# Multi-objective genetic algorithm for energy-efficient job shop scheduling

Gökan May[a]*, Bojan Stahl[a], Marco Taisch[a] and Vittal Prabhu[b]

[a]*Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Milano, Italy;* [b]*Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA, USA*

The paper investigates the effects of production scheduling policies aimed towards improving productive and environmental performances in a job shop system. A green genetic algorithm allows the assessment of multi-objective problems related to sustainability. Two main considerations have emerged from the application of the algorithm. First, the algorithm is able to achieve a semi-optimal makespan similar to that obtained by the best of other methods but with a significantly lower total energy consumption. Second, the study demonstrated that the worthless energy consumption can be reduced significantly by employing complex energy-efficient machine behaviour policies.

**Keywords:** energy efficiency; job shop; scheduling; genetic algorithms; machine control policies; sustainable manufacturing

## 1. Introduction

Energy consumption has become a major issue in our society, the world's energy demand has doubled over the last 40 years and is estimated to double once again by 2030. The manufacturing sector is responsible for 33% of the total con-sumption and is accountable for over 38% of CO2 emissions (IEA 2008). It is therefore important to reduce this energy consumption by means of energy-efficient measures both for economic and environmental reasons (May et al. 2013).

The significant potential of increasing energy efficiency is therefore found in the manufacturing industry. For this reason, several articles investigated approaches to increase energy efficiency in various levels of manufacturing (Devoldere et al. 2007; Jin et al. 2009; Prindle 2010; Sun et al. 2011; Hu et al. 2012). One of the several ways of achieving energy efficiency in a discrete manufacturing environment is to address the production scheduling function by considering green metrics alongside traditional performance indicators. Using this approach, it is possible to achieve non-negligible improvements with virtually zero costs.

Although since the 1950s, many scheduling algorithms have been developed, the field of green scheduling is rela-tively new. There is the need to develop new green objectives in order to change the point of view of optimisation and to evaluate the trade-off between traditional economic performance indicators and new green performance metrics, particularly when these objectives are in conflict with each other.

The first two papers related to manufacturing scheduling while dealing with green objectives were written by He, Liu, and Cao (2005) and by Subai, Baptiste, and Niel (2006). Later, Gutowski, Dahmus, and Thiriez (2006) showed that for different machine tools, the total power required is the sum of a constant term $P_{idle}$, which is required to run the basic functions of the machine tool, such as keeping the pumps for the oil and the coolant up and running. Additionally, Drake et al. (2006) showed a similar behaviour, experimentally calculating the power requirements of the various components of a machine tool by connecting a power-measuring device to the machine's main power box. Dietmair, Verl, and Eberspaecher (2010) modelled the power consumption of a machine tool, separating its behaviour in different operational states, based on the state of the subcomponents and therefore with different power consumptions.

He and Liu (2010), proposing an upgrade of their previous paper (He, Liu, and Cao 2005), integrated in their approach, the evaluation of other environmental impacts, such as the production of solid, liquid, gaseous and other kind of wastes. Prabhu, Jeon, and Taisch (2013) modelled energy behaviours of discrete manufacturing systems using a simulation tool that links machine-level energy control policies and production control policies.

In the more specific literature about energy-efficient scheduling using green genetic algorithms (GGAs), there are quite a few research attempts. For instance, the first paper proposing some dispatching rules was published in 2007 by

*Corresponding author. Email: gokan.may@polimi.it

Mouzon, Yildirim, and Twomey (2007). Later, Yildirim and Mouzon (2011) published their third paper on the green scheduling subject: using the same case as the second paper (Mouzon and Yildirim 2008), they implemented a multiple objective genetic algorithm in order to obtain the Pareto front with respect to the total completion time and total energy consumption (TEC) objectives.

Fang et al. (2011) developed yet another flow shop model with three different objectives: makespan (MKS), peak total power consumption and carbon footprint. Chen et al. (2013) analysed the effect of switching the machines off and on again in a production system with M machines placed in series, with finished buffers between them. In a recent article, Liu and Huang (2014) studied the power consumption and carbon footprints in batch-processing and parallel-processing machines by use of a genetic algorithm. Finally, Dai et al. (2013) addressed the scheduling issue of a flexible flow shop and proposed an energy-efficient model. They found significant energy efficiency improvement potential in their energy-aware model.

From the review of the pertinent literature, we derived the following conclusions:

- Although the genetic algorithms have been extensively studied, it is generally impractical to establish that one implementation of the algorithm is better than the others because of their heuristic nature.
- Instead, from the green manufacturing point of view and in particular for green scheduling, the research is still at a primordial state. All analysed papers contain simple cases, with just a few number of jobs processed on one machine or flow shops with a couple of machines in series.
- The last consideration is about the difficulties in evaluating energetic states and the environmental impacts associated with the processes. The only states which have been studied and practically implemented in the papers are ‘on’, ‘idle’ and ‘off’, although theoretical studies highlighted the presence of other states, such as ‘standby’.

Based on the above-mentioned review and identified gaps, this study focuses on the production scheduling function of a job shop environment. This function has been researched through the years from the productivity point of view, but has not been investigated thoroughly from the energetic point of view.

This work therefore intends to investigate the effects of production scheduling policies, aimed towards improving energy efficiency on the performances of a job shop manufacturing system. Hence, a new GGA which is able to consider objectives related to productivity and energy consumption is developed, giving as a result, a series of different Pareto front solutions, which allow the possibility to assess the energetic consumption of different literature test problems with respect to the different policies.

The paper is organised as follows. Section 2 discusses energetic machine behaviour, energy-efficient scheduling policies and performance indicators for energy-efficient manufacturing systems. The new genetic algorithm is illustrated in Section 3, tested in Section 4 and validated through predefined methods in Section 5. Section 6 applies the algorithm for a quasi-real job shop problem. The conclusion in Section 7 ends highlighting the results of the study, implications for research and practice, and drawing future outlooks for research.

## 2. Modelling energy-efficient job shop

### 2.1 *Evaluation of the operational and energetic states of a machine tool*

A machine can be modelled in various ways. For the underlying research purpose, we build our model based on states of the machine. A traditional view is based on discrete operational states of a machine:

- Idle: the machine is active and ready to work. There are different reasons for the machine to be in this state: it could be starving, which means it is waiting for a job; it could be blocked, which means that it is waiting to release the job it has already processed.
- Down: the machine has failed in one among different ways that affect the production.
- Set-up: the machine is changing the tool in order to process a new part.
- Maintenance: the machine is being repaired or undergoing planned maintenance operations.
- Working: the machine is processing a job.

The working state has a distinctive role compared to the other states. It is responsible for the creation of direct value, which can be associated to the single workpiece. The state set-up does not directly add value to the WIP, but its contribution is still needed to ensure that the machines are able to work properly with the right tools and without failures. Therefore, it is considered indirectly valuable. All the other states (idle, blocked and down) are not contributing in any form to value generation and should therefore be avoided or minimised.

A machine can be modelled also from an energetic perspective opposing to the traditional kinematics perspective described above:

- Off: machine is off and no power is required.
- Standby: machine has most of the components switched off and the machine is not ready to process parts. Only some components are fed by power in order to speed up activation time.
- Idle: machine is ready to process parts and significant amount of power consumption is required to run all the auxiliary equipment.
- Set-up: machine is changing the tool and additionally to the idle power, the energy consumed for actual changing of the tool is required.
- Working: machine is actually processing the part and maximum power is needed to process the part.

Considering the energy-consuming transition of ramp-down and ramp-up, Figure 1 displays the energetic states diagram.

The states analysed from the productive and energetic perspective do not have one-to-one relation. E.g. the idle state from the traditional perspective can be seen threefold from an energetic perspective: off, standby or idle. From the productivity point of view, there is no difference detected for the current situation of the production system, but the energetic values for the three energetic states off, standby and idle vary significantly. The different relationships between productivity and energetic states are presented in Figure 2.

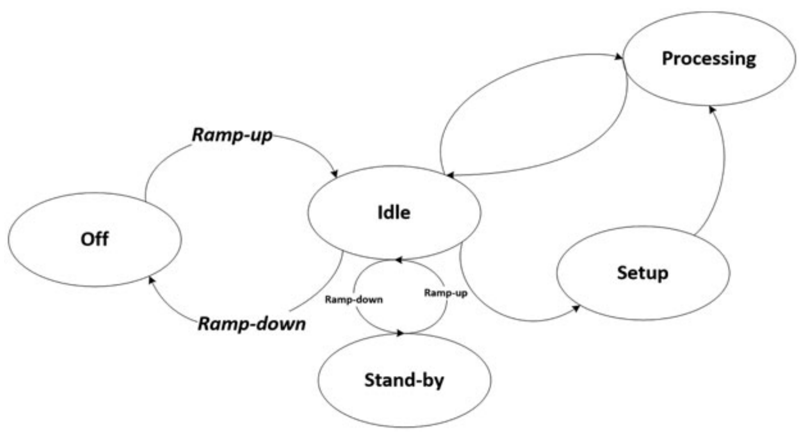



Figure 1. Energetic states diagram.

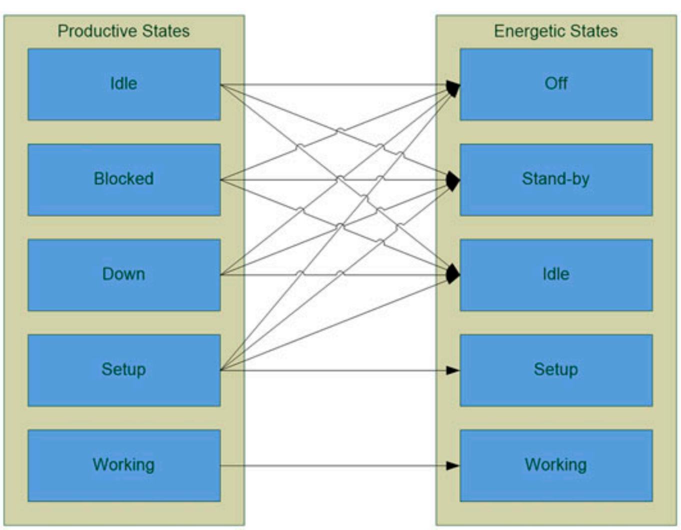



Figure 2. Links between productivity states and energetic states.

### 2.2 *Behaviour policies of machine tools*

In order to assess the benefits of managing the different energetic states of machines to reduce the consumption of electricity of the productive system, in this section, we introduce different machine behaviour polices that are then applied using genetic algorithm.

The first policy is inspired by the business-as-usual method in manufacturing firms. Since we are working with batch of jobs, we consider that all the machines are switched on when the first operation is started and switched off when the last operation is finished.

The second policy is similar to the first one with the difference being the machines are individually switched on when required to process their first operation and switched off upon conclusion of the last operation.

In the third policy, we use the concept of different energetic states developed in the previous section. The machines are still individually switched on at their first operation and off at their last, but in addition, there is the possibility to switch them on and off when they are idle in the middle of the production batch. When the TEC of switching the machine off and on again for a certain time period t is less than the energy consumption during the idle state, it is favourable to apply this policy. Inequality (1) represents the trade-off between the energy used for a start-up cycle, while the energy consumed in the idle state represents the *constraint*.

$$P_{\text{on-off}} \cdot t_{\text{on-off}} < P_{\text{idle}} \cdot t_{\text{idle}} \tag{1}$$

In the fourth policy, the standby state is added. Depending on the least energy consumed, it is possible to decide to (i) switch the machine on and off again, (ii) switch the machine to standby or (iii) leave the machine in idle state. The inequalities below describe when a certain behaviour is to be chosen, based on the energy consumption.

The machines should be switched off and then on if:

$$\begin{gathered} P_{\text{on-off}} \cdot t_{\text{on-off}} < P_{\text{idle}} \cdot t_{\text{idle}} \\ \& \\ P_{\text{on-off}} \cdot t_{\text{on-off}} < P_{\text{standby-idle}} \cdot t_{\text{standby-idle}} + P_{\text{standby}} \cdot t_{\text{standby}} \end{gathered} \tag{2}$$

The machines should be put in standby mode if:

$$\begin{gathered} P_{\text{standby-idle}} \cdot t_{\text{standby-idle}} + P_{\text{standby}} \cdot t_{\text{standby}} < P_{\text{on-off}} \cdot t_{\text{on-off}} \\ \& \\ P_{\text{standby-idle}} \cdot t_{\text{standby-idle}} + P_{\text{standby}} \cdot t_{\text{standby}} < P_{\text{idle}} \cdot t_{\text{idle}} \end{gathered} \tag{3}$$

The machines, instead, should be kept idle if:

$$\begin{gathered} P_{\text{idle}} \cdot t_{\text{idle}} < P_{\text{on-off}} \cdot t_{\text{on-off}} \\ \& \\ P_{\text{idle}} \cdot t_{\text{idle}} < P_{\text{standby-idle}} \cdot t_{\text{standby-idle}} + P_{\text{standby}} \cdot t_{\text{standby}} \end{gathered} \tag{4}$$

where $P_{\text{idle}}$, Power required to stay in idle state [kW]; $t_{\text{idle}}$, Time passed in idle state [s]; $P_{\text{on-off}}$, Power required to switch machine off and again on [kW]; $t_{\text{on-off}}$, Time passed to switch machine off and again on [s]; $P_{\text{standby-idle}}$, Power required to switch machine between idle and standby state [kW]; $t_{\text{standby-idle}}$, Time passed to switch machine between idle and standby state [s]; $P_{\text{standby}}$, Power required in standby state [kw]; $t_{\text{standby}}$, Time passed in standby state [s].

### 2.3 *Performance indicators*

An important indicator to represent the behaviour of a production system is the MKS, which expresses the ability of the manufacturing system to complete a batch of jobs in a defined interval of time. Usually, a lower MKS implies a higher utilisation of the machines and lower cycle times.

The MKS is the standard objective that is used to assess the performances of an algorithm in resolving a manufacturing scheduling problem. It is defined as the difference between the completion date of the last job and the starting date of the first job:

$$\text{MKS} = \max(\text{c}_j) - \min(\text{s}_j) \tag{5}$$

$c_j$, Completion date of job $j$ [s]; $s_j$, Starting date of job $j$ [s].

Taking environmental perspective into account, the TEC indicator expresses the total amount of energy consumed in order to produce a specific batch of jobs, assuming that the only form of energy consumed is electrical energy. TEC is another important indicator used in this study, and is calculated as below:

$$\mathrm{TEC} = \sum_{k=1}^{m}\sum_{j=1}^{n} P_{\mathrm{work},k,j}\cdot t_{\mathrm{work},k,j} + P_{\mathrm{idle},k}\cdot t_{\mathrm{idle},k} + P_{\mathrm{setup},k,j-1,j}\cdot t_{\mathrm{setup},k,j-1,j} + P_{\mathrm{standby},k}\cdot t_{\mathrm{standby},k} + P_{\mathrm{ramp-up},k}\cdot (n_{\mathrm{ramp-up,off},k} \cdot t_{\mathrm{ramp-up,off},k} + n_{\mathrm{ramp-up,standby},k}\cdot t_{\mathrm{ramp-up,standby},k}) \tag{6}$$

where $P_{\mathrm{work},k,j}$, Power required by machine $k$ to process job j [kW]; $t_{\mathrm{work},k,j}$, Time required by machine $k$ to process job $j$ [$s$]; $P_{\mathrm{idle},k}$, Power required by machine $k$ to stay in idle state [kW]; $t_{\mathrm{idle},k}$, Time passed by machine $k$ in idle state [$s$]; $P_{\mathrm{setup},k,j-1,j}$, Power required by machine $k$ to perform set-up from the $j-1$th job to $j$th job [$s$]; $t_{\mathrm{setup},k,j-1,j}$, Time required by machine $k$ to perform set-up from the $j-1$th job to $j$th job [$s$]; $P_{\mathrm{standby},k}$, Power required by machine $k$ to stay in standby state [kW]; $t_{\mathrm{standby},k}$, Time passed by machine $k$ in standby state [$s$]; $P_{\mathrm{ramp-up},k}$, Power required by machine $k$ to switch from off or standby states to idle state [kW]; $n_{\mathrm{ramp-up,off},k}$, Number of transitions from off to idle state performed by machine $k$; $t_{\mathrm{ramp-up,off},k}$, Time required by machine $k$ to switch from off to idle state [$s$]; $n_{\mathrm{ramp-up,standby},k}$, Number of transitions from standby to idle state performed by machine $k$; $t_{\mathrm{ramp-up,standby},k}$, Time required by machine $k$ to switch from standby to idle state [$s$].

For the underlying research purpose, we have built upon the concept of the TEC and have created a new environmental performance indicator called worthless energy consumption (WEC).

$$\mathrm{WEC} = \sum_{k=1}^{m}\sum_{j=1}^{n} P_{\mathrm{idle},k}\cdot t_{\mathrm{idle},k} + P_{\mathrm{standby},k}\cdot t_{\mathrm{standby},k} + P_{\mathrm{ramp-up},k}\cdot (n_{\mathrm{ramp-up,off},k} \cdot t_{\mathrm{ramp-up,off},k} + n_{\mathrm{ramp-up,standby},k}\cdot t_{\mathrm{ramp-up,standby},k}) \tag{7}$$

The main idea is to study the impact of different energy policy behaviours on productive and environmental performances. It is of special practical relevance, if and when the adoption of a certain energy policy may lead to a substantial performance increase or when improvements are marginal and negligible.

## 3. The multi-objective GGA

The new GGA has been built by merging, tweaking and upgrading two of the current state-of-the-art GAs [NSGA-II (Deb et al. 2002) and SPEA-II (Zitzler, Laumanns, and Thiele 2001)] in order to create a new algorithm able to cope with a multi-objective job shop optimisation problem with both productivity and environmental objectives.

The basic structure of the algorithm is represented in Figure 3: at first an initial population $P_0$ is created, then this population undergoes the evolution process to build the new population $Q_t$. The newly created population and the old one (which is called $P_0$ in the first iteration and $P_t$ in the subsequent iteration) are compared and the best individuals between them are selected to compose the new population $P_{t+1}$ (for this reason, this implementation has a strong component of elitism, which has been proven to increase the quality of the solutions obtained with genetic algorithms).

The evolution process goes on and on until one termination criteria is satisfied. Finally, when the iterative part of the algorithm is finished, all the final solutions are compared once again in order to delete all the dominated ones and to obtain the final Pareto front.

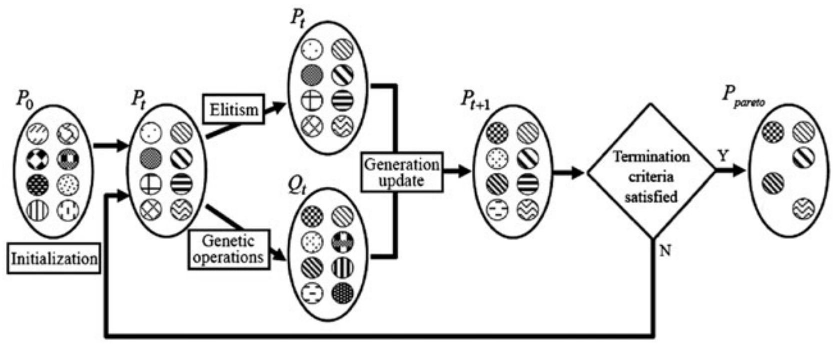



Figure 3. Basic structure of the GGA algorithm.

The algorithm has been developed with MATLAB 2012a and it is composed by a main framework which contains the principal instruction and the calls to the functions, and a series of different functions which are used to perform specific tasks (i.e. encoding; initialisation; generation of the initial population; decoding and objectives evaluation; removal of overlapping solutions; fitness; selection; crossover; mutation; and termination).

### 3.1 *Encoding*

The encoding that is used in this algorithm is a modified version of the operation-based representation: each job is represented by a natural number and each number is present in the chromosome as many times as the number of operations of the job it represents. The first occurrence represents the first operation of the job, the second occurrence represents the second operation and so on. This approach avoids the need of complicated repair procedures to handle the infeasibility of the chromosomes due to the constraints in the order of the operations, but at the same time, limits jobs to have only linear precedence relationship between operations.

### 3.2 *Initialisation of the algorithm*

In order to run smoothly, the algorithm needs to be initialised with different variables both related to itself and to the production system:

- The maximum number of generations.
- The size of the population.
- The crossover probability.
- The mutation probability.
- The information about the duration of each operation for each job.
- The information about which machine will process a specific operation.
- The power requirements for the different energetic states of different machines and the times required for the two-state transitions (off–on and standby–on).
- Set-up times needed to process a job on a machine after having processed another different job.

### 3.3 *Generation of the initial population*

Although some papers demonstrated that a population generated by intelligent designed heuristics can lead to better solutions, the majority of approaches still use a randomly initialised population. In order to have a better search within the solution space, this algorithm starts with a randomly generated population.

### 3.4 *Decoding*

When the first population is created and every time a new population is generated through genetic operators, there is the need to translate the chromosomes into actual schedules in order to evaluate their performances and to calculate the values of the objectives.

The function 'decoding_semiactive' is responsible for the decoding of a chromosome into an actual schedule and when called returns three matrices containing the start time of the operations on different machines (MStartTime), the completion time of the operations on different machines (MCompletionTime) and the jobs to which the specific operations on different machines represented in the other two matrices belong (MJobOrdered).

Iteratively, the function takes the $i$th gene of the chromosome and checks to which operation the gene refers (by checking the number of previous occurrences of the natural number associated with the job) and on which machine it will be processed. Then, it checks the time at which the job finished its $(i-1)$th operation and the time at which the machine finished its last operation. Consequently, it sets the biggest between the two as the starting time of the process, in order to respect the two constraints of 'no jobs processed on multiple machines at the same time' and 'no machines processing multiple jobs at the same time'. At this point, the algorithm checks which operation has just been processed on the machine in order to define the correct set-up time (if the current operation is the first on the machine, the set-up is not required) and adds it in the three matrices. Finally, the job can be virtually processed. Eventually, the starting time, completion time and job number are inserted into the matrices.

### 3.5 *Objectives evaluation*

The function 'objectives_evaluation' returns the values of the two main objectives for each chromosome: MKS and TEC.

The MKS objective is easily calculated as the difference between the maximum value in the MCompletionTime matrix and the minimum value in the MStartTime matrix.

The TEC objective is more difficult to assess and is therefore calculated in different steps. The first step is to find all the set-up operations for each machine on the MJobOrdered matrix, obtaining their durations as the difference between the completion times and the starting times, then summing them all up in order to obtain the total set-up time for each machine. Then, the algorithm searches for gaps between the starting time of an operation and the completion time of the previous one, which indicates that the machine was in the idle operational state. If such gap is found, then a comparison is made for the energy consumed in different energy states (i.e. idle vs. standby vs. off) as in inequalities (1)–(4). Subsequently, the alternative with the lower overall consumption is selected. If the best alternative is to keep the machine idle, the duration of the idle time is summed to the idle time value for the machine. If the standby state is found to be the best, the duration of the gap is added to the standby time vector and the counter for standby to idle transitions is incremented by one unit. Instead, if the switch-off alternative is selected, the number of start-ups is increased by one. The TEC for each machine is calculated by summing up the time which the specific machine passed in the various energetic states multiplied by the power required for those states and the number of state transitions multiplied by the power required for the transitions. Finally, the overall TEC is found by summing up all the TECs of each single machine, as represented in Equation (6).

### 3.6 *Removal of overlapping solutions*

An innovation with respect to the classic algorithms is represented by the removal of overlapping solutions. The overlapping removal algorithm is employed in two different moments during the running time of the algorithm. First, it is used in the creation of the initial population, checking for solutions that share the same objective vector (two chromosomes have the same objective vector if they have both the same MKS and the same total electrical consumption). If two or more solutions share the same objective vector, all of them except a single random one are removed from the population and a number of new chromosomes equal to the number of deleted solutions are created. The procedure is repeated until every chromosome has a different objective vector. Then the algorithm is called during the evolution phase, precisely after the generation of the *i*th population, again in order to delete chromosomes with the same objective vector.

### 3.7 *Fitness*

The fitness assignment procedure, taken from the SPEA-II algorithm, works by exploiting the concept of Pareto dominance: at first, all the members of the merged population are compared between themselves to find how many solutions are dominated by a specific chromosome *i*and by which solutions this chromosome $i$ is dominated (a chromosome $i$ dominates a chromosome $j$ if $\text{ObjMKS}_i \leq \text{ObjMKS}_j$ and $\text{ObjTEC}_i \leq \text{ObjTEC}_j$, and if at least one of the values of the two objectives of *i*is strictly lower than the same objective of $j$).

Each chromosome is therefore assigned a strength value $S_i$ equal to the number of solutions that it dominates:

$$S_i = \sum_{j=1}^{\text{PopTot}} n_j, \begin{cases} n_j = 1, & i \prec j, i \neq j \\ n_j = 0, & \text{otherwise} \end{cases} \tag{8}$$

From the strength values, it is possible to calculate the raw fitness $R_i$, which is equal to the sum of the strength of the chromosomes that dominate chromosome $i$:

$$R_i = \sum_{j=1}^{\text{PopTot}} S_j, \quad \forall j \prec i \tag{9}$$

It is important to note that in any type of problem (i.e. maximisation or minimisation), a high value of the raw fitness corresponds to a bad solution, which is therefore dominated by many individuals.

In case many individuals do not dominate each other, it may happen that a lot of chromosomes share the same fitness value, making it more difficult to differentiate between their qualities. For this reason, the fitness value is obtained from the sum of the raw fitness value and a density estimator. The scope of the density estimator is to worsen the fitness of solutions which are close together in a definite region of the search space, in order to favour more distant solutions to obtain a more disperse Pareto front.

The density estimator is created by adapting the $k$th nearest neighbour method proposed by Silverman (1986): at first, the vector $\overline{\sigma}_i$, which represents the Euclidean distance between the $i$th chromosome and each of the other chromosomes is calculated and then sorted in ascending order:

$$\sigma_i = \sqrt{\left(\text{ObjMKS}_i - \text{ObjMKS}_j\right)^2 + \left(\text{ObjTEC}_i - \text{ObjTEC}_j\right)^2} \tag{10}$$

From this vector, the $k$th element $\sigma_i^k$ is taken, with $k$ equal to the square root of the sample size ($k = \sqrt{\text{size}(\text{PopTot})}$ in our algorithm). The density estimator $D_i$ can finally be calculated as the inverse of the $k$th Euclidean distance:

$$D_i = \frac{1}{\sigma_i^k + 2} \tag{11}$$

In the denominator, the term 'two' is added to ensure that $0 < D_i < 1$ because the aim is to compare solutions with the same raw fitness and not to downgrade it.

Finally, the fitness can be calculated as the sum of the two terms:

$$F_i = R_i + D_i \tag{12}$$

### 3.8 *Selection*

The selection process operates on two different levels. First, the chromosomes with the best fitness values from the total population pool are selected to fill a temporary population. Then, the mating pool is formed from this population by using a binary tournament selection method.

The selection process for the mating pool works in a very simple way: two random chromosomes are chosen from the intermediate population and the one with the lowest fitness is chosen and copied into the mating pool. If the two chromosomes share the same fitness, a random winner is chosen.

### 3.9 *Crossover*

The function crossover takes care of choosing two different random parents from the mating pool and creating two different children which will be included in the future population by crossing the parents over. In this specific algorithm, the crossover probability is always equal to one because if a chromosome is passed as it is to the next population, it is eliminated by the overlapping solutions removal algorithm.

The PPX crossover works by creating a vector map as long as the chromosome, filled with elements of the set $\{1, 2\}$, defines the order in which the genes are taken from parent 1 and parent 2, respectively. After a gene is drawn from one parent and deleted from the other one, it is appended to the offspring chromosome. The process is repeated until both parents are empty and the offspring contains all genes involved.

### 3.10 *Mutation*

Mutation is the background genetic operator used to maintain genetic diversity in the population. Every offspring created by the crossover has a possibility to be subject to the mutation process. A random number of genes between one and one-thirtieth of the length of the chromosome is chosen and exchanged with other genes at random positions.

After the mutation process, the new $(i+1)$th population is created and is ready to undergo the whole genetic process once again, starting from the decoding and objectives evaluation.

### 3.11 *Termination*

Due to its high convergence speed, the only termination criteria for the algorithm is the maximum number of generations. Finally, after the last iteration is done, all the solutions except the non-dominated ones are eliminated, therefore obtaining the final Pareto front solutions.

## 4. Testing the GGA

In this section, the GGA is tested using different test problems from the literature and its results are compared with the other scheduling methods, highlighting its strength and weaknesses. Moreover, we analyse the effects of the different policies on the energetic performances of the system.

### 4.1 *Literature test problems*

Due to the high number of different algorithms and techniques to tackle the job shop problem, many different test prob-lems were developed during the years for being able to test the approaches on the same common ground for compara-tive purposes, and to find their strengths and weaknesses. These benchmarks were created to have different dimensions and complexity, to be able to determine the capabilities and the limitations of the algorithms. A common reference fam-ily of problems is the one introduced by Fisher and Thompson (1963), counting three different instances called, respec-tively, FT06, FT10 and FT20.

Those three problems have different dimensions both for the number of jobs and for the number of machines. In fact, the FT06 problem has 6 jobs and 6 machines and FT10 has 10 jobs and 10 machines, while FT20 has 20 jobs and only 5 machines. The processing times for these problems are randomly generated from a uniform distribution. The interval from which the numbers are picked is [1,10] for FT06 and [1,99] for FT10 and FT20. Moreover, in the last two instances, in order to simulate a typical machine shop, the lower numbered machines are assigned for earlier operations, while higher numbered machines are assigned for later operations.

Those three problems, created in 1963, due to their complexity were optimally resolved only many years later, using Branch & Bound approaches:

- FT06 was resolved in 1969 with an optimum MKS of 55.
- FT20 was resolved in 1975 with an optimum MKS of 1165.
- FT10 was resolved in 1987 with an optimum MKS of 930.

### 4.2 *MKS tests*

The first test performed in order to assess the performances of the GGA is the MKS test. The objective of this test is to find the best solution evaluated only in terms of MKS given by the GGA, and to compare this solution to the ones given by other scheduling methods.

The four different algorithms which have been chosen to perform the test are two dispatching rules, the longest pro-cessing time rule and the shortest processing time rule, and two different heuristics which are local search (LS) and the shifting bottleneck (SB).

The following table shows the MKS results obtained with the different scheduling methods and the distances from the optimum solution expressed as a percentage:

From Table 1, it is possible to see how the dispatching rules, despite being the fastest methods, computationally speaking, yield a MKS that is at least 20% worse than the optimum solution. These performances are not acceptable any-more due to superior solutions and low convergence time of other methods. The SB method does not provide great results too, ranging from a 7% difference for the easiest case to a 17% for the most difficult one. Contrarily, the LS method achieved in each of the three cases almost optimal results, which are at maximum 2% worse than the optimal solution.

The GGA, modified to run for a single objective case, scored quite well in all of the three cases. The results of 100 runs for each problem are represented in Table 2.

Table 1. Makespan for the three problems obtained with Lekin scheduling methods.

| | FT06 | | FT10 | | FT20 | |
|---|---|---|---|---|---|---|
| | Makespan | % | Makespan | % | Makespan | % |
| SPT | 73 | 32.7 | 1338 | 43.9 | 1558 | 33.7 |
| LPT | 67 | 21.8 | 1168 | 25.6 | 1516 | 30.1 |
| SB | 59 | 7.3 | 1094 | 17.6 | 1291 | 10.8 |
| LS | 55 | 0 | 945 | 1.6 | 1178 | 1.1 |
| OPT | 55 | | 930 | | 1165 | |

Table 2. Results for the single-objective GGA.

| | FT06 | | FT10 | | FT20 | |
|---|---|---|---|---|---|---|
| | Makespan | % | Makespan | % | Makespan | % |
| Min | 55 | 0 | 969 | 4.2 | 1242 | 6.6 |
| Max | 55 | 0 | 1022 | 9.9 | 1306 | 12.1 |
| Avg | 55 | 0 | 1000.4 | 7.6 | 1287.2 | 10.5 |

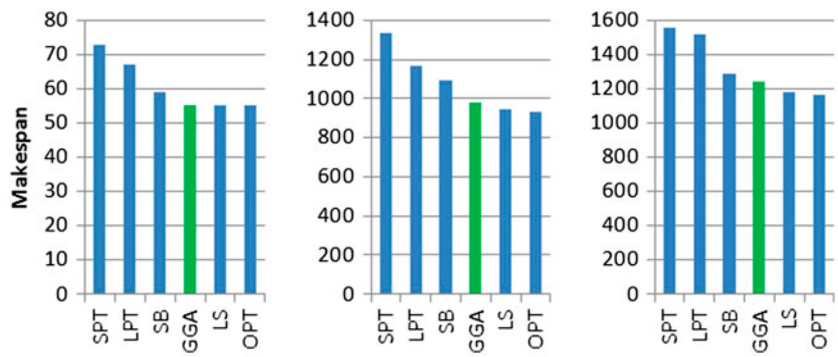



Figure 4. Comparison of the methods for the different problems.

The solutions given by the GGA, despite being quite good compared to the various methods as highlighted in Figure 4 and Table 2, are not as outstanding as the ones provided by the LS method. The main reason for this difference is that the LS method has been highly optimised to resolve a single-objective scheduling problem, i.e. MKS in this case. In fact, in order to more thoroughly explore the solution space both from MKS and energy efficiency point of views, the starting schedules are generated randomly without following any specific procedure which can greatly improve the results as demonstrated by Park, Choi, and Kim (2003). Another reason is that the chromosomes are decoded in order to create semi-active schedules instead of active ones.

### 4.3 *Adding energy consumption considerations to the MKS tests and testing behaviour policies*

Before employing the multi-objective GGA, some other preliminary considerations about the TEC have to be made. In this section, the four policies developed in the third chapter are employed to calculate the TEC of the different schedules obtained with LEKIN®, in order to make comparisons and individuate trends and areas of interest to focus further research.

The data used in order to calculate the energy consumption is fictional, but belongs to reasonable ranges. However, pure numbers have only a relative importance with respect to the ratios between the different power requirements, which have been derived by studying different papers, such as Jones, Mirrazavi, and Tamiz (2002), Zitzler, Laumanns, and Thiele (2001) and Pinedo (2012). Power consumptions and transition times for the machines are thus provided below:

$$P_{\text{work}} = 10\text{kW}; \quad P_{\text{idle}} = 6\text{kW}; \quad P_{\text{standby}} = 4\text{kW}; \quad P_{\text{ramp-up}} = 8\text{kW}$$

$$t_{\text{ramp-up,off}} = \frac{1}{O_k} \cdot t_{j,k} \quad \text{(the average of the processing times on the } k\text{th machine)} \tag{13}$$

$$t_{\text{ramp-up,standby}} = 0.5 \cdot t_{\text{ramp-up,off}} \tag{14}$$

Applying the different scheduling methods for the three problems and the four policies leads to the results demonstrated in Table 3. The saved energy percentages in the table refer to the base case of Policy 1.

Table 3. MKS and TEC for the three problems and the four policies.

| | | Policy 1 | Policy 2 | Policy 3 | Policy 4 |
|---|---|---|---|---|---|
| SPT (FT06) | MKS | 73 | | | |
| | TEC | 3680 | 2894 | 2642 | 2642 |
| | %[a] | | −21.4% | −28.2% | −28.2% |
| SB (FT06) | MKS | 59 | | | |
| | TEC | 3176 | 2642 | 2530 | 2526 |
| | % | | −16.8% | −20.3% | −20.5% |
| GGA MKS (FT06) | MKS | 55 | | | |
| | TEC | 3032 | 2636 | 2492 | 2492 |
| | % | | −13.1% | −17.8% | −17.8% |
| LS (FT06) | MKS | 55 | | | |
| | TEC | 3032 | 2618 | 2450 | 2448 |
| | | | −13.7% | −19.2% | −19.3% |
| OPT (FT06) | MKS | 55 | | | |
| | TEC | 3032 | 2636 | 2496 | 2496 |
| | % | | −13.1% | −17.7% | −17.7% |
| SPT (FT 10) | MKS | 1338 | | | |
| | TEC | 104,804 | 80,666 | 69,358 | 69,262 |
| | % | | −23.0% | −33.8% | −33.9% |
| SB (FT 10) | MKS | 1094 | | | |
| | TEC | 90,164 | 66,326 | 61,550 | 61,512 |
| | % | | −26.4% | −31.7% | −31.8% |
| GGA MKS (FT 10) | MKS | 969 | | | |
| | TEC | 82,664 | 62,180 | 61,298 | 61,266 |
| | % | | −24.8% | −25.8% | −25.9% |
| LS (FT 10) | MKS | 945 | | | |
| | TEC | 81,224 | 63,692 | 62,448 | 62,376 |
| | % | | −21.6% | −23.1% | −23.2% |
| OPT (FT 10) | MKS | 930 | | | |
| | TEC | 80,324 | 62,516 | 61,194 | 61,144 |
| | % | | −22.2% | −23.8% | −23.9% |
| SPT (FT 20) | MKS | 1558 | | | |
| | TEC | 69,216 | 61,458 | 58,690 | 58,604 |
| | % | | −11.2% | −15.2% | −15.3% |
| SB (FT 20) | MKS | 1291 | | | |
| | TEC | 61,206 | 59,550 | 57,576 | 57,560 |
| | % | | −2.7% | −5.9% | −6.0% |
| GGA MKS (FT 20) | MKS | 1242 | | | |
| | TEC | 59,736 | 54,714 | 54,714 | 54,714 |
| | % | | −8.4% | −8.4% | −8.4% |
| LS (FT 20) | MKS | 1178 | | | |
| | TEC | 57,816 | 53,376 | 53,376 | 53,376 |
| | % | | −7.7% | −7.7% | −7.7% |
| OPT (FT 20) | MKS | 1165 | | | |
| | TEC | 57,426 | 54,138 | 54,138 | 54,120 |
| | % | | −5.7% | −5.7% | −5.8% |

[a]The saved energy percentage in comparison to the base case of Policy 1.

Regarding policy number one, it is possible to see how in this case, the optimisation of the energetic consumption and the optimisation of the MKS are actually the same objective. Since the processing times on the machines are fixed and all the machines are switched on at time 0 and switched off when the last operation is processed, the idle time for each $k$th machine can be calculated as:

$$\text{idle}_k = \text{makespan} - \text{processing}_k \tag{15}$$

Therefore, since the processing time for the $k$th machine is fixed, minimising the MKS also minimises the energy consumption caused by the idle time.

Moreover, it is possible to assess that by using more and more complex policies with respect to the first one, there is always a decrease of the energy consumption of the schedule or, at most, an even situation, where the consumed energy stays at the same level. The best energetic improvements can be seen on the schedules with the highest MKS

Table 4. GGA TEC results with TEC and WEC values.

| | | FT06 | | | FT10 | | | FT20 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | P2 | P3 | P4 | P2 | P3 | P4 | P2 | P3 | P4 |
| GGA | MKS | 55 | 60 | 60 | 1089 | 1220 | 1220 | 1630 | 1630 | 1630 |
| TEC | TEC | 2558 | 2380 | 2380 | 60,218 | 58,520 | 56,520 | 53,130 | 53,130 | 53,130 |
| | % | −2.3 | −2.9 | −2.8 | −3.7 | −4.4 | −4.3 | −0.5 | −0.5 | −0.5 |
| | WEC | 324 | 146 | 146 | 5040 | 3342 | 3342 | 0 | 0 | 0 |
| | % | −15.6 | −32.4 | −31.8 | −31.3 | −44.4 | −44.0 | −100 | −100 | −100 |

and on the squared problems (i.e. FT06 and FT10 in our case), with improvements up to 34% of energy saved with respect to the base case, due to the fact that the operative idle times are higher and therefore there are more opportunities to perform transitions between states. On the contrary, in the rectangular problems such as FT20, the jobs are more and more numerous with respect to the machines, leading to a highly consolidated schedule and shorter as well as scarcer operative idle times to play with. In fact, even if the improvements with respect to the base case are in the order of 5–20%, the schedules built with the GGA, the LS and the optimal procedure show almost the same energetic consumption with all the three advanced policies.

It is also possible to see how, for the data we are using, the differences between Policy 3 and Policy 4 in terms of TEC are almost negligible. Anyhow, the TEC improvements obtained for the three different problems are not negligible and therefore any manufacturer could employ these advanced policies to save electrical energy, even without an ad hoc algorithm built to make the most out of them.

Finally, the GGA has been used again in single-objective mode to find a suboptimal minimum for the energetic consumption, this time trying to optimise the TEC objective, giving the very encouraging results which are reported in Table 4. The algorithm has not been used for policy one because, as mentioned before, the two objectives coincide.

Since WEC indicator considers only the energetic consumption in the operative idle phase, which is exactly the kind of energy consumption that the policies are trying to reduce, Table 4 also includes WEC values that better highlight the improvements obtained with the use of different policies.

From the application of the algorithm to derive the results shown in Table 4, two very interesting particular cases stand out. The first one is the FT20 case, where the algorithm was able to achieve the absolute minimum for energy consumption, which corresponds to a perfectly consolidated schedule with any idle time left for any machine. The other one is represented by the FT06 schedule with the second policy applied: here the GGA was able to find a solution which optimises both the MKS and the energetic consumption at the same time. In fact, the new schedule besides hav-ing the optimal value of MKS which is 55 s, is able to consume 2.1% less total energy than the schedule obtained with the LS method.

## 5. Validating the multi-objective GGA

With the confirmation from the previous tests that the algorithm works properly and yields good results, that the policies are effective and there are still some energy savings to achieve with respect to the schedules obtained with the tradi-tional methods, it is finally time to employ the GGA for the scope it was designed, that is optimising both the MKS and the energy consumption of a productive system.

Using still the same test problems developed by Fisher and Thompson used in the previous sections, the algorithm will be run in order to find the Pareto front solutions for the MKS and the TEC objectives for the different machine behaviour policies. The Pareto fronts represented in this section are obtained by taking the best solutions individuated through multiple runs of the algorithm.

The FT20 case is not analysed because it has been discovered from preliminary tests that the room for improvement is negligible. In fact, the optimal solutions from the MKS point of view are already suboptimal for energetic consumption.

### 5.1 *FT06 test problem*

On the FT06 problem, the GGA behaves as expected, being able to optimise both objectives and find all the possible trade-off solutions. As represented in Figure 5(a)–(c), the solutions obtained by the GGA are always better, or in the worst case, equal to the solutions proposed by the other algorithms.

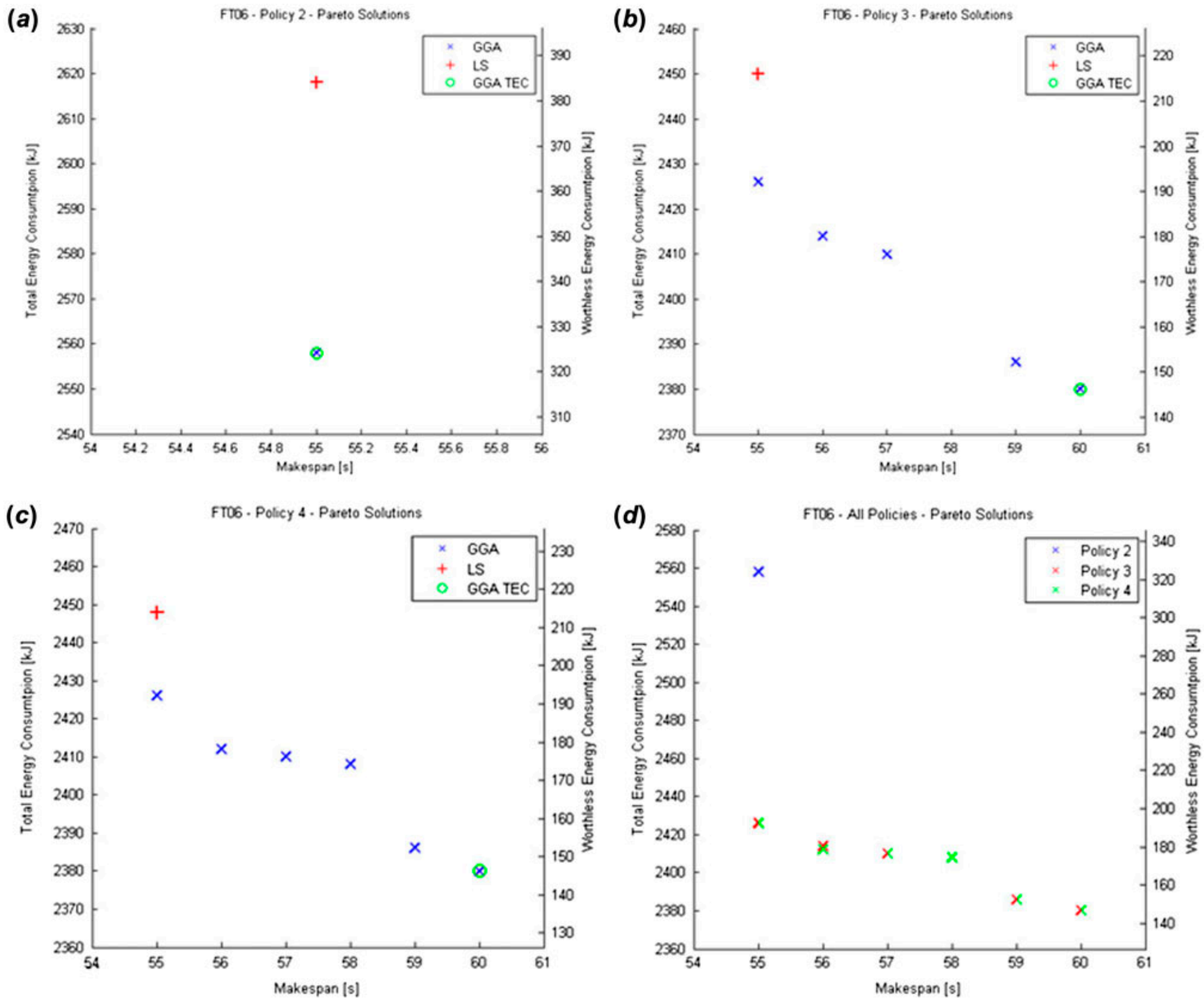



Figure 5. Pareto front solutions for the FT06 problem.

The algorithm was able, for the second policy, to find a single solution, which optimises both the MKS and the energy consumption, while, for the other two policies, it was able to find different Pareto solutions which allow us to observe how the relationship between the two objectives is inversely proportional. In fact, a higher MKS implies a reduction of the energy needed to process the schedule. It is however quite difficult to individuate a precise trend which links the two objectives, mainly because of the scarce number of solutions.

Furthermore, it is again possible to observe that by employing more complex policies the consumption of worthless energy decreases substantially. The WEC decreases about 59% when employing Policy 2 instead of Policy 1, an average of 79% when employing Policy 3 or Policy 4 (the differences between these two policies are almost negligible, as said before) instead of Policy 1 and an average of 48% when employing Policy 3 or 4 instead of Policy 2. All the Pareto solutions are provided in Table 5.

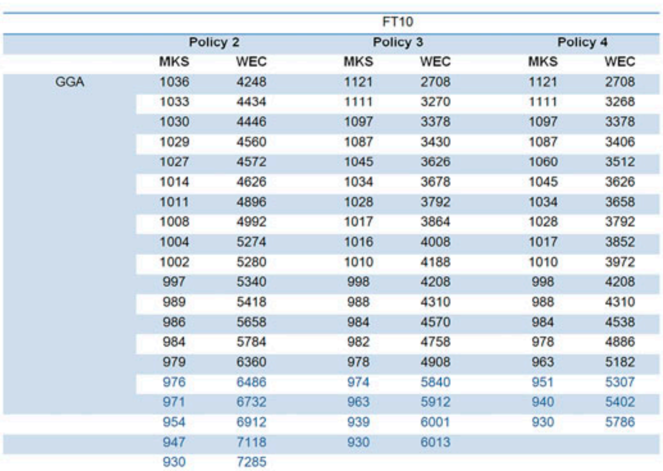
### 5.2 *FT10 test problem*

The FT10 runs just strengthen the considerations obtained from the FT06 runs. In fact, the algorithm was able to obtain many more solutions and therefore making it easier to outline a trend.

The trend between MKS and TEC is quite different between the second policy and the last two ones. The solutions appear to lie on a straight line for the second policy and on a branch of a hyperbola for the third and fourth.

Table 5. GGA results for the FT06 problem.

| | FT06 | | | | | |
|---|---|---|---|---|---|---|
| | Policy 2 | | Policy 3 | | Policy 4 | |
| | MKS | WEC | MKS | WEC | MKS | WEC |
| GGA | 55 | 324 | 60 | 146 | 60 | 146 |
| | | | 59 | 152 | 59 | 152 |
| | | | 57 | 176 | 58 | 174 |
| | | | 56 | 180 | 57 | 176 |
| | | | 55 | 192 | 56 | 178 |
| | | | | | 55 | 192 |



| | FT10 | | | | | |
|---|---|---|---|---|---|---|
| | Policy 2 | | Policy 3 | | Policy 4 | |
| | MKS | WEC | MKS | WEC | MKS | WEC |
| GGA | 1036 | 4248 | 1121 | 2708 | 1121 | 2708 |
| | 1033 | 4434 | 1111 | 3270 | 1111 | 3268 |
| | 1030 | 4446 | 1097 | 3378 | 1097 | 3378 |
| | 1029 | 4560 | 1087 | 3430 | 1087 | 3406 |
| | 1027 | 4572 | 1045 | 3626 | 1060 | 3512 |
| | 1014 | 4626 | 1034 | 3678 | 1045 | 3626 |
| | 1011 | 4896 | 1028 | 3792 | 1034 | 3658 |
| | 1008 | 4992 | 1017 | 3864 | 1028 | 3792 |
| | 1004 | 5274 | 1016 | 4008 | 1017 | 3852 |
| | 1002 | 5280 | 1010 | 4188 | 1010 | 3972 |
| | 997 | 5340 | 998 | 4208 | 998 | 4208 |
| | 989 | 5418 | 988 | 4310 | 988 | 4310 |
| | 986 | 5658 | 984 | 4570 | 984 | 4538 |
| | 984 | 5784 | 982 | 4758 | 978 | 4886 |
| | 979 | 6360 | 978 | 4908 | 963 | 5182 |
| | 976 | 6486 | 974 | 5840 | 951 | 5307 |
| | 971 | 6732 | 963 | 5912 | 940 | 5402 |
| | 954 | 6912 | 939 | 6001 | 930 | 5786 |
| | 947 | 7118 | 930 | 6013 | | |
| | 930 | 7285 | | | | |

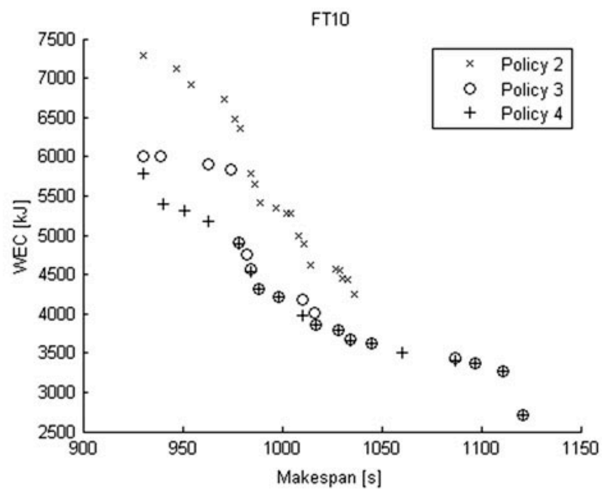



Figure 6. GGA results for the FT10 problem.

Again, the differences between the solutions with the same MKS for the different policies in terms of WEC are quite substantial, with an average difference of 79% from Policy 2 with respect to Policy 1, 84% from Policy 3 and 4 to Policy 1, 19% from Policy 2 to 3 or 4, and less than 1% from Policy 3 to Policy 4, as represented in Figure 5.

The difference between the results from the comparison between the second and the third policies and the ones obtained from the previous problem is due to the fact that the ratio between the average length of the idle times and the average ramp-up and ramp-down time of the machine is lower for the FT10 problem. Therefore, when applying Policy 3 there are more difficulties to perform a switch off and on with respect to the FT06 problem (i.e. lower difference of WEC between Policy 2 and Policy 3).

As in the previous case, the results are summarised in Figure 6. The optimal solution in terms of MKS has 1.7–2 times WEC compared to the best schedule in terms of energy consumption.

## 6. Application of the algorithm

### 6.1 *Setting the case*

In the previous sections, we analysed the variation of the energetic consumption on the different test problems with respect to the various machine tools' behaviour policies and we confirmed the performances of the GGA on multi-objective optimisation problems with conflicting objectives.

In this section, a simple application case along the lines of the test problems is developed in order to test the trend of the energetic consumption and the other variables of the schedules with respect to changes in the saturation of the productive system.

Table 6. Application case.

| Job 1 | | Job 2 | | Job 3 | | Job 4 | | Job 5 | |
|---|---|---|---|---|---|---|---|---|---|
| $M_i$ | $t_i$ | $M_i$ | $t_i$ | $M_i$ | $t_i$ | $M_i$ | $t_i$ | $M_i$ | $t_i$ |
| 1 | 14 | 2 | 13 | 3 | 17 | 3 | 10 | 4 | 14 |
| 4 | 16 | 1 | 17 | 4 | 11 | 1 | 16 | 3 | 15 |
| 5 | 20 | 4 | 17 | 5 | 11 | 2 | 19 | 2 | 20 |
| 3 | 14 | 3 | 15 | 2 | 20 | 5 | 17 | 5 | 11 |
| 2 | 20 | 5 | 17 | 1 | 11 | 4 | 12 | 1 | 19 |
| Job 6 | | Job 7 | | Job 8 | | Job 9 | | Job 10 | |
| $M_i$ | $t_i$ | $M_i$ | $t_i$ | $M_i$ | $t_i$ | $M_i$ | $t_i$ | $M_i$ | $t_i$ |
| 3 | 17 | 1 | 11 | 5 | 12 | 5 | 20 | 2 | 19 |
| 5 | 14 | 2 | 16 | 1 | 13 | 2 | 18 | 1 | 19 |
| 4 | 12 | 4 | 12 | 3 | 16 | 3 | 13 | 5 | 18 |
| 2 | 14 | 3 | 14 | 2 | 12 | 1 | 16 | 3 | 12 |
| 1 | 15 | 5 | 16 | 4 | 19 | 4 | 11 | 4 | 16 |

It has already been noticed that for the two test problems FT10 and FT20, which have the same processing energy and therefore the same processing time, the WEC is lower for the latter than for the former. In fact, the FT20 case presents a much higher saturation of the machines and therefore shorter and scarcer idle periods with respect to the FT10 problem. Nevertheless, these two problems have a different number of machines and the processing times and the routing of the jobs are not the same, therefore, representing two different and distinct production systems.

This application case is therefore built in order to analyse the variation of the energetic performances on the same identical system with respect to its saturation. The application case starts from a highly unsaturated situation, namely three jobs on five machines, and progressively increases the number of the jobs in order to reach a situation counting ten jobs on the same five-machine system.

Similar to the test problems, the processing time of the operation on different machines are randomly defined using values belonging to [10,20] interval. The order of the operations on the machine are random and there is also the necessity to set the machines up between the jobs. The set-up times, which can represent the tool change times but also the time needed to position and clamp the piece inside the machine are also randomly selected using values from [5,10] interval.

Moreover, this time, the algorithm also executes the start-up of the machines before their first operation, therefore, incrementing the MKS of the schedules because the first jobs do not start at $t = 0$, but they have to wait for the ramp-up of the machines. This aspect was not considered before in order not to create discrepancies with the MKS results of the test problems.

We decided to use a fairly simple problem because we were able to see how the performances of the GGA stood out, especially in the FT06, being always able to find the absolute optimum for the MKS. Moreover, for this stage of preliminary test, it is not absolutely necessary to create an excessively complex application case.

Table 6 shows the machines and the times required for the operations. The jobs are added from case to case following the table's order. The machine data remains the same as in the previous sections.

### 6.2 *Key performance indicators*

In this application case, we consider many different KPIs, other than MKS, TEC and WEC, which are able to synthetically describe the general performances of the production system and can help us to understand more deeply its behaviour and the relationship between the energetic consumption and other variables.

The KPIs which have been chosen are:

- MKS and TEC, as also used previously.
- The three contributions that constitute the TEC: the processing energy (ProcE), the set-up energy (SetupE) and the WEC.
- The ratio between the WEC and the useful energy consumption (UEC), with UEC calculated as the sum of the processing energy, the set-up energy and the energy needed for the first ramp-up and the last ramp-down of the machines.

- The time spent in the operation idle state (OpIdle) and the energy consumed in the three related energetic states: the idle state (IdleE), the standby state (StandbyE) and the off state (OnOffE), considering of course also the energy needed for the transition for the last two cases.
- The saturation of the whole systems (SatSYS) and the saturation of the single machines (SatMachX). The former is calculated as the time the whole system spends in the useful states (working, set-up and first start-ups) over the total time in which the system is online; the latter is calculated in the same way, but for each separate machine.

### 6.3 Results

Using again the GGA for each of the instances of the application case ranging from three to ten jobs on five different machines, we were able to study the behaviour of the productive system and draw different conclusions.

The first policy is the simplest one, where all the machines are switched on at $t = 0$ and switched off at the end of the schedule. In this case, the algorithm is able to always find a single solution which optimises both the objectives.

As illustrated in the Policy 1 results in Figure 7, the MKS and the TEC of the schedule increase as the number of jobs increases. However, at the same time also, the overall saturation of the system (and the average saturation of the machines since the two indicators coincide for this policy) does increase, therefore, reducing the time spent in the operative idle state and leading to a substantial reduction of WEC and WEC/UEC ratio (Table 7).

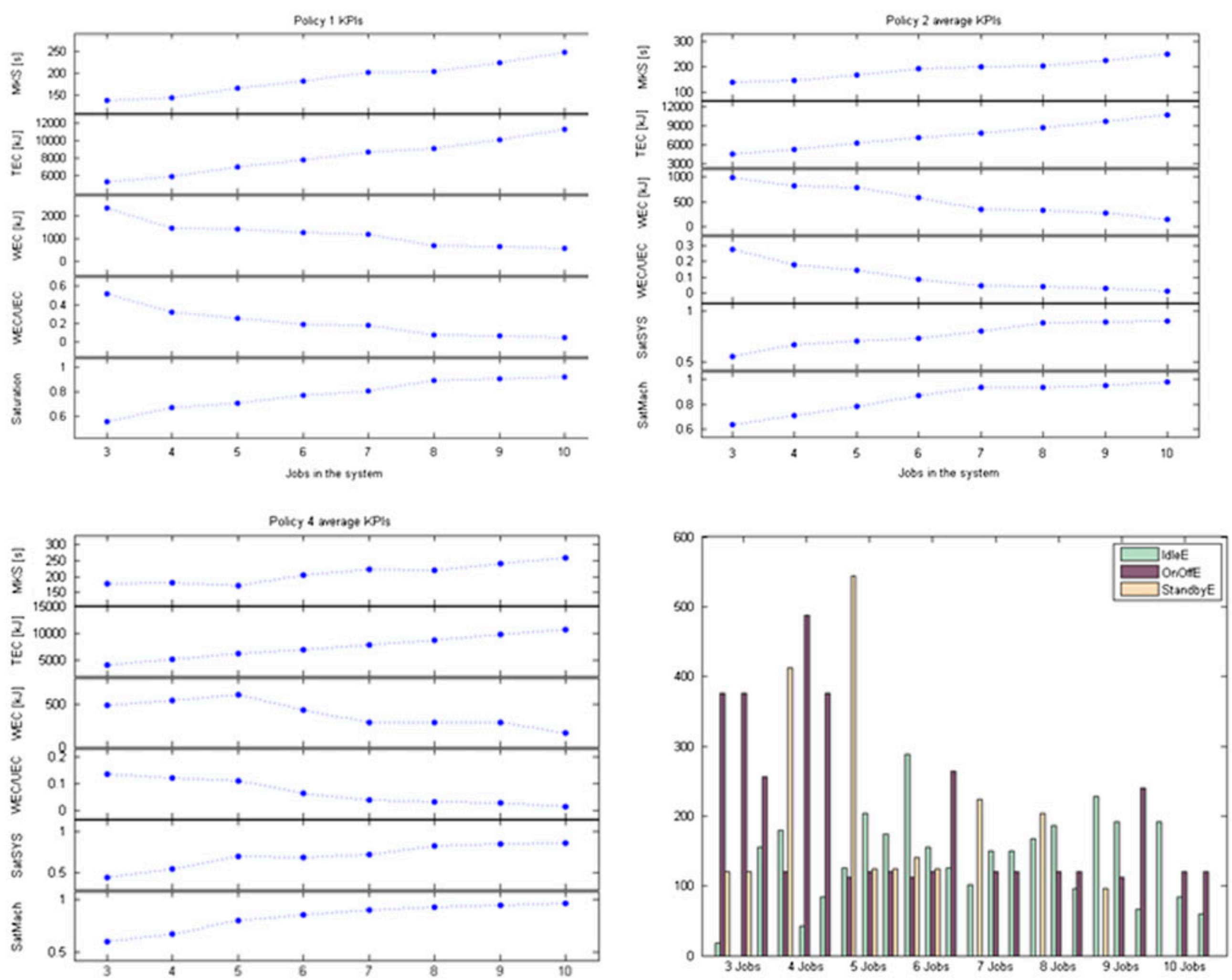



Figure 7. KPI results for different policies.

Table 7. KPI results for the policy 1.

| | MKS [s] | TEC [kJ] | SetupE [kJ] | WEC [kJ] | WEC/UEC | OpIdle [s] | SatSYS | Mean SatMach |
|---|---|---|---|---|---|---|---|---|
| 3 Job | 138 | 5376 | 592 | 1830 | 0.516 | 305 | 0.56 | 0.56 |
| 4 Job | 145 | 5934 | 808 | 1440 | 0.320 | 240 | 0.67 | 0.67 |
| 5 Job | 167 | 6976 | 1080 | 1428 | 0.257 | 238 | 0.71 | 0.71 |
| 6 Job | 182 | 7780 | 1336 | 1248 | 0.191 | 208 | 0.77 | 0.77 |
| 7 Job | 201 | 8704 | 1664 | 1170 | 0.155 | 195 | 0.81 | 0.81 |
| 8 Job | 204 | 9126 | 1832 | 696 | 0.083 | 116 | 0.89 | 0.89 |
| 9 Job | 224 | 10,102 | 2088 | 636 | 0.067 | 106 | 0.91 | 0.91 |
| 10 Job | 248 | 11,130 | 2336 | 576 | 0.054 | 96 | 0.92 | 0.92 |

For the second policy, as already happened for the FT06 problem, it is possible to find for the different instances either a single solution, which optimises both the objectives, or a handful of trade-off solutions, which are however very close between themselves in terms of MKS and TEC.

The results for the third and fourth policies are almost identical. Using these policies, we were able to obtain many more Pareto solutions with a considerably high MKS range from the fastest to the slowest solution, which varies from 30% for the 3 jobs case to 10% for the 10 jobs case.

As for policy 4, the relationship between WEC and the average saturation of machines doesn't hold true anymore. In fact, since the machines now have the possibility to be switched off or enter the standby mode, they can remain in the operative idle state for a long period without having a high energetic consumption. As a matter of fact, the solutions with the lower consumption within the same instance have a lower average saturation of the machines due to this possibility.

Figure 7 demonstrates the results of the KPIs for different policies.

Finally, after having analysed the results of the single policies, the final step is to compare the results of the different policies on the single instances, to be able to further assess the benefits of their applications. Below, we compare for each instance, the three solutions with the best MKS and the three solutions with the best TEC taken from each policy.

The solutions with the best MKS clearly have the same MKS for each instance; therefore, the differences between the policies can only be seen in terms of TEC. The differences between the first and the second policies or between the first and the fourth policies are quite substantial, while the differences between the second and the fourth policies are much more restrained. There is a general trend for these differences to decrease as the saturation of the system increases, becoming very close to zero for the last instances of the problem and for the most complex policies.

The solutions with the best TEC, instead, beyond the differences in TEC additionally present some quite substantial differences in terms of MKS. Between the first and the second policies, there are only negligible differences in terms of MKS, while contrarily, the differences in terms of TEC are quite substantial. Between the first and the fourth, the differ-ences in TEC are predictably even higher, but also the differences in terms of MKS are quite substantial except for the already quoted 5 jobs case. On the contrary, between the second and the fourth policies, the differences in terms of TEC are not that outstanding as in the previous two comparisons (but still higher than in the best MKS case), while the differences in terms of MKS are still quite substantial and only a little lower than in the P1/P4 comparison.

Finally, it is possible to notice how, also in this case, all the values of these differences have the same general tendency to decrease as the jobs in the system increase.

## 7. Conclusion

This study focused on developing different policies in order to control the behaviour of machine components and therefore decide in which energetic state the machines should be operated depending on the duration of the period and the power requirements of the different states. The major characteristics of the proposed algorithm are its ability to concurrently consider objectives related to the productivity and objectives related to the energy consumption, giving as a result a series of different Pareto front solutions, which represent the trade-offs between the objectives and the possibility to remove solutions with the same objective vector, thereby improving the overall quality of the Pareto front.

Two main considerations have emerged from the application of the algorithm. First, the algorithm performs relatively better than other methods, being able to achieve a semi-optimal MKS similar to that obtained by the best of the lot, but with a significantly lower TEC from 0.5 to 5% depending on the problem and on the policy. Second, by employing more complex policies, the study demonstrated that the WEC can be reduced by an average value of 69% passing from the first policy to the second one and then by another 34% passing from the second one to the third and fourth ones.

## Acknowledgements

The authors thank the editor and two anonymous reviewers for their insightful criticisms, encouragement and helpful suggestions, which greatly improved this paper. We would like to thank also our colleague Fabrizio Chiesa for his support in the research. A special thanks goes to our British colleague Emmanuel Damilola Lawal for reading and editing the manuscript.

## ORCID

*Gökan May* http://orcid.org/0000-0002-9634-999X

## References

Chen, G., L. Zhang, J. Arinez, and S. Biller. 2013. "Energy-efficient Production Systems Through Schedule-based Operations." *IEEE Transactions on Automation Science and Engineering* 10 (1): 27–37.

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

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

Devoldere, T., W. Dewulf, W. Deprez, B. Willems, and J. R. Duflou. 2007. "Improvement Potential for Energy Consumption in Discrete Part Production Machines." In *Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses – Proceedings of the 14th CIRP Conference on Life Cycle Engineering*, edited by S. Takata and Y. Umeda, 1–16. London: Springer-Verlag.

Dietmair, A., A. Verl, and P. Eberspaecher. 2010. "Predictive Simulation for Model Based Energy Consumption Optimisation in Manufacturing System and Machine Control." Flexible Automation and Intelligent Manufacturing, FAIM2009, Teesside, UK.

Drake, R., M. B. Yildirim, J. T. Twomey, L. W. Whitman, J. Ahmad, and P. Lodhia. 2006. "Data Collection Framework on Energy Consumption in Manufacturing." Institute of Industrial Engineering Research Conference. Orlando, FL.

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

Fisher, H., and G. L. Thompson. 1963. "Probabilistic Learning Combinations of Local Job-shop Scheduling Rules." In *Industrial Scheduling*, edited by J. F. Muth and G. L. Thompson, 225–251. Englewood Cliffs, NJ: Prentice Hall.

Gutowski, T., J. Dahmus, and A. Thiriez. 2006. "Electrical Energy Requirements for Manufacturing Processes." 13th CIRP International Conference on Life Cycle Engineering. Leuven.

He, Y., and F. Liu. 2010. "Methods for Integrating Energy Consumption and Environmental Impact Considerations into the Production Operation of Machining Processes." *Chinese Journal of Mechanical Engineering* 23 (4): 428–435.

He, Y., F. Liu, and H. L. C. Cao. 2005. "A Bi-objective Model for Job-shop Scheduling Problem to Minimize Both Energy Consumption and Makespan." *Journal of Central South University of Technology* 12 (2): 167–171.

Hu, S., F. Liu, Y. He, and T. Hu. 2012. "An On-line Approach for Energy Efficiency Monitoring of Machine Tools." *Journal of Cleaner Production* 27: 133–140.

IEA. 2008. "Worldwide Trends in Energy Use and Efficiency: Key Insights from IEA Indicator Analysis." Accessed June 29, 2013. http://www.iea.org/Textbase/Papers/2008/indicators_2008.pdf

Jin, K., H. C. Zhang, P. Balasubramaniam, and S. Nage. 2009. "A Multiple Objective Optimization Model for Environmental Benign Process Planning." IEEE International Conference on Industrial Engineering and Engineering Management, 869–873. Beijing.

Jones, D. F., S. K. Mirrazavi, and M. Tamiz. 2002. "Multi-objective Meta-heuristics: An Overview of the Current State-of-the-art." *European Journal of Operational Research* 137 (1): 1–9.

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

May, G., M. Taisch, B. Stahl, and V. Sadr. 2013. "Toward Energy Efficient Manufacturing: A Study on Practices and Viewpoint of the Industry." *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services, IFIP Advances in Information and Communication Technology* 397: 1–8.

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

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

Park, B. J., H. R. Choi, and H. S. Kim. 2003. "A Hybrid Genetic Algorithm for the Job Shop Scheduling Problems." *Computers & Industrial Engineering* 45 (1): 597–613.

Pinedo, M. L. 2012. *Scheduling: Theory, Algorithms and Systems*. 4th ed. New York: Springer.

Prabhu, V. V., H. W. Jeon, and M. Taisch. 2013. "Simulation Modelling of Energy Dynamics in Discrete Manufacturing Systems." *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics Studies in Computational Intelligence* 472 (1): 293–311.

Prindle, W. R. 2010. "From Shop Floor to Top Floor: Best Business Practices in Energy Efficiency." *Virginia: Pew Center on Global Climate Change, ICF International*. Accessed May 25, 2013. http://www.pewclimate.org/docUploads/PEW_EnergyEfficiency_FullReport.pdf

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Subai, C., P. Baptiste, and E. Niel. 2006. "Scheduling Issues for Environmentally Responsible Manufacturing: The Case of Hoist Scheduling in an Electroplating Line." *International Journal of Production Economics* 99 (1–2): 74–87.

Sun, Z., S. Biller, F. Gu, and L. Li. 2011. "Energy Consumption Reduction for Sustainable Manufacturing Systems Considering Machines with Multiple Power States." ASME 2011 International Manufacturing. *Science and Engineering Conference* 2: 99–103.

Yildirim, M. B., and G. Mouzon. 2011. "Single-machine Sustainable Production Planning to Minimize Total Energy Consumption and Total Completion Time Using a Multiple Objective Genetic Algorithm." *IEEE Transactions on Engineering Management* PP (99): 1–13.

Zitzler, E., M. Laumanns, and L. Thiele. 2001. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Technical Report 103. Zurich: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH).