EFJSP) which arises is described mathematically, and a memetic algorithm (MA) is presented as a solution. In the MA, the evolutionary process is supplemented with a local search. Furthermore, repair procedures are used in order to rectify any infeasible solutions that have arisen in the evolutionary process. The potential for lowering the real arising costs of a production plan through consideration of energy consumption levels is highlighted.

Keywords**—**Energy costs, flexible job-shop scheduling, memetic algorithm, power peak.

I. Introduction

IN manufacturing industry one of the tasks within production planning is the allocation of production jobs to a machine and a time domain. Examples of logistic decision variables that need to be optimized within this process include makespan and output lateness [1]. One of the most complicated combinatorial optimization problems within production planning is the JSP. Within the JSP, i jobs and k machines are considered. Each job consists of j operations which have to be processed in a specified sequence. Each operation is assigned to a technologically suitable machine. The aim of the JSP is to find a suitable sequence of operations on the machines which typically optimizes one decision variable under consideration. Makespan [2] is a decision variable that is frequently used for the JSP. Garey et al. have proved that the JSP is NP-hard [3]. The flexible job-shop scheduling problem (FJSP) is a generalization of the JSP. Here, the operations can be freely assigned to the available machines. Operations firstly have to be assigned to machines (routing) before scheduling can take place on the machine. As the FJSP belongs to the same complexity class as the JSP, it is likewise NP-hard. For the problems which include more than 20 jobs and 20 machines, it is already difficult to find an optimal solution in a reasonable computation time by using a precise commercial solver [1]. Especially practical problems will exceed the mentioned problem size easily. Therefore, in the more recent past, the focus has moved to the development of high-performance heuristics. Heuristics do not guarantee to find the optimal solution, but they are suitable for solving most planning problems and they require very little computing time.

The consideration of energy costs is also becoming an increasingly important part of production planning. Developments such as the increasing scarcity of fossil fuels, the turn away from nuclear energy in some countries, and the expansion of renewable sources of energy in order to reduce global worldwide CO_2 emissions are causing a general rise in energy costs. In Germany, the average electricity price (cents/ kWh) for the industrial sector rose by over 200% in the period from 2000 to 2012 [4]. In industrial companies, energy costs make up an ever increasing share of the total cost of ownership. That is why manufacturing industry is striving to reduce its energy costs. Research carried out in this field has focused on the energy-efficient designing of machines, energy-efficient product design, and energy-efficient production planning [5]. The first two approaches seek to reduce energy costs by decreasing energy consumption. By contrast, the energy efficient production planning approach reduces the costs of the amount of energy that is used, while energy consumption itself remains constant. However, since this approach can be implemented without the need for major investments – compared to the other two approaches – it is of particular interest for small and medium-sized companies. There are two approaches that can be used for influencing energy costs through production planning. On the one hand, the price of energy varies in many countries depending on the time of day. Depending on the country, there are roughly two or three time domains across which the price of energy per kWh varies [6], [7]. The time domain with the lowest energy price usually begins in the late evening and lasts until the early hours of the following morning. If a company therefore adjusts its production planning to ensure that energy-intensive production jobs are increasingly produced in time domains with low energy costs, a reduction in energy costs can be achieved. On the other hand, a reduction in energy costs can be achieved by reducing the power peak within production operations [8]. The power peak is formed by the energy consumption of all the machines within a specific time domain (a quarter-hour period for example). In many industrial energy tariffs, a higher power peak leads to higher energy costs. Furthermore, some tariffs of energy-intensive companies include a particular threshold regarding the power peak, which in case of a one-time exceedance, leads to a higher price per kW. However, if production planning is now used to ensure that the processing of energy-intensive production jobs is

C. Böning and H. Prinzhorn are Project Engineers with the Hanover Institute of Integrated Production (IPH) (non-profit limited company), 30169 Hannover, Germany (e-mail: boening@iph-hannover.de, prinzhorn@iph-hannover.de).

spread over time rather than being undertaken in parallel on several machines at the same time, the power peak within production operations can be reduced, and energy costs will also decrease.

The minimizing of energy costs as a decision variable within production planning must take place parallel to the existing logistic decision variables. According to Nyhuis and Wiendahl [10], the fundamental goal of production logistics is the achievement of a maximum delivery capability and reliability with the lowest possible logistic and production costs. Therefore, the four logistic decision variables throughput time, delivery reliability, utilization and work in process (WIP) level have to be considered [9], [10]. The throughput time of an operation within a job consists of processing time, set up time and waiting time, which can be split up in post-process waiting, transport and pre-process waiting. Short throughput times will lead to short deviations of completion times from due dates and therefore to a high delivery reliability. The delivery reliability can be measured by the output lateness. In order to achieve low logistic and production costs, it is necessary to achieve a maximal utilization of the available capacities and low storage and WIP levels to minimize the costs of tied-up capital [9], [10]. The conflict between the objectives of the logistic decision variables is named as the dilemma of operations planning [11], [10]. To ensure a high level of capacity utilization, a high WIP level is required. However, a high WIP level leads to extended and variating throughput times and therefore to a lower delivery reliability. When formulating a combinatorial optimization problem which considers the energy costs, a multi-objective function is therefore necessary in order to avoid unbalanced optimization, and consequently any deterioration of the established logistic decision variables.

The structure of the further article is as follows. Section II provides a brief overview of existing research approaches to the subject of this paper. Section III presents the combinatorial optimization problem. The heuristic approach that has been developed to solve the optimization problem is presented in Section IV. Section V contains a presentation of computational results and a validation of the developed approach. The article concludes with a summary in Section VI.

II. Literature Review

The FJSP is widespread in academic literature. The last two decades have seen the development of a whole series of new procedures for generating better solutions to this problem. Bruckner and Schlie were the first to describe the FJSP. They developed a polynomial algorithm which provides an optimal FJSP solution for a problem size of two jobs [12]. Many different heuristic approaches were then used in order to be able to solve problems of the size actually encountered in industry. These include Tabu Search (TS) [13], Simulated Annealing (SA) and evolutionary procedures such as genetic algorithms (GA), ant colony optimization (ACO) and artificial bee colony (ABC) algorithms [14], or particle swarm optimization (PSO) [15]. The heuristic procedures referred to above can be categorized into hierarchical and integrated approaches [2]. Hierarchical approaches reduce the complexity of the FJSP by subdividing the problem into two sub-problems which are solved sequentially: the assignment to machines takes place first, and then the scheduling on the machines. Once the first sub-problem has been solved, the second sub-problem represents the classic JSP [16]. Brandimate uses a hierarchical algorithm in which both sub-problems are solved by using TS [17]. The hierarchical approach developed by Paulli solves the machine assignment problem by using a dispatching rule, and it then uses TS to solve the scheduling problem [18]. Both the hierarchical approaches that are presented minimize the makespan decision variable. Integrated approaches carry out both planning tasks in parallel. Although they are much more difficult to formulate and solve, they usually lead to better results [2]. The integrated approach proposed by Dauzère-Pérès and Paulli is based on a TS algorithm which based on a new neighborhood structure [19]. Mastrolilli and Gabardella likewise use a TS algorithm, and they present two neighborhood functions for the FJSP [20]. Among the heuristic approaches to solving the FJSP, GAs have proved to be very effective and have therefore frequently been used. They differ in terms of their encoding and decoding schemes, the generation of the initial population, and the offspring generation strategy [21]-[23]. In the more recent past, increased use has been made of MA in which a GA is combined with a local search (LS) in order to provide a better quality of solution. Several GAs are being combined with LS procedures in order to improve individuals prior to and during the evolutionary process [24], [25]. Raaesi and Kobti use an MA in which the LS heuristic removes critical operations and reassigns them to improve the schedule [26]. The Gutiérrez and García-Magarino MA initially devises solutions which do not comply with all the constraints, but which then use repair heuristics for the respective constraints in order to obtain permissible solutions [27]. Jiang et al. combine a GA that has a multi-objective function with SA for scheduling on the machines [28].

The consideration of energy in the form of costs or consumption levels within production planning, and specifically in scheduling problems (SP), is the subject of more recent research and is not yet common practice. Rager extends the identical parallel machine scheduling (PMS) problem by considering energy costs which result from the power peak. Energy consumption may vary within a job due to the introduction of operations. In the objective function the number of machines required is minimized as well as the power peak. To solve the problem, a GA is combined with an LS procedure [29], [30]. The extended flow shop problem (FSP) by Fang et al. considers energy costs which result from the power peak. They are minimized as well as the makespan. The problem is solved using a commercial solver [31]. Luo et al. extend the FSP by considering energy costs which are dependent on the time of day. In the objective function, equal priority is given to minimizing the makespan and the energy costs. Energy consumption is assumed to be constant for a particular operation. An ACO algorithm is used to solve the extended FSP [6]. Bruzzone et al. extend the flexible flow shop problem (FFSP) by considering energy costs which result from the power peak. They follow a sequential approach to finding a solution in which the first step is the solving of the conventional FFSP. This minimizes the makespan and output lateness decision variables. The second step is then the minimization of the power peak although the assignment of operations to machines and the scheduling on the machines are not altered. Optimization is achieved by moving the start and finish time points without thereby violating any constraints. The solution is found using a commercial solver (CPLEX) and a NS algorithm [32]. Dai et al. extend the FSP with the minimization of energy consumption. Savings in energy used are achieved by deciding whether to switch an idle-running machine off or keep it running. In order to facilitate this decision, the energy consumption entailed in switching on and off is compared with the energy consumed during idle running. In the objective function makespan is minimized as well as energy consumption. To solve the problem, a GA is used in combination with a SA [5]. Moon and Park extend the FJSP by considering energy costs which are dependent on the time of day. Furthermore, the storage and retrieving of energy from an energy storage system is considered. Energy consumption is stated not for an operation, but for a machine. In the objective function the energy costs and the penalty costs for an increased makespan are considered. In order to solve the problem, it is subdivided into sub-problems and an optimal solution is found [7].

It should be mentioned in relation to the aforementioned energy-oriented approaches that the lack of consideration of logistic decision variables such as makespan and output lateness, as in the case of the PMS, may have negative consequences. These take the form of the missing of completion deadlines or increased WIP levels. The costs which result from this may exceed the savings achieved in respect of energy costs. All the approaches that have been presented have a multi-objective function. However, only in the last presented approach the decision variables are weighted with costs. In the other approaches the decision variables are treated on an equal basis. However, unless costs are considered it is not possible to find the optimal trade-off between logistic decision variables such as the makespan on the one hand and energy on the other hand. In almost all the approaches that have been presented, energy consumption is assumed to be constant during an operation, and a mean value is used to reflect this. However, in relation to the minimization of energy costs which arise owing to the power peak, this may lead to inaccuracies. The power peak is usually measured within a specific time domain. In Germany, this is 15 minutes. If the processing time for an operation exceeds this time domain, the power peak may not be correctly recorded. In one approach, the energy consumption of a machine is assumed to be constant. This is surely correct for many technological procedures and processing steps. However, in such cases the potential offered through the consideration of energy costs must also be classed as low since the interchange of jobs does not produce any effect. It is only possible to exert an influence if the capacity utilization of the machines is low and idle times consequently arise. Jobs can then be scheduled for time domains with low energy costs, and time domains with high energy costs remain unused. However, a company should aim to operate its machines as economically as possible [33]. This means that, in the best case scenario, only short idle times arise. In one of the outlined approaches, energy is saved through machines being switched off during idle running. Low capacity utilization of the machines is also necessary for this.

In the following, energy costs, which arise due to the power peak, are integrated in the FJSP. Moreover WIP and throughput time are integrated into the objective function as logistic decision variables. The pursued objective is to develop a simple method for practical application to help especially small and medium-sized enterprises (SME) considering energy costs in their production planning. The consequence of not considering energy costs and the existing cost-saving potential should be presented. Therefore, the real arising costs of a production plan have to be identified. To do this, the individual decision variables are weighted with costs. In order to solve the extended FJSP for practical problem sizes, an integrated heuristic approach is proposed in the form of a MA. Therefore, the MA's practical application has a higher priority than the performance to compute the best possible solution. It is assumed that the energy consumption levels can vary within an operation. To do this, energy phases are created as proposed by Weinert et al. [34]. In the evolutionary process, an LS is supplemented similar to the approach of Raaesi and Kobti [26]. Furthermore, similar to the approach of Gutiérrez and García-Magarino [27], repair procedures are supplemented in the MA to correct infeasible solutions which may have arisen during the evolutionary process. As in the approach of Jiang et al., a multi-objective function is used to minimize the real arising costs of the production plan [28].

III. Problem Description

The energy-costs-aware flexible job-shop scheduling problem (EFJSP) consists of $i = 1, \ldots, I$ jobs. Each job i consists of $j = 1, \ldots, J$ operations. Each operation j must be run on one of the $k = 1, \ldots, K$ machines within the planning horizon of $t = 1, \ldots, T$ periods that is under consideration. The energy consumption over time is known for each operation. The energy consumption and processing time of an operation are assumed to be non-machine-dependent. In order to be able to map energy consumption levels which fluctuate over time as accurately as possible, the energy consumption of an operation is estimated as proposed by Weinert et al., and it is subdivided into $s = 1, \ldots, S$ energy phases with constant energy consumption. An energy consumption value e_{ij} is therefore assigned to each energy phase s. This value is determined through the calculation of a mean value. Each operation consists of at least one energy phase. In this case, the energy phase includes the same number of periods as the operation. The maximum number of energy phases of an operation is equal to the number of periods the operation includes. In this case, the energy phases each comprise one period. A period length of 15 minutes is chosen since, in

Germany at least, this enables the power peak to be recorded as precisely as possible. As the basic precondition for the EFJSP, it is assumed that the processing of different operations leads to differing levels of energy consumption on a machine. This means that the energy consumption profile of production operations, and consequently the power peak which arises, can be influenced by the interchanging of jobs in the time dimension. In the EFJSP multi-objective function, the logistic decision variables output lateness, WIP and throughput time are optimized together with the power peak. The production plan which is produced can be shown in the form of a GANTT diagram (see Fig. 1).

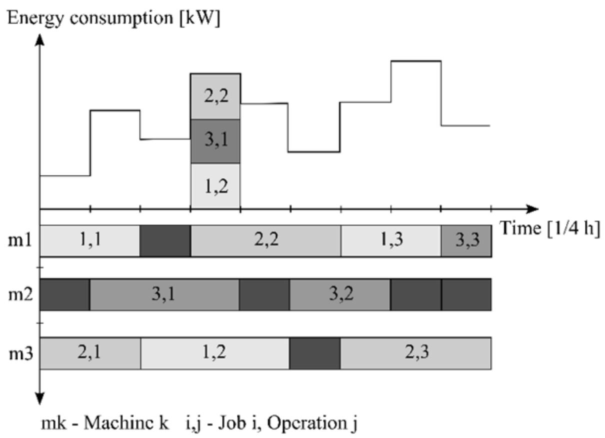

Fig. 1 Energy consumption profile and production plan

The logistic decision variables can be derived from this diagram. Output lateness is determined by comparing the time when a job is completed with the planned completion deadline (which is assumed to be equal with the delivery deadline). This involves distinguishing between completion which is too early (negative deviation) on the one hand and too late (positive deviation) on the other hand. WIP is calculated as the total of the job's waiting times. Three types can be distinguished: waiting times in the unprocessed parts store before the processing of a job begins; waiting times between the end of one operation within a job and the start of the next one; and waiting times in the finished products store between the time when a job is completed and the planed completion deadline. Together with the processing time (including setup times), the cumulative waiting times result in the throughput time of a job. The total energy consumption of the production operations is calculated as the aggregate energy consumption for all the jobs which are processed on the machines within a time period. The highest value which occurs within the planning horizon is the power peak. Minimizing the waiting times lead to the minimization of WIP and throughput times, as well as negative output lateness of jobs. Minimization of the power peak is achieved by leveling out the energy consumption of all the machines over the planning horizon that is being considered. Positive output lateness of jobs is prevented via a constraint rather than being achieved via the objective function. This means that only a reliable production plan can be produced if the completion deadlines of all the jobs are satisfied. Capacity utilization is not considered in the EFJSP's objective function as a decision variable. The machines' capacity utilization depends on the workload due to the regarded jobs and operations. The number of jobs is predetermined before starting production planning and therefore fixed during the optimization. But, the capacity utilization can be variated in the generation of the test instances (see experimental setup). Nevertheless, the costs of unused capacity which result from a low capacity utilization have to be considered to identify the real arising costs of a production plan. The costs of unused capacity of a machine consist of the machine costs per hour. Total costs of unused capacity can be minimized, if machines with higher machine costs per hour are utilized more than machines with lower machine costs per hour.

In order to identify the real arising costs of a production plan, the decision variables have to be weighted with costs for each specific application. The monetary evaluation of the waiting times is undertaken based on the costs of tied-up capital which are incurred during each period [10]. These costs must be known after each operation within a job. The costs of tied-up capital consist of the interest on the manufacturing costs that have actually been incurred for the job up to and including the current operation. The manufacturing costs consist of the total material costs (as per the parts list), machine costs (machine costs per hour), and labor costs (labor costs rates). Before the processing of a job begins, the manufacturing costs consequently only consist of the costs of materials. As the number of operations that have been completed within a job increases, the incurred costs of tied-up capital therefore also increase. As the overrunning of the completion deadline is not permitted, a costs variable for evaluating positive output lateness of jobs is not required. The power peak that is identified within a production plan is multiplied by an energy price per kW in order to obtain the energy costs. If the energy tariff contains a threshold regarding the power peak, there are two energy prices available which are determined in the energy tariff. The threshold determines which energy price is used. The cheaper energy price *LP1* is used when the power peak of the production plan is below the threshold. The more expensive energy price *LP2* is used when the power peak exceeds the threshold. The total costs of a production plan are calculated based on the logistics costs and the energy costs. The identified costs variables are incorporated into the objective function, which makes it possible to undertake a monetary comparison of different allocation schedules within the EFJSP. The following assumptions are made for the EFJSP:

- All the jobs together with their associated operations must be fully processed within the planning horizon.
- The processing of an operation must not be interrupted.
- The capacity that is utilized by the jobs must not exceed the available capacity of all the machines within the planning horizon.
- The processing sequence for the operations within each job must be adhered to.

- Multiple operations within a manufacturing order must not be run in parallel on different machines.
- Processing of an operation may only be carried out on a permitted machine.
- At least one technologically suitable machine must be available for each operation.
- Within a single period only one operation may be processed on any one machine
- Each machine is available in every period within the planning horizon.
- The processing time of an operation is identical for each permitted machine.
- The energy consumption of an operation is identical on each permitted machine.
- Energy phases within an operation may only be scheduled without any time interruptions.
- Energy phases within an operation may only be scheduled on one machine.
- The sequence of energy phases within an operation must be adhered to.

Table I shows the indices, parameters and variables used in the EFJSP. Afterwards the mathematical model is presented.

$$Min\ Z = E_{max} \cdot LP1 \cdot (1-b) + E_{max} \cdot LP2 \cdot b + \sum_k uc_k \cdot \left(T - \sum_i \sum_{j=1}^{No_i} \sum_{s=smin_{i,j}}^{smax_{i,j}} \sum_t x_{i,j,s,k,t}\right) + \sum_i \left(tsj_{i,1} - 1\right) \cdot c1_i + \sum_i \sum_{j=1}^{No_i - 1} \left(tsj_{i,j+1} - tej_{i,j} - 1\right) \cdot c_{i,j} + \sum_i \sum_{j=No_i}^{No_i} \left(d_i - tej_{i,j}\right) \cdot c_{i,j} \quad (1)$$

$$E_t = \sum_i \sum_j \sum_s \sum_k e_{i,s} \cdot x_{i,j,s,k,t} \quad \forall\, t \in T \quad (2)$$

$$E_{max} = \max_{1 \le t \le T} \{E_t\} \quad (3)$$

$$b \cdot M \ge E_{max} - ET \quad (4)$$

$$tss_{i,j,s} + p_{i,s} = tss_{i,j,s+1} \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j} \quad (5)$$

$$\gamma_{i,j,s,k} = \gamma_{i,j,s+1,k} \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j}, k \in K \quad (6)$$

$$\sum_k \gamma_{i,j,s,k} = 1 \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j} \quad (7)$$

$$\sum_t x_{i,j,s,k,t} = \gamma_{i,j,s,k} \cdot p_{i,s} \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j}, k \in K \quad (8)$$

$$tsj_{i,j} + \sum_{s=smin_{i,j}}^{smax_{i,j}} p_{i,s} \le tsj_{i,j+1} \quad \forall\, i \in I, j \in AA_i \quad (9)$$

$$tej_{i,j} - tsj_{i,j} + 1 = \sum_{s=smin_{i,j}}^{smax_{i,j}} p_{i,s} \quad \forall\, i \in I, j \in AA_i \quad (10)$$

$$tsj_{i,j} = tss_{i,j,s} \quad \forall\, i \in I, j \in AA_i, s \in smin_{i,j} \quad (11)$$

$$tej_{i,j} = tes_{i,j,s} \quad \forall\, i \in I, j \in AA_i, s \in smax_{i,j} \quad (12)$$

$$\left(x_{i,j,s,k,t} - x_{i,j,s,k,t-1}\right) \le Ys_{i,j,s,k,t} \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j}, k \in K, t \in T \quad (13)$$

$$\left(x_{i,j,s,k,t} - x_{i,j,s,k,t+1}\right) \le Ye_{i,j,s,k,t} \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j}, k \in K, t \in T \quad (14)$$

$$tss_{i,j,s} = \sum_k \sum_t Ys_{i,j,s,k,t} \cdot t \quad \forall\, i \in I, j \in AA_i,\ s \in PA_{i,j} \quad (15)$$

$$tes_{i,j,s} = \sum_k \sum_t Ye_{i,j,s,k,t} \cdot t \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j} \quad (16)$$

$$\sum_k \sum_{t=1}^{d_i} Ye_{i,j,s,k,t} = 1 \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j} \quad (17)$$

$$\sum_k \sum_t Ys_{i,j,s,k,t} = 1 \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j} \quad (18)$$

$$\sum_i \sum_{j=1}^{No_i} \sum_{s=smin_{i,j}}^{smax_{i,j}} x_{i,j,s,k,t} \le 1 \quad \forall\, k \in K, t \in T \quad (19)$$

$$x_{i,j,s,k,t} \le m_{i,j,k} \quad \forall\, i \in I, j \in AA_i, s \in PA_{i,j}, k \in K, t \in T \quad (20)$$

The objective (1) requires the minimization of energy and logistics costs. In the first line, the power peak is minimized. In the second line, the costs of unused capacity are minimized. In the subsequent lines of the objective function, waiting times before the start of (third line), between (fourth line), and after the end of (fifth line) the processing of a job's operations are minimized. Constraint (2) provides the total energy consumption for all orders within each period. Constraint (3) determines the power peak. Constraint (4) determines the energy price which is used. Constraints (5) to (8) relate to the correct scheduling of the energy phases. Constraint (5) guarantees that all energy phases within a job are scheduled to follow each other without interruption. Constraint (6) leads to all energy phases within an operation being scheduled on only one machine. Constraint (7) ensures that each energy phase of an operation within a job is scheduled exactly once. Constraint (8) represents the link between the non-period-dependent consideration of energy phases and the EFJSP's fundamental binary decision variable $x_{i,j,s,k,t}$. Constraints (9) to (18) ensure the correct scheduling of the operations. Constraint (9) ensures that the processing sequence of the operations is followed, and it prevents any overlaps in the scheduling. Constraint (10) ensures that the interval between the start and end time of an operation comprises as many periods as are required in order to process all the energy phases within this operation. In constraints (11) and (12) the start and end times of the operations are determined based on the start and end times of the energy phases. Constraints (13) and (14) identify when an energy phase begins and when it ends. Constraints (15) and (16) determine the start and end times of the energy phase as a numerical value. Constraint (17) prevents positive output lateness of jobs. It requires all the energy phases of an operation within a job to be completed by the job completion deadline. Constraint (18) guarantees that each energy phase within the planning horizon is started just once. Constraints (19) to (20) relate to the selection of a machine. Constraint (19) ensures that in each period on any machine only one energy phase of an operation within a job can be processed. This prevents any double allocations. Constraint (20) ensures that an energy phase of an operation within a job can only be scheduled on a machine if this energy phase is actually allowed to be processed on the regarded machine.

TABLE I
INDICES, PARAMETERS AND VARIABLES OF THE EFJSP

Indices	
$i = 1, \ldots, I$	Jobs
$j = 1, \ldots, J$	Operations within job i
$s = 1, \ldots, S$	Energy phases within operation j of job i
$k = 1, \ldots, K$	Machines
$t = 1, \ldots, T$	Time periods ($T = \max \{d_i\}$)
Parameters	
AA_i	Quantity of all operations j within job i
PA_{ij}	Quantity of all energy phases s within operation j of job i
No_i	Number of operations within a job i
d_i	Completion deadline for job i
$e_{i,s}$	Energy consumption within energy phase s of job i
$p_{i,s}$	Processing time for energy phase s of job i
$c_{i,j}$	Costs of tied-up capital per period after operation j of job i has been processed
$c1_i$	Costs of tied-up capital before processing of job i begins
uc_k	Costs of unused capacity per period of machine k
ET	threshold in kW, above which the energy price per kW increases
LP1	Energy price per kW, if the power peak not exceeds ET
LP2	Energy price per kW, if the power peak exceeds ET
M	very big number
$m_{i,j,k}$	= 1 if operation j of job i can be processed on machine k; = 0, otherwise
$smax_{ij}$	Last assigned energy phase s of operation j within job i
$smin_{ij}$	First assigned energy phase s of operation j within job i
Variables	
$x_{i,j,s,k,t}$	= 1 if energy phase s of operation j within job i in period t is processed on machine k; = 0, otherwise
$\gamma_{i,j,s,k}$	= 1 if energy phase s of operation j within job i is scheduled for machine k; = 0, otherwise
$Ye_{i,j,s,k,t}$	= 1 if energy phase s of operation j within job i in period t ends on machine k; = 0, otherwise
$Ys_{i,j,s,k,t}$	= 1 if energy phase s of operation j within job i in period t begins on machine k; = 0, otherwise
E_t	Cumulative energy value over all machines in period t
E_{max}	Power peak of the production plan in kW
b	= 1 if power peak of the production plan exceeds the threshold; = 0, otherwise
$tss_{i,j,s}$	Start time of energy phase s of operation j within job i
$tes_{i,j,s}$	End time of energy phase s of operation j within job i
$tsj_{i,j}$	Start time of operation j within job i
$tej_{i,j}$	End time of operation j within job i

IV. PROPOSED ALGORITHM

As the EFJSP is an extension of the FJSP, it can likewise be regarded as being NP-hard. A heuristic approach is therefore required for solving large problem sizes within the EFJSP. A MA is used to solve the EFJSP. The MA attempts to mimic the natural evolutionary process. Starting with an initial population, the algorithm executes genetic operators in order to produce offspring which ideally will have a higher level of fitness than their parents. The structure of the MA can be described as follows:

1) Coding: The coding based on a solution of the EFJSP (production plan) in the shape of a GANTT diagram. This produces a chromosome with several strings which contains the production plan information in coded form. Each chromosome contains a solution of the EFJSP.
2) Initial population: The production of the initial population takes place based on the latest starting time rule and a random component. This ensures that only permissible solutions are generated which do not violate any constraints.
3) Fitness evaluation: The fitness of each chromosome within the current population is calculated. The EFJSP objective function is used as the fitness evaluation function.
4) Selection: In each iteration chromosomes in the population are selected for reproduction through n-size tournament selection.
5) Crossover: In the evolutionary process the exchanging of genetic information between two chromosomes creates new chromosomes. Infeasible solutions which arise are repaired by means of repair procedures.
6) Mutation: The chromosomes within the evolutionary process undergo further change due to random mutation. The mutation takes place due to the moving of an operation, including all its associated energy phases, to another machine. This happens separately for each chromosome.
7) Local search: A search within the neighborhood of the chromosomes in the evolutionary process is performed in order to further improve fitness. This involves the fitness function specifically searching for operations which cause high costs and attempting to reduce these costs by changing the scheduling.
8) Reinsertion scheme: At the end of the evolutionary process the reinsertion scheme is used to decide which chromosomes will be removed from the current population and which chromosomes that have been newly created in the evolutionary process will be incorporated into the population. In this MA, elitist reinsertion is used.
9) Stopping criterion: Steps 3. - 8. describe the running of a generation and they are repeated until the stopping criterion is reached. The stopping criterion used by this MA is the maximum computing time.

The description of the MA now continues with a more detailed explanation of the individual steps in the algorithm.

A. Coding

In the presented coding, a chromosome consists of an energy phase string and a machine string. As encoding scheme for the energy phase string, a permutation with repetition is used to encode a solution for the EFJSP as proposed by Bierwirth [35]. This is particularly suitable for sequencing problems, such as the JSP [36]. The energy phase string contains the sequence of energy phases *s* of an operation within all jobs *I* (A_{ijs}). Only the job index and not the operation and energy phase index are integrated in the string. In the energy phase string, the job index for a job is repeated according to the number of energy phases within the job. This ensures that any permutation of job indices can be interpreted as a feasible sequence of energy phases [37]. Therefore, decoding of a chromosome always leads to a feasible production plan, because a violation of the processing sequence of the energy phases and operations within each job is not possible.

In Fig. 2, job 1 consists of two operations and a total of three energy phases. Job index 1 consequently occurs three times within the energy phase string. The occurrence of the job index provides information about the energy phase. Job index 1 appears for the second time in the fifth gene of the energy phase string. This is consequently the second energy phase of job 1 (A_{122}). Unallocated time domains are depicted by idle time phases ($B_l = 1, \ldots, QtB$). The total number of idle time phases (QtB) is calculated by means of (21) as the capacity available in the planning horizon less the total processing times of the jobs. All the idle time phases have job index 0 and a processing time of one period.

$$QtB = T \cdot K - \sum_i \sum_s p_{i,s} \qquad (21)$$

In the machine string, the energy phases are assigned to a specific machine. A gene within the machine string describes a set energy or idle time phase. The gene value (allele) describes the machine on which the energy or idle time phase is scheduled.

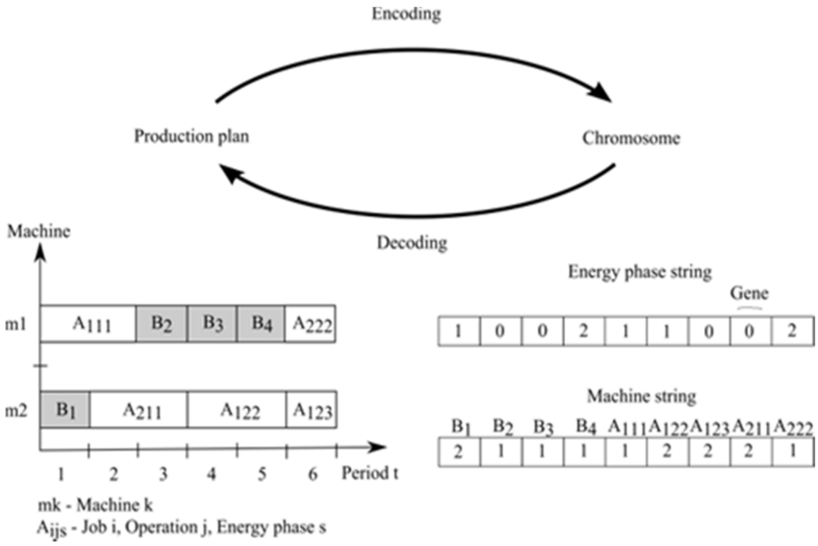

Fig. 2 Coding

Decoding a chromosome in a production plan requires the energy phase string and the machine string. The gene values of the energy phase string are considered from left to right and the corresponding energy or idle time phases is allocated in the production plan to the machine, specified in the machine string. An energy or idle time phase is always allocated at the earliest possible available period. In Fig. 2, the first gene value of the energy phase string is job index 1. This is the first time that job index 1 occurs in the energy phase string. Hence, the corresponding energy phase is A_{111} which has to be carried out on machine 1. Energy phase A_{111} is now allocated at the earliest possible available period of machine 1 (period 1). The next gene value of the energy phase string is job index 0 and represents the first idle time phase B_1. B_1 is allocated at machine 2, again at the earliest possible available period (period 1). The idle time phases are necessary to allocate the energy phases at a definite start and end period in the production plan. Without the idle time phases decoding of a chromosome could lead to several possible production plans. Due to the consideration of idle time phases during encoding, the information of the definite position of an energy phase in the production plan is contained in the chromosome without the explicit values of the start and end periods of each energy phase.

B. Initial Population

The generation of the initial population ensures that only permissible chromosomes can be produced which do not violate any constraints. This is ensured by an initialization procedure which consists of a priority rule and a random component. The generating of a permissible chromosome is divided into two initialization phases. In the first initialization phase the operations are scheduled on the machines together with the associated energy phases (without idle time phases). In order to do this, (22) is initially used to determine the latest possible start time $slate_{i,s}$ for each energy phase based on the completion deadline and the subsequent energy phases within the manufacturing job. In this equation, w is the energy phase which is currently being considered within job i. Quantity S_i contains all the energy phases within job i.

$$slate_{i,s} = d_i - \sum_{s=w}^{S_i} p_{i,s} \quad \forall\, i \in I, s, w \in S_i \qquad (22)$$

The subsequent steps are repeated until all operations and associated energy phases have been scheduled. The energy phase with smallest latest possible start time value is selected. Energy phases of an operation are always jointly scheduled. Energy phase w represents the first energy phase of the operation under consideration. The capacity requirements of the operation are calculated by adding together the processing times for all the energy phases of the operation. Then, the earliest possible start time $searly_i$ for job i to which the selected operation relates is determined according to (23). This consists of the total processing times of the operations, and the associated energy phases, within job i which occur before energy phase w. If energy phase w is the first energy phase in job i, $searly_i$ equals zero.

$$searly_i = \sum_{s=1}^{w-1} p_{i,s} \quad \forall\, s, w \in S_i \tag{23}$$

After that, a machine on which the regarded operation can be processed is selected at random. Furthermore, the free capacity in the form of unallocated scheduling periods for the machine must be equal to or greater than the capacity requirements of the regarded operation. Now all the potential start times spot on the selected machine between $slate_{i,s}$ and $searly_i$, are determined. If there are several possible start times, a selection is made at random. The start and end times of the scheduled operation are incorporated into the ancillary strings. If no potential start time can be found, all the other permissible machines are considered in turn. If this also fails to find a start time, the initialization is stopped, the production plan is deleted, and the initialization is then restarted. In the second initialization phase, idle time phases are then assigned in the planning periods for the individual machines which have not yet been scheduled. The initialization is repeated until a specified population size μ is achieved. However, a chromosome must only occur once in the population. Furthermore, each chromosome must also have a different level of fitness.

C. Fitness Function

The fitness evaluation function calculates the respective fitness value for each chromosome in the population. The EFJSP objective function is used as the fitness function. This necessitates the decoding of the chromosomes. The GANTT diagram is used to determine fitness. As the energy and logistics costs have to be minimized, the MA searches for solutions with a low fitness value. A low fitness value equates to a high level of fitness. In the evolutionary process, infeasible solutions may arise (see crossover). These have a penalty term applied to them when fitness is determined. The penalty term is comparable to a “big M”, a very high value of 1,000,000.

D. Selection

The selection process involves selecting chromosomes from the population for the evolutionary process. Chromosomes are firstly selected for the mating pool through tournament selection. This involves ξ chromosomes being randomly selected from the population and the chromosome with the best fitness being copied to the mating pool. The process is carried out μ-times until the number of chromosomes in the mating pool matches the number in the population. Chromosomes can occur several times in the mating pool. Two chromosomes are then selected at random from the mating pool for the evolutionary process.

E. Crossover

In order to produce two new chromosomes, both selected chromosomes are initially crossed with each other in the evolutionary process. In order to do this, the precedence preserving order-based crossover (POX) procedure is used for the energy phase string, and the one-point crossover procedure is used for the machine string. The POX procedure guarantees together with the used encoding scheme the correct number of energy and idle time phases in the energy phase string of the new chromosomes. In the case of the one-point crossover procedure in the machine string, the crossover point must not be placed between energy phases of the same operation. This ensures that all the energy phases of an operation following the crossover are on the same machine. The crossover procedures may nevertheless produce infeasible solutions. The following impermissibilities may arise:

- Breaching of machine capacity (the total of the processing times of the energy and idle time phases on the machine is greater or less than T).
- Over running of the completion deadline.
- Deviation from the sequence in which the operations should be processed.
- Contravention due to time interruptions between the processing of energy phases of an operation.

If use of the crossover procedures produces infeasible solutions, two repair procedures are used to repair the chromosomes. The repair procedures are carried out one after the other. If a chromosome cannot be fully repaired, the described crossover procedures are carried out again. If no permissible chromosomes could be produced after running a specified number of crossover procedures (Limit_CO), the evolutionary process is continued using an impermissible chromosome. The aim of the first repair procedure is to rectify any breach of machine capacity. In order to do this, operations within a chromosome, together with all the associated energy phases, are moved from the most heavily over-loaded machine to the most lightly loaded machine. Changes are made to the specific genes which have caused the impermissibility. The aim of the second repair procedure is to rectify the three other impermissibilities that have been referred to. In order to do this, the machine assignment is fixed in the machine string, and an attempt is made to interchange the sequence of energy and idle time phases within the energy phase string in such a way that permissible chromosomes are produced. Both repair procedures include a fixed number of repair attempts (Limit_shiftingKap and Limit_repair). If the number of repair attempts is exceeded, the crossover procedures recommence.

F. Mutation

The chromosomes within the evolutionary process undergo further change due to random mutation. Whether a chromosome mutates is decided by the mutation rate (Mutate_rate). The mutation is based on the random moving of an operation to another machine. This firstly involves the random selection of a machine. The largest continuous range of idle time phases on this machine into which an operation can be moved is identified. Then, another machine is specified from which an operation including all its associated energy phases can be moved. One operation is selected at random from all the operations assigned to this machine. If this operation fits into the other machine's unused time domain and the operation is allowed to be produced on that machine, the move is made. If these preconditions are not fulfilled, another permissible machine is considered, or once all the permissible machines have been considered, another operation that has to be moved is selected. An operation may only be moved once. A mutation counter counts the attempts and stops the mutation after a randomly generated number of attempts.

G. Local Search (LS)

An LS performs a search within the neighborhood of the chromosomes in the evolutionary process in order to further improve fitness. Within the LS, the fitness function determines the largest cause of costs within a chromosome in the form of a critical operational pair. This operational pair can be determined in the decoded solution by the corresponding start and end times and the cost rates. The high costs may be caused by the costs of tied-up capital due to long waiting times between the operational pair or between the completion deadline and the last operation within a job. Furthermore, penalty costs for impermissibilities (see Crossover) may also cause the high level of costs. The time domain in which the critical operational pair is situated is completely removed from the production plan for the machines in question. An attempt is made to reschedule the removed operations so that the high costs no longer arise and the chromosome's fitness is consequently improved. The neighborhood is therefore defined as the number of all permissible allocations of the removed operations in the considered time domain. If no improvement in the fitness of the critical operational pair can be achieved after a specified number of attempts (Limit_noAssign), the local search is continued with the operational pair which causes the next highest level of costs. Furthermore, there is a limit on the number of attempts a chromosome may be searched through without any improvement in fitness being produced (Limit_noimprove) before the next chromosome is considered. The local search ends once a specified number of chromosomes to be searched through has been reached (Limit_tabuCounter).

H. Reinsertion Scheme

The reinsertion scheme governs which chromosomes are removed from the current population, and which chromosomes that have been newly created in the evolutionary process will be incorporated into the population. In this MA, elitist reinsertion is used. The chromosomes in the population with the worst fitness are replaced by the chromosomes created in the evolutionary process if the latter has a higher level of fitness.

V. Computational Results

The goal of the following evaluation is to validate the EFJSP. Therefore, the potential for lowering costs by the consideration of energy costs in production planning is proved. For that the performance of the MA is evaluated first. Since the MA's practical application has a higher priority than the performance to compute the best possible solution, especially a short computing time is necessary. It is not the MA's purpose to reach or exceed the performance of existing state-of-the-art heuristic approaches solving the pure FJSP. The MA should be capable of solving practical problem sizes with an adequate performance.

A. Experimental Setup

13 test instances were generated for validating the EFJSP and the MA that had been developed. Since the literature on the subject only includes test instances for the conventional FJSP and these are not applicable to the EFJSP, it was necessary to generate new test instances. The setup of the test instances is shown in Table II.

TABLE II
SETUP OF THE TEST INSTANCES USED

Name	# of jobs	# of machines	# of operations	# of energy phases	# of periods	Utilization (%)
TI01	2	3	3	3	32	28.1
TI02	4	6	3	3	32	28.6
TI03	6	6	3	3	32	42.7
TI04	8	9	3	3	32	38.9
TI05	10	9	3	3	32	48.6
TI06	12	9	3	3	32	56.9
TI07	12	6	3	3	64	42.7
TI08	12	9	3	3	64	28.4
TI09	14	6	3	3	64	49.2
TI10	14	9	3	3	64	32.8
TI11	16	9	3	3	64	37.7
TI12	18	9	3	3	64	42.4
TI13	20	9	3	3	64	47.6

The test instances are based on the assumption of job-shop production in which the jobs have to undergo three different process steps on three different types of machine. This produces a constant number of three operations for all the test instances. The number of energy phases is also constant since in the case of jobs with similar work content and similar processing times it is assumed that the energy profile over time will be basically similar and will mainly differ in terms of its overall level. The test instances differ in terms of the number of jobs that are considered, the machines that are available to undertake processing and the capacity utilization. The capacity utilization U of a test instance is determined by (24).

$$U = \frac{\sum_i \sum_s p_{i,s}}{T \cdot K} \tag{24}$$

The machines considered in the test instances are assigned to one of the three machine types, and each of them can consequently process one specific operation. Table III shows the assignment of the machines to the machine types and the operations.

TABLE III
ASSIGNMENT OF THE MACHINES TO MACHINE TYPES AND OPERATIONS

	Machine type 1	Machine type 2	Machine type 3
Operation 1	k1; k2; k3		
Operation 2		k4; k5; k6	
Operation 3			k7; k8; k9

TABLE IV
MA SETTING PARAMETERS USED

Parameters	Description	Value
TimeDuration	Time duration in minutes	15
PopSize (μ)	Population size	50
mue (ξ)	Number of randomly chosen chromosomes for the mating pool (ξ)	2
Limit_CO	Number of Crossover tests	10
Limit_shiftingKap	Number of Repair tests (procedure 1)	20
Limit_repair	Number of Repair tests (procedure 2)	20
Mutate_rate	Mutation probability	0.6
Limit_tabuCounter	Number of considered chromosomes at LS	2
Limit_noAssign	Number of allocation attempts for critical operation pair at LS	10
Limit_noImprove	Number of improving attempts per chromosome at LS	2

The periods that are considered in a test instance are based on a one- or two-shift model, and they therefore produce a planning horizon of one operational calendar day with 32 or 64 periods. For example, one shift is considered in test instance TI05. Using a period length of 15 minutes and a shift length of eight hours produces 32 periods. The job processing times, energy consumption levels, and costs variables used in the test instances are based on a normal distribution of random numbers. The mean values used for generating the normal distribution of random numbers and the price per kW are based on empirically ascertained values in the mechanical engineering sector. 60 €/kW is assumed to be the energy price *LP1* per kilowatt. Since most manufactures in the mechanical engineering sector are SMEs and not energy-intensive, it is assumed that there is no threshold regarding the power peak. Therefore, LP2 is set to an inaccessible value of 100,000 kW. The average processing time for an operation is one hour. The energy consumption levels which occur during the processing of the operations within the respective energy phases fluctuate between 60 and 180 kW. In test instances TI01 to TI06, the jobs which are considered must be completed by period 32, and similarly in test instances TI07 to TI13, the jobs considered must be completed by period 64.

The values shown in Table IV for the MA setting parameters which are described in section 4 were used for the calculations that were undertaken. In tests that were carried out and statistically analyzed, this combination of values produced the best results.

B. Computational Results and Comparisons

All tests were carried out on a computer with an Intel Core i7 3.5 Ghz processor and 16 GB ram. In the literature, there is no similar heuristic against which the MA for solving the EFJSP could be validated. The test instances were therefore initially solved using a commercial solver which can produce optimal solutions. Reference solutions could therefore be produced which allowed the quality of the MA solution to be determined. In order to do this, the EFJSP was modeled in the GAMS modeling language, and it was solved using the BARON version 14.4.0 solver. BARON is a branch-and-cut solver for global optimization [38]. The solver is suitable for solving mixed-integer, non-linear problems (minlp) such as the EFJSP.

Each test instance was solved once using the BARON solver, with the solver stopping in each case after a maximum computing time of 24 hours. Except for test instance TI01, BARON was unable to find an optimal solution for any further test instances after 24 hours. In Table V column BR shows the best solutions found by BARON for all test instances. Column BR_time_best shows the required computing time to find each of the best solution. During the remaining computing time, BARON did not find a better solution. BARON found the optimal solution for test instance TI01 after 22.2 seconds. The solution in Table V is therefore marked with an asterisk. After 24 hours, no solution could be found for test instance TI10. Column dev_LB shows the percentage deviation from the lower bound found by BARON.

TABLE V
COMPARISON OF THE BARON AND MA SOLUTIONS

Name	i	k	t	BR	BR_Time_best (s)	dev_LB (%)	MA_best	dev_BR (%)	MA_avg	dev_BR (%)
TI01	2	3	32	14307	22.2	0.00	14307	0.00	14307	0.00
TI02	4	6	32	20621	718.4	6.69	21371	3.64	22135	7.34
TI03	6	6	32	25789	1290.3	13.83	26756	3.75	28273	9.63
TI04	8	9	32	39059	207.6	18.82	36549	-6.43	39162	0.26
TI05	10	9	32	41462	334.2	11.86	43428	4.74	46476	12.09
TI06	12	9	32	44102	78540.0	7.39	49283	11.75	51084	15.83
TI07	12	6	64	37362	717.1	24.33	35501	-4.98	36713	-1.74
TI08	12	9	64	51078	56261.0	13.16	50020	-2.07	51819	1.45
TI09	14	6	64	37623	901.2	22.09	37079	-1.45	38627	2.67
TI10	14	9	64	53337	61200.0	16.85	52048	-2.42	53154	-0.34
TI11	16	9	64	52221	745.3	12.88	53220	1.91	54599	4.55
TI12	18	9	64	54025	1113.2	14.51	54667	1.19	56365	4.33
TI13	20	9	64	-	-	-	54595	-	57639	-
									avg.	+4.67

The MA was implemented in JAVA. Twenty runs with a computing time of 15 minutes were executed for each test instance. The MA_best column of Table V shows the best solution found from the twenty runs. Column MA_avg shows the average solution which is calculated from the best solution found in each of the twenty runs. Columns dev_BR each show the percentage deviation between the best solution produced by the MA and the solution found by BARON, or between the average MA solution and the solution produced by BARON.

The MA found the optimum in all of the twenty TI01 processing runs. The MA is consequently able to find the optimum solution for smaller test instances. For test instances TI07, TI08, TI09 and TI10 the MA was able to find better solutions than BARON. Across all test instances, the average deviation from the solutions found by BARON is 4.67%. The MA is capable to solve the EFJSP with an adequate quality of solution within a short computing time for a practical application.

Fig. 3 shows the convergence curves for the best and the average solution produced by the MA based on the example of test instance TI04. It can be seen that the objective function value decreases very sharply within the first 100 generations. Then, in the following generations, the objective function value decreases only slightly. Only the first 1500 generations are shown in Fig. 3. The subsequent 650 generations produced no change in the objective function variable and they are therefore not shown in Fig. 3. To compute the first 1500 generations, the MA needed 619 seconds. It should be noted that the MA can generate good solutions within a small number of generations.

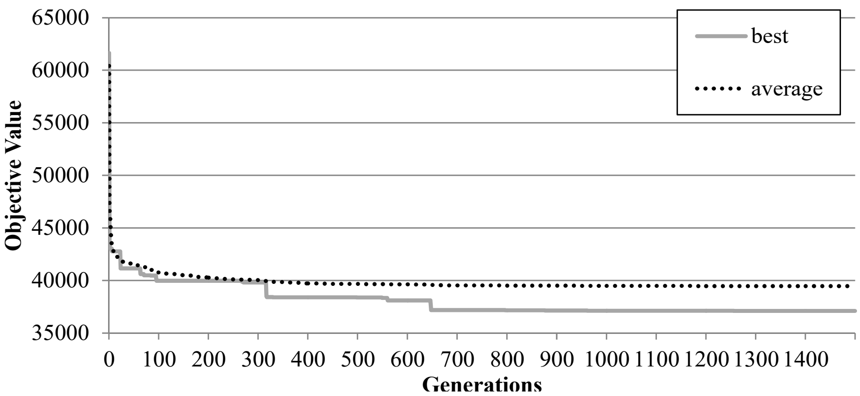

Fig. 3 MA convergence curve for T104

TABLE VI
RESULTS FOR TEST INSTANCE TI04 WITH AND WITHOUT CONSIDERATION OF ENERGY

Run	Without energy consideration				With energy consideration			
	Power peak	Energy costs	Logistics costs	Total costs	Power peak	Energy costs	Logistics costs	Total costs
1	530	31800	12760	44560	418	25080	12839	37919
2	690	41400	12793	54193	441	26460	12905	39365
3	600	36000	12752	48752	451	27060	12844	39904
4	600	36000	12776	48776	446	26760	13062	39822
5	691	41460	12783	54243	446	26760	12847	39607
6	607	36420	12740	49160	393	23580	12969	36549
7	565	33900	12816	46716	434	26040	12909	38949
8	703	42180	12803	54983	475	28500	12946	41446
9	464	27840	12843	40683	447	26820	12896	39716
10	581	34860	12867	47727	419	25140	12897	38037
11	633	37980	12862	50842	427	25620	12931	38551
12	716	42960	12836	55796	458	27480	13086	40566
13	530	31800	12779	44579	409	24540	12965	37505
14	608	36480	12804	49284	428	25680	12981	38661
15	557	33420	12807	46227	428	25680	13032	38712
16	602	36120	12823	48943	427	25620	12784	38404
17	602	36120	12805	48925	481	28860	12930	41790
18	493	29580	12824	42404	464	27840	13070	40910
19	575	34500	12791	47291	419	25140	12973	38113
20	628	37680	12826	50506	431	25860	12854	38714
avg.	589	35925	12804	48729	437	26226	12936	39162
avg_dev. (%)	+36.9	+36.9	-1.0	+24.4				

C. Cost Minimizing Potential

In order to validate the potential for lowering costs that is provided by consideration of the energy costs which arise owing to the power peak, test instance TI04 was solved again with the MA without taking account of the power peak in the objective function. This involved only optimizing the logistic decision variables within the production plan derived from the EFJSP. The energy consumption profile that was produced was recorded separately. The results of both solutions of the test instance (with and without consideration of energy) are shown in Table VI.

The failure to take energy costs into account in the objective function leads to much higher power peaks, and consequently also to higher energy costs in the individual runs. The latter rose on average by 36.9% compared to when energy costs were taken into account in the objective function. On the other hand, the logistics costs fall by only 1.0%. Based on the assumed cost levels, the average increase in overall costs when energy costs are not taken into account is 24.4%. The achievable costs-saving potential naturally always depends on the energy and logistics costs which are used as a basis, and these must be individually determined for each specific application. As already stated above, an attempt has been made to use realistic cost levels for the validation that is described here. Overall, it can be stated that the EFJSP can lead to a reduction in the costs which arise in relation to a production plan. The production plans produced by the best respective runs of test instance TI04 are shown below. Fig. 4 shows the production plan from run 6 with consideration of energy costs, and Fig. 5 shows the production plan from run 9 without consideration of energy costs.

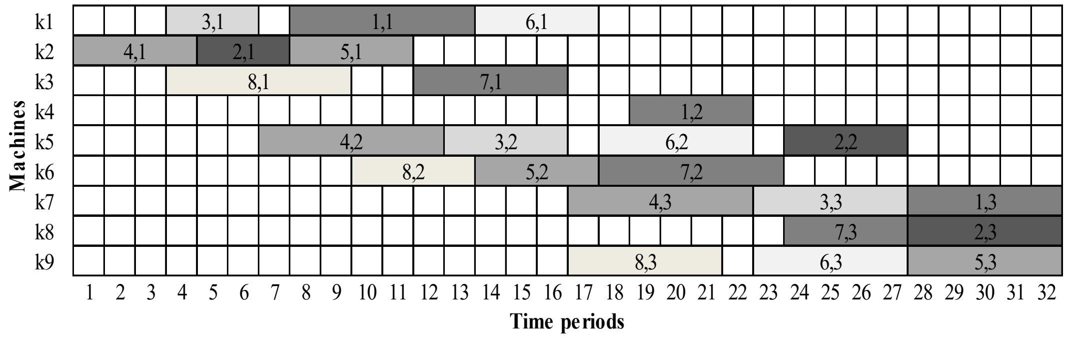

Fig. 4 Production plan from run 6 of TI04 with consideration of energy costs

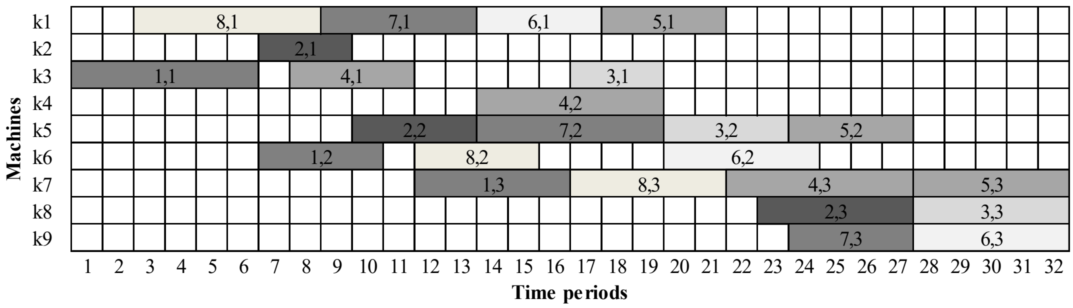

Fig. 5 Production plan from run 9 of TI04 without consideration of energy costs

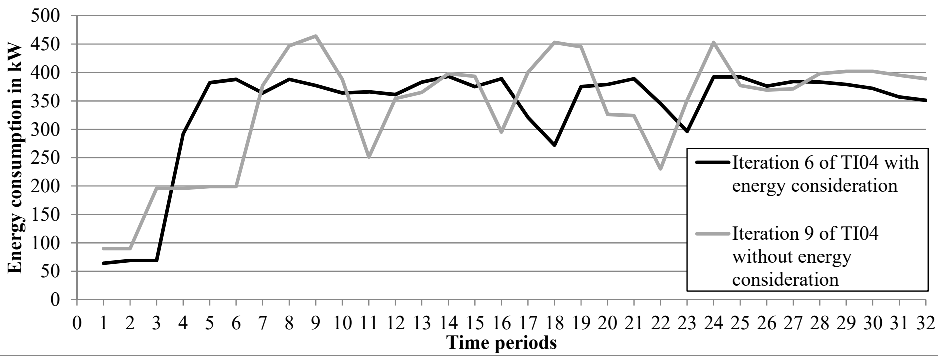

Fig. 6 Comparison of the energy consumption profiles

Fig. 6 shows the energy consumption profiles arising from the two production plans. It can be seen that consideration of energy costs reduces the power peak. Furthermore, the energy consumption profile is much more even, and it fluctuates less than the energy consumption profile when energy costs are not considered. This leads to reduced energy costs that can be seen in Table VI (with energy consideration) when energy costs are considered in the EFJSP objective function.

It can be seen that the gap between the individual operations of the respective jobs in Fig. 4 is larger than in Fig. 5. As shown in Table VII, this leads to an increase in the waiting times between the operations, and consequently also to an increase in the throughput time of the jobs. Comparison of the two production plans shows that the throughput times increase by 28 periods when energy costs are considered.

TABLE VII
COMPARISON OF THE WAITING TIMES AND THROUGHPUT TIME

Job	Without energy consideration (Run 9 of TI04)		With energy consideration (Run 6 of TI04)	
	Waiting Time	through put time	Waiting Time	throughput time
1	1	16	10	25
2	9	21	16	28
3	4	16	12	24
4	4	20	6	22
5	2	15	13	25
6	5	19	0	14
7	4	19	1	16
8	4	19	3	18
total	33	145	61	172
dev. (%)			+84.8	+18.6

VI. SUMMARY AND CONCLUSION

In this article, the FJSP was extended to include consideration of energy costs which arise owing to the power peak. Moreover further decision variables, such as WIP and throughput time, were incorporated into the objective function. In addition, cost values were applied to the individual decision variables as a way of weighting the decision variables. Therefore, a production plan was enabled to be simultaneously optimized regarding the real arising energy and logistics costs. An MA was used to solve the EFJSP. The MA was enhanced with a local search and repair procedures. Afterwards, the MA was tested using test instances that had been generated, and was validated against a commercial branch-and-cut solver. Here, the MA achieved comparably good results while using significantly less computing time. The MA can therefore be said to be a suitable heuristic for practical application of the EFJSP. The EFJSP was compared with the conventional FJSP which does not consider energy costs. With this the positive effect of consider energy costs within production planning was demonstrated. The developed MA provides an inexpensive means for even small and medium-sized companies to exploit the existing cost-saving potential. In further research, the EFJSP could be extended by considering energy costs depending on the time of the day.

ACKNOWLEDGMENT

The authors declare that they have no conflict of interest.

REFERENCES

[1] Pinedo M (2012) Scheduling: Theory, Algorithms and Systems. 4th edn. Springer, New York, Heidelberg, London.
[2] Yazdani M, Amiri M, Zandieh M (2010) Flexible Job-shop Scheduling with Parallel Variable Neighborhood Search Algorithm. Expert Syst Appl 37:678–687.
[3] Garey M R, Johnson D S, Sethi R (1976) The complexity of flowshop and jobshop scheduling. Math Oper Res 1:117–129.
[4] BMWI - German Federal Ministry for Economic Affairs and Energy (2014) Entwicklung von Energiepreisen und Preisindizes zu nominalen Preisen Deutschland. http://www.bmwi.de/BMWi/Redaktion/Binaer/Energiedaten/energiepreise-und-energiekosten1-entwicklung-energiepreise-preisindizes,property=blob,bereich=bmwi2012,sprache=de,rwb=true.xls 12.01.2015http://www.bmwi.de/BMWi/Redaktion/Binaer/Energiedaten/energiepreise-und-energiekosten1-entwicklung-energiepreise preisindizes,property=blob,bereich=bmwi2012,sprache=de,rwb=true.xls. Accessed 07 April 2016.
[5] Dai M, Tang D, Giret A, Salido M A, Li W D (2013) Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. Robot Cim-Int Manuf 29:418-429.
[6] Luo H, Du B, Huang G Q, Chen H, Li X (2013) Hybrid flow shop scheduling considering machine electricity consumption cost. Int J Prod Econ 146: 423-439.
[7] Moon J-Y, Park J (2014) Smart production scheduling with time-dependent and machine-dependent electricity cost by considering distributed energy resources and energy storage. Int J Prod Res 52:3922-3939.
[8] Fink R (2013) A priority rule based multi criteria scheduling framework for energy efficiency aware detailed production planning. 39th Annual Conference of the IEEE Industrial-Electronics-Society, Vienna, 39:7508-7512.
[9] Wiendahl H-P (1995) Load-Oriented Manufacturing Control. Springer, Berlin, Heidelberg, New York.
[10] Nyhuis P, Wiendahl H-P (2009) Fundamentals of Production Logistics. Springer, Berlin, Heidelberg.
[11] Gutenberg E (1951) Grundlagen der Betriebswirtschaftslehre. Volume 1: Die Produktion. 1st edn. Berlin.
[12] Brucker P, Schlie R (1990) Job-shop scheduling with multi-purpose machines. Computing 45:369–375.
[13] Pezzella F, Morganti G, Ciaschetti G (2008) A genetic algorithm for the flexible job-shop scheduling problem. Comput Oper Res 35:3202–3212.
[14] Wang L, Zhou G, Xu Y, Wang S, Liu M (2012) An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. Int J Adv Manuf Technol 60:303–315.
[15] Xia W, Wu Z (2005) An effective hybrid optimization approach for multiobjective flexible job-shop scheduling problem. Comput Ind Eng 48:409–425.
[16] Yuan Y, Xu H, Yang J (2013) A hybrid harmony search algorithm for the flexible job shop scheduling problem. Appl Soft Comput 13:3259–3272.
[17] Brandimarte P (1993) Routing and scheduling in a flexible job shop by tabu search. Ann Oper Res 41:157–183.
[18] Paulli J (1995) A hierarchical approach for the FMS scheduling problem. Eur J Oper Res 86:32–42.
[19] Dauzére-Pérés S, Paulli J (1997) An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. Ann Oper Res 70:281–306.
[20] Mastrolilli M, Gambardella L M (2000) Effective neighborhood functions for the flexible job shop problem. J Sched 3:3–20.
[21] Chen H, Ihlow J, Lehmann C (1999) Agenetic algorithm for flexible Job-shop scheduling. IEEE Int conf robot, Detroit, 1120–1125.
[22] Kacem I, Hammadi S, Borne P (2002) Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE T Syst Man Cyb 32:1–13.
[23] Zhang H P, Gen M (2005) Multistage-based genetic algorithm for flexible jobshop scheduling problem. Journal of Complexity International 11:223–232.
[24] Gao J, Sun L, Gen M (2008) A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling

problems. Comput Oper Res 35:2892–2907.
[25] Al-Hinai N, ElMekkawy T Y (2011) An Efficient Hybridized Genetic Algorithm Architecture for the Flexible Job Shop Scheduling Problem. Flexible Services and Manufacturing Journal 23:64–85.
[26] Raeesi M N, Kobti Z (2012) A memetic algorithm for job shop scheduling using a critical-path-based local search heuristic. Memetic Computing 4:231–245.
[27] Gutiérrez C, García-Magariño I (2011) Modular design of a hybrid genetic algorithm for a flexible job-shop scheduling problem. Knowl-Based Syst 24:102–112.
[28] Jiang J, Wen M, Ma K, Long X, Li J (2011) Hybrid Genetic Algorithm for Flexible Job-Shop Scheduling with Multi-objective. Journal of Information & Computational Science 8:2197-2205.
[29] Rager M (2008) Energieorientierte Produktionsplanung. GWV Fachverlage, Wiesbaden.
[30] Rager M, Gahm C, Denz F (2015) Energy-oriented scheduling based on Evolutionary Algorithms. Comput Oper Res 54:218-231.
[31] Fang K, Uhan N, Zhao F, Sutherland J W (2011) A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. J Manuf Syst 30:234-240.
[32] Bruzzone A A G, Anghinolfi D, Paolucci M, Tonellia F (2012) Energy-aware scheduling for improving manufacturing process sustainability: a mathematical model for flexible flow shops. CIRP Ann-Manuf Techn 61:459-462.
[33] Wiehndahl H-P (2014) Betriebsorganisation für Ingenieure. 8th edn. Hanser, München, Wien.
[34] Weinert N, Chiotellis S, Seliger G (2011) Methodology for planning and operating energy-efficient production systems. CIRP Ann-Manuf Techn 60:41-44.
[35] Bierwirth C (1995) A generalized permutation approach to job shop scheduling with genetic algorithms. OR Spektrum 17:87-92.
[36] Eiben A E, Smith J E (2003) Introduction to Evolutionary Computing. Springer, Berlin, Heidelberg.
[37] Zäpfel G, Braune R, Bögl M (2010) Metaheuristic Search Concepts. Springer, Berlin, Heidelberg.
[38] Tawarmalani M, Sahinidis N V (2005) A polyhedral branch-and-cut approach to global optimization. Math Program 103:225-249.