CrossMark

SHORT ORIGINAL PAPER

Optimal scheduling of manufacturing processes across multiple production lines by polynomial optimization and bagged bounded binary knapsack

A. Legarretaetxebarria[1] · M. Quartulli[1] · I. Olaizola[1] · M. Serrano[2]

Abstract Key objectives of 'Industry 4.0' methodologies in manufacturing include improved quality control, predictive maintenance and tracking. We consider whether a further objective that can easily and cost-effectively be achieved is the implementation of aggressive optimization of production schedules across multiple manufacturing lines. The present contribution describes how this objective can be achieved by exploiting operations research techniques for the supporting of plant operation scheduling. We focus on methodologies for optimizing energy and time consumption in particular. The data acquisition from networked sensors ends with the aggregation and insertion of measures in a relational database of multi-variate sample records representing the state of separate production lines. Starting from these records, production plans can be generated automatically. Plans can be represented as tables reporting sequences of the numbers of pieces of a specific type to be produced by each line. We describe schedule optimization methodologies in detail, reporting in particular on the costs of their computation. We finally discuss issues related to the design of the end user interface, which can usefully be based on a web service-oriented architecture meant to allow communicating the obtained results to human operators, finally allowing them to monitor and implement the resulting schedules. Results are evaluated on real data acquired by sensors installed in a metal injection molding plant in Bizkaia, Spain.

Keywords Energy optimization · Scheduling

✉ A. Legarretaetxebarria
  alegarreta@vicomtech.org

  M. Quartulli
  mquartulli@vicomtech.org

  I. Olaizola
  iolaizola@vicomtech.org

  M. Serrano
  mikel.serrano@dominion.es

[1] Vicomtech-IK4, Paseo Mikeletegi 57, Parque Científico y Tecnológico de Gipuzkoa, 20009 Donostia/San Sebastián, Spain

[2] Dominion Solutions, Ibaez de Bilbao 28, 48009 Bilbao, Spain

1 Introduction

Product manufacturing typically improves competitiveness by increasing quality and reducing cost. Interactive simulation techniques contribute to this improvement by helping identify potential optimisations without incurring in heavy physical testing processes [2,3,16]. There are many different approaches in the literature that are based in physical simulations and deterministic models that optimise industrial processes in terms of quality, time, raw material, etc. From a design perspective, Cherifi et al. [11] propose the use of methodologies such as TRIZ [8] to introduce new factors in the design process.

On the other hand, the machine tool industry is currently creating devices fully equipped with sensors that provide massive and heterogeneous data on the production process that is available in real-time. This data can contain highly valuable information that, if properly characterised and modelled, could dramatically improve optimisation processes.

As a specific realization of this approach, we propose the use of linear and non-linear optimization methods to generate optimal schedules for reducing energy and time consumption, increasing in this way the competitiveness of aluminium injection processes. The main limitation of scheduling processes for manufacturing is the computational cost in terms of processing time for the creation of optimal schedules, especially in non-linear optimization methods: for

🙋 Springer

real world applications, it is crucial to be able to provide a low time response for the scheduling task.

The formal understanding of industrial production processes and their optimization have represented valuable objectives for the effective management of manufacturing plants at least since the work of Frederick Taylor [28]. The possibility for innovation is brought forward by massive real time collection of sensor data which represent a valuable asset to these ends. Methodologies to exploit this data are typically based on on-line analytical procedures that can be set up to summarize the state of the production plant and to identify and suggest possibilities for improvement in the overall efficiency.

Energy consumption optimization is growing in societal significance [13] and as a consequence energy-efficient manufacturing [25] is one of the top priorities for future years [22]. To this end, energy-optimal production scheduling methodologies are gaining growing importance in manufacturing [14].

Process scheduling is based on optimization procedures that need to consider large numbers of input variables (the types of parts that are to be produced, the different steps and components in the production process) and of operational constraints (for instance, not all the machines are capable to produce every kind of part) across multiple manufacturing lines. In principle, multiple locally optimal production schedules exists, while global maxima need if at all possible to be identified.

The present contribution introduces the results of a specific study carried out on real data resulting in the definition, implementation and validation of procedures for the time- and energy-optimal on-line scheduling of production across a manufacturing plant operating by metal injection molding. The on-line capabilities of the proposed methodology imply that unforeseen contingency situations related e.g. to the interruption of manufacturing processes in one or more lines can be accounted for by rapidly rescheduling the production of the required output on available facilities.

The structure of the present contribution can be described as follows: Sect. 2 describes the state of the art and related work, Sect. 3 introduces the proposed methodology in detail, Sect. 4 describes the data acquisition process and characteristics and in Sects. 5 and 6 the employed techniques to optimize energy and time consumption are analyzed. In Sect. 7 computational results are evaluated and in Sect. 8 the quality of the final optimization results is evaluated. The interactive scheduling visualization component is introduced in Sect. 9, which is followed by the Conclusions.

2 Related work

The present work builds on contributions put forward in the state of the art by a number of authors proposing related methodologies [1,5,24,27]. All these contribution focus on process scheduling optimization given the constraints imposed by manufacturing process planning. The relation between the two can be described as follows. Manufacturing process planning defines how a product will be manufactured. It entails the minimization of objective functions describing for instance the cost of the produced good or its makespan. It therefore is a prerequisite to scheduling and implicitly defines its context. In contrast to this, process scheduling has the objective of producing a timetable describing the utilization of the manufacturing resources defined by the planning for the optimal production of the intended items [26]. In the remainder of this contribution, we focus solely on scheduling, treating the results of planning as a given that is not subject to further optimization.

A number of classes of Parallel Machine Scheduling Problems (PMSPs) are investigated in the literature [15]. The problem is given by a set $N = \{1, 2, 3, \ldots, n\}$ of n jobs to be scheduled in m unrelated parallel machines $M = \{1, 2, 3, \ldots, m\}$. Any job $j \in N$ can be carried out on any machine $k \in M$ that is able to process it, with an associated time cost $t_{j,k}$ and an energy cost $e_{j,k}$ which depends on the characteristics of the machine and on those of the specific job. Each job is released at the beginning of the scheduling period.

Contributions in the literature distinguish among three PMSP categories: identical, uniform and unrelated PSMPs [10]. The problem presented in this paper concerns the optimization of the scheduling of N jobs in M parallel unrelated PMSP. The jobs' processing times and consumptions are machine-dependent and without any relationship among machines.

In further detail, in the frame of this contribution, we focus on PSMPs with machine-independent setup times [17], assuming setup time variations negligible with respect to actual production times and therefore disregarding them. This assumption represents a simplification with respect to the full framework considered by contributions in which setup times are explicitly considered as dependent for instance on the specific manufacturing line to be employed and on the manufactured item type [4,18].

In order to find optimal solutions to the scheduling, most contribution reduce the optimization problem to one of the standard NP-hard combinatorial problems [20]. Multiple constraints related to available production times and possibly materials need to be taken into account.

A further constraint refers to the fact that that each job needs to be allocated to a machine or manufacturing line that is known to be able to process it. This machine–product compatibility constraint stems from the fact that in general machines/production lines are non-identical, and need therefore to be considered incapable of producing a product of a specific type, unless evidence is available of the contrary—

for example given by the fact that they have been observed producing products of that class according for instance to historical log records. More complex variations of this constraint are of course possible, and can be described in terms of criteria taking into account multiple characteristics of product types and machines. We consider this simplified approach a good approximation of reality for the purposes of this contribution.

Resulting processing schedules are typically represented as tables reporting sequences of the numbers of pieces of a specific type to be produced by each specific processing line or manufacturing machine.

A final issue refers to the visualization of the obtained optimal schedules. For time oriented schemes of this type, conventional diagrams in the form of Gantt charts [29] are widely used and understood in applications as diverse as project management, the representation of patient histories [12] or manufacturing planning/scheduling visualizations [19]. Their integration into interactive interfaces allows their exploration and interpretation even when highly detailed plans are output by the planning system [23].

3 General optimization methodologies

Operations research methods can be used to optimize production task schedules taking into account energy consumption and time limitations. Their general objective is the definition of methodologies for science based decision making taking into account on the best utilization of available resources.

The general format of operations research models includes or objective function which can in general be considered as representing the cost of a given solution to the problem. A second component of the standard problem definition is a set of constraints which are represented by additional inequalities. These inequalities bound the the feasible set, the sub-space of valid solutions.

As a specific example, in the case of linear problems, the cost function can be represented as

$$minimize : f(x) = c^T x. \qquad (1)$$

Linear constraint inequalities can be written as

$$subject\ to : Ax = b \\ x <= 0. \qquad (2)$$

In this case, a polygonal feasible set results (Fig. 1).

In the case in which both the inequalities and the objective are linear, the level curves of the objective functions can be understood as hyperplanes orthogonal to c, while an optimal $x*$ point can be found in P that goes as far as possible in the direction c [6].
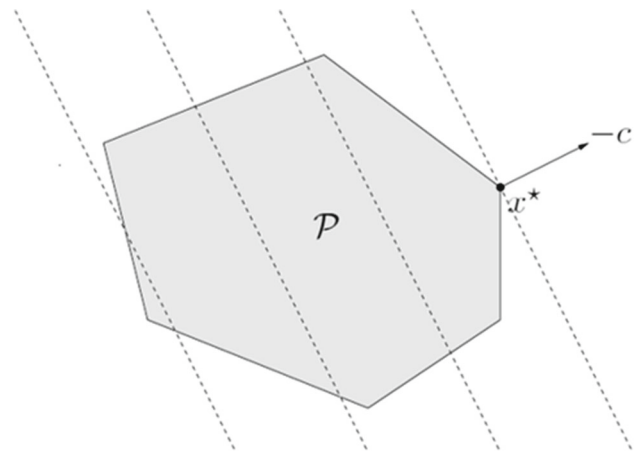

Fig. 1 Geometric interpretation of an Linear Programming [6]

In the case that the objective function has non-linear components, more complicated methodologies need to be taken into account. A possible way to address the problem is to reduce the optimization problem to a known combinational optimization problem of the kind of the Bagged Binary Knapsack [7]. In this standard problem, we are given a set of n items and m sacks to save them in. Each item i has a well-defined associated profit or value $p(i)$ and a size $s(i)$, and each sack j has a known capacity $c(j)$. Solving the problem amounts to finding a subset of items with maximum profit such that they have a feasible packing in the available bins [9]. Finding an optimal solution is in general hard: the Knapsack problem is known to be NP-hard and, as a consequence, methods for efficiently finding approximated or locally-optimal yet good-enough solutions are often employed.

4 Specific problem setup

Optimal schedules need to be defined for a metal injection molding plant equipped with 31 injector machines. Every injector is provided with components for the real-time collection of sensor data designed to allow the monitoring of the energy consumed by the elements of the plant.

The collected sensor data is aggregated locally by control systems and immediately transmitted to a central server for secure storage. The central server ingests the acquired measures in a relational database that keeps track of energy and time consumption with a sampling period of around one second. As additional information, the number of correct and invalid output pieces produced is inserted in the database at the end of every shift.

According to the available historical records, the machines are collectively capable of producing 57 different kinds of output parts, yet as mentioned above not all the parts can be produced by any machine. Again, for extended schedules we

assume we can neglect mold changing time variations and machine-dependent setup time variations with respect to the actual production times.

Operators read the proposed production schedules from graphical terminals installed next to the production machines, and are able to monitor the progress of the production on the same terminals. A consequence of this configuration is that keeping computational costs controlled across the different schedule optimization algorithms is essential to allow the final user who is in front of the interface to keep working efficiently.

5 Energy optimization methodology

For M given machines, an energy-optimal production task schedule can be defined based on linear programming methods based on operations research.

An objective function E_i specific to the production of part i can be defined based on the total number of parts produced. If i is the part index $(1 \ldots I)$ and m is the machine index $(1 \ldots M)$, n_i^m is the number of parts per hour that a specific machine m is capable of producing, t_i^m is the time that the machine needs to produce each part and e_i^m is power consumption in W for the given injector for each part per hour, then obviously:

$$
\begin{aligned}
E_i &\left(t_1^1, \ldots, t_1^M, t_2^1, \ldots, t_2^M, t_I^1, \ldots, t_I^M \right) \\
&= \sum_{m=1}^{M} \sum_{i=1}^{I} \left(n_i^m \cdot t_i^m \right) \cdot e_i^m
\end{aligned}
\tag{3}
$$

Equations 4 and 5 show the optimization constraints, where O_i is the total number of parts of type i that must be produced and T is the maximum amount of time that can be used to produce all parts:

$$
\sum_{m=1}^{M} \left(n_i^m \cdot t_i^m \right) = O_i
\tag{4}
$$

$$
\sum_{m=1}^{M} \sum_{i=1}^{I} \left(n_i^m \cdot t_i^m \right) \leq T
\tag{5}
$$

The structure of the energy optimization objective function and of its constraints has the linear structure described in Sect. 3. As a consequence, while a large variety of optimization techniques can be employed depending on the structure of the objective function and on the constraints, simple Linear Programming techniques can be used for energy optimization.

An example optimized schedule is represented in Figs. 6 and 8.

6 Time optimization methodology

If i is the part index $(1 \ldots I)$ and m is the machine index $(1 \ldots M)$, n_i^m is the number of parts per hour that a specific machine m is capable of producing, t_i^m is the time that the machine needs to produce each part.

$$
T \left(t_1^1, \ldots, t_1^M, t_2^1, \ldots, t_2^M, t_I^1, \ldots, t_I^M \right) = \max_{m} \sum_{i=1}^{I} t_i^m
\tag{6}
$$

While energy optimization can easily be carried out by linear methods, the objective function used in time optimization (Eq. 6) evidently includes a non-linear expression that represents the minimum time taken by the slowest machine.

Given the non-linear nature of the objective function considered for the time optimization, we map it to a well-known Bagged Binary Knapsack problem in order to more easily define solution methodologies for it.

To this end, we map the processing times in the scheduling problem to the item weights in the standard formulation of the problem, and the available machine time to the space left in the sack representing the machine. A measure of urgency for a given piece (a ratio between the number of items to be produced for a given product class and the available time left on the machines capable of producing that kind of piece) is taken as a measure of profit or value for a given piece.

Approximate solution methods capable of efficiently identifying locally-optimal solutions to the Bagged Binary Knapsack can therefore be exploited. We employ an approximation based on the Dantzig method [21] to greedily solve the problem, sorting the items to be produced in decreasing order of value per unit of weight and inserting them in order into the current proposed scheduling solution set.

The obtained methodology represents a heuristic converging to a local minimum. If S^m represents the slack for given machine m then $(\sum_m n_p^m \cdot S^m)_{p \in P}$ is the maximum number of of parts that can be produced considering all machines concurrently. A valid scheduling solution must verify the condition that the scheduling order it represents does not exceed production capacity:

$$
\left(\sum_{m} n_p^m \cdot S_p^m \right)_{p \in P} \quad \forall p \in P = O_p.
\tag{7}
$$

The mapping of the quantities from the scheduling to the Bagged Binary Knapsack problem can be described as follows:

– the available machine time S_p^m corresponds to the slack for machine m, and is mapped to the available space in the Knapsack to be filled

- a measure of the urgency of each piece to be produced, $O_p / \sum_m n_p^m S_p^m$, is mapped to the value of the item to be added to the Knapsack
- the number of maximum parts that can be produced in the available time, $n_p^m S_p^m$, is mapped to the number of available items for the Knapsack
- the weight of the element in the Knapsack is mapped to the number of minutes to produce the required product $1/n_p^m$.

The most urgent product $\hat{p}$ can therefore be identified by

$$\hat{p} = \arg \max_p \left(\frac{O_p}{\sum_m n_p^m \cdot S_p^m} \right)_p \qquad (8)$$

The machine $\hat{m}_{\hat{p}}$ that is most available for it

$$\hat{m}_{\hat{p}} = \arg \max_m \left(n_{p=\hat{p}}^m \cdot S_p^m \right)_m \qquad (9)$$

can be assigned to it to progressively allocate the available time to parts to be produced.

Equivalent rescaled and quantized weights for the machine that is fastest at the most urgent product can be described as

$$\left(w_p^{m=\hat{m}_{\hat{p}}} \right)_{p \in P} = \left(1/n_p^{m=\hat{m}_{\hat{p}}} \right)_{p \in P} \qquad (10)$$

Iteratively updating the order matrix

$$\sum_p \hat{S}_p^{m=\hat{m}} = \sum_p n_p^{\hat{m}} / \hat{O}_p^{\hat{m}} \qquad (11)$$

and the computed slack by $S_p^m - \sum_p \hat{S}_p^m$ results in an effective optimization of the production schedule across production lines that is attainable in a limited number of iterations and is therefore usable for on-line scheduling of manufacturing orders.

If the constant quantity C denotes the maximum available time to produce all parts, then:

$$C \geq \sum_{m=1}^{M} \sum_{i=1}^{I} \left(n_i^m \cdot t_i^m \right) \qquad (12)$$

Examples of locally time-optimal schedules are represented in Figs. 6 and 8.

7 Experimental results

The proposed optimization algorithms were implemented in the Python interpreted language and exposed through a web

service from a network server equipped with an Intel Core i7-4790 3.60 GHz CPU and 4GB of RAM.

Three sets of experiments were designed and ran in the server comparing the results between 2 algorithms. In the first test, an iterative production of a variable number n of jobs for 200 parts were simulated in m machines. In the second one, the same iterative jobs and machines were used for 1000 parts. In the third set, 5000 parts were considered.

The experimental parameters and the obtained results are summarized in Table 1. The nomenclature for iterations is $m/n/p$ where the m index denotes the number of machines, n indicates the number of jobs to be inserted in the schedule, and p is the number of parts to be produced.

Figures 2, 3 and 4 presents the results in graphical format. The format of the index used in the diagrams deserves a detailed explanation. The index is composed of two integers separated by a slash, with a format $k\,j$ according to the notation in Sect. 2. Again, the first integer $k \in M = 1 \ldots m$ represents the number of concurrent machines, while the second one $j \in N = 1 \ldots n$ represents the number of jobs.

The maximum number of jobs that we can consider, given the historical event logs made available by the real plant, is until the writing of this contribution of 10: this is the current number of different pieces ever produced in the two most prolific machines of the plant since the installation of the metrics logging system.

The maximum number of machines that we consider is again of 10: a significant set of existing machines whose work is recorded in the event logs have been necessarily left out of the present analysis because they had been essentially used isolation, processing jobs that had not been assigned to almost any other one. The jobs meant for them would have been "stuck" in this work centres without the possibility for the scheduler to re-allocate them to a different one.

Experimental energy optimization results In the case of energy-optimal scheduling without any maximum time constraint, computational costs in time for the three different tests have measures between 4.07 seconds for the fastest and 18.30 seconds for the slowest one. The consumed time grows linearly between 4 and 10 second for scheduling the production of 200 parts (Fig. 2), it stays between 10 and 12 for the experiment in which the production of 1000 parts is considered (Fig. 3), and it grows exponentially from 4 to 18 seconds for the 5000 parts scheduling experiment (Fig. 4). When a constraint specifying the maximum available production time is added to the problem formulation, experimental time consumption results are not affected by the addition.
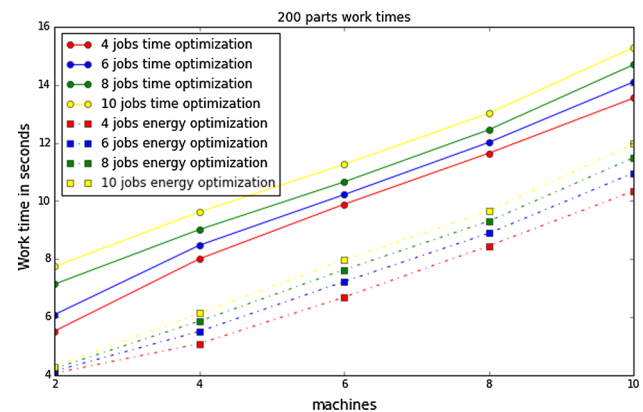
Experimental time optimization results In the case of time-optimal scheduling, the computational cost grows linearly in the 200 parts and in the 1000 parts experiments (Figs. 2, 3) from 5.51 to 22.03 s. For the 5000 parts scheduling

Table 1 consumption times for optimization without time constraint in seconds

Work size	Energy optimization (CPU time)		Time optimization (CPU time)	
m / n / p	Mean	Stddev	Mean	Stddev–index4
2 / 4 / 200	4.07	0.11	5.51	0.66
2 / 6 / 200	4.14	0.12	6.08	0.76
2 / 8 / 200	4.23	0.17	7.13	1.71
2 / 10 / 200	4.30	0.19	7.74	1.84
4 / 4 / 200	5.08	1.66	8	1.76
4 / 6 / 200	5.50	1.81	8.47	2.01
4 / 8 / 200	5.86	1.88	9.01	2.33
4 / 10 / 200	6.13	1.9	9.62	2.71
6 / 4 / 200	6.68	2.43	9.89	2.67
6 / 6 / 200	7.22	2.86	10.22	2.72
6 / 8 / 200	7.63	3.05	10.65	2.94
6 / 10 / 200	7.97	3.15	11.26	3.46
8 / 4 / 200	8.45	3.47	11.65	3.63
8 / 6 / 200	8.89	3.69	12.02	3.79
8 / 8 / 200	9.30	3.9	12.45	4.03
8 / 10 / 200	9.65	4.02	13.02	4.47
10 / 4 / 200	10.34	4.75	13.55	4.83
10 / 6 / 200	10.96	5.3	14.1	5.24
10 / 8 / 200	11.49	5.62	14.69	5.72
10 / 10 / 200	11.97	5.87	15.29	6.16
2 / 4 / 1000	11.68	5.91	15.06	6.1
2 / 6 / 1000	11.37	5.97	15.16	5.99
2 / 8 / 1000	11.06	6.01	15.65	6.28
2 / 10 / 1000	10.77	6.02	16.29	6.89
4 / 4 / 1000	10.64	5.93	16.22	6.77
4 / 6 / 1000	10.53	5.84	16.4	6.7
4 / 8 / 1000	10.43	5.75	16.92	7.09
4 / 10 / 1000	10.35	5.66	17.7	8.06
6 / 4 / 1000	10.36	5.57	17.68	7.94
6 / 6 / 1000	10.39	5.47	17.86	7.87
6 / 8 / 1000	10.44	5.4	18.32	8.13
6 / 10 / 1000	10.48	5.32	19.02	8.91
8 / 4 / 1000	10.61	5.3	19.06	8.78
8 / 6 / 1000	10.72	5.27	19.26	8.73
8 / 8 / 1000	10.87	5.27	19.68	8.95
8 / 10 / 1000	10.99	5.25	20.36	9.71
10 / 4 / 1000	11.26	5.43	20.52	9.63
10 / 6 / 1000	11.53	5.61	20.83	9.69
10 / 8 / 1000	11.79	5.77	21.33	10.06
10 / 10 / 1000	12.02	5.89	22.03	10.85
2 / 4 / 5000	4.20	0.27	139.74	1.49
2 / 6 / 5000	4.23	0.2	212.61	73.81
2 / 8 / 5000	4.30	0.2	302.06	141
2 / 10 / 5000	4.38	0.26	374.14	175.02
4 / 4 / 5000	5.02	1.32	327.16	182.66

Table 1 continued

Work size	Energy optimization (CPU time)		Time optimization (CPU time)	
m / n / p	Mean	Stddev	Mean	Stddev–index4
4 / 6 / 5000	5.49	1.59	320.35	167.3
4 / 8 / 5000	5.95	1.89	343.61	165.03
4 / 10 / 5000	6.22	1.93	441.55	187.05
6 / 4 / 5000	7.95	1.99	393.6	207.2
6 / 6 / 5000	8.41	2.15	362.38	205.98
6 / 8 / 5000	9.54	2.03	353.07	191.09
6 / 10 / 5000	9.80	2.02	433.81	205.96
8 / 4 / 5000	11.46	2.06	386.46	221.17
8 / 6 / 5000	12.03	2.45	354.4	220.02
8 / 8 / 5000	13.00	2.08	337.38	194.02
8 / 10 / 5000	13.31	2.15	405.73	193.18
10 / 4 / 5000	15.54	3.22	368.73	209.4
10 / 6 / 5000	16.36	3.67	345.97	205.26
10 / 8 / 5000	17.84	3.65	328.47	186.72
10 / 10 / 5000	18.30	3.63	395.12	182.83

Fig. 2 Comparison between energy optimization and time optimization CPU work time for 200 parts

experiment, with two or more machines CPU time consumption rises until 139.74 in the best case. This represents a problem, since as we described in a previous Section, a practical requirement for the exploitation of the scheduler in the operational environment is that the response is produced in less than one minute. Again, adding a maximum total available time constraint does not affect computation times.

Comparison between experimental results Figures 2 and 3 confirm that time consumption is larger for time-optimal than for energy-optimal scheduling, as expected from the different performance of linear optimization and non-linear optimization algorithms. The comparison between Figs. 4 and 5 shows that the time cost of non-linear algorithms grows
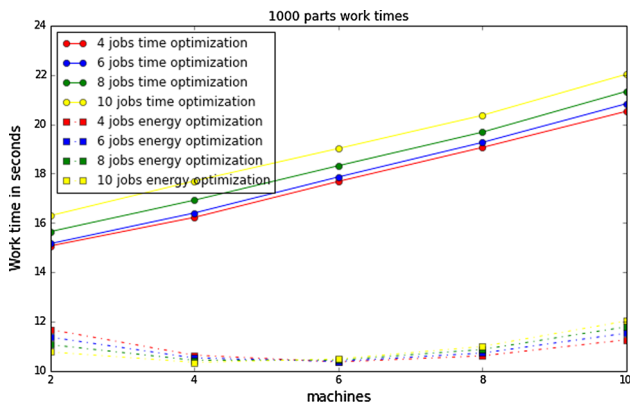
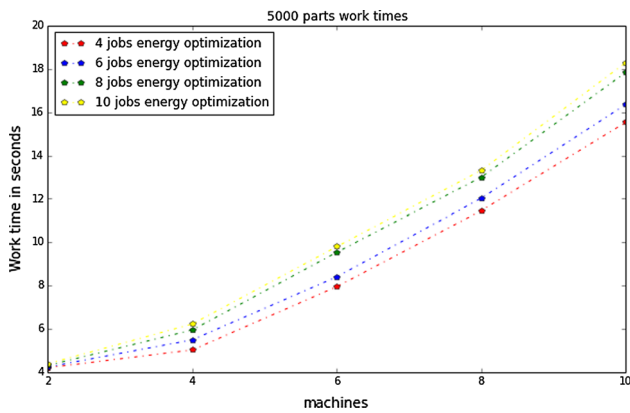**Fig. 3** Comparison between energy optimization and time optimization CPU work time for 1000 parts

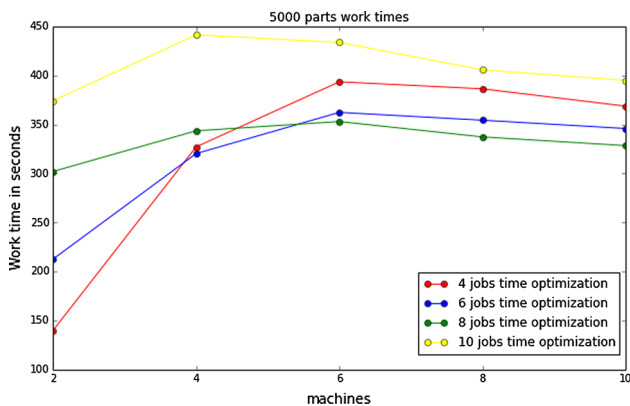**Fig. 4** Energy optimization CPU work time for 5000 parts

Fig. 5 Time optimization CPU work time for 5000 parts

rapidly, bringing them out of the range of practical applicability.

8 Optimization results

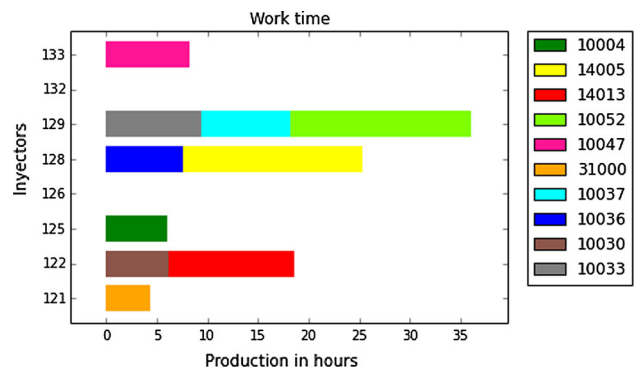Figure 6 shows an energy-optimal schedule for the parts to be produced with no maximum production time constraint.

Fig. 6 Cumulative time in hours for the production of a specific parts in different injectors—energy optimal scheduling without any time constraint
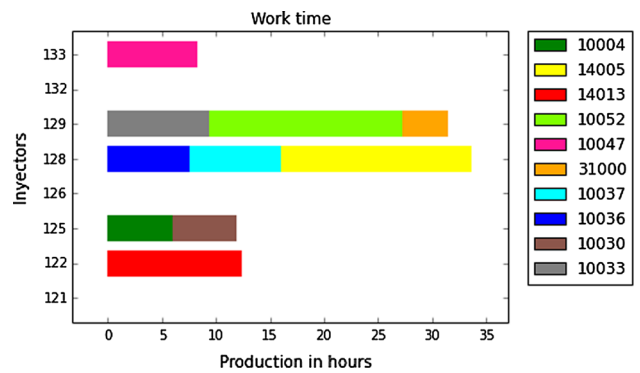

Fig. 7 Cumulative time in hours for the production of a specific parts in different injectors—time locally optimal scheduling without any time constraint

Figure 8 shows how the introduction of a constraint imposing a maximum production time of 27 h forces the energy-optimal scheduling to result in a balanced workload among three energetically efficient injectors.

In the procedure for time-optimal scheduling, the constraint imposing a maximum production time can be made inactive by specifying a number of hours ample enough for all pieces to be produced in any possible order. In this case, the scheduler converges as expected to a locally optimal schedule, as can be seen for example in Fig. 7.

When, on the other hand, the maximum production time constraint is imposed by specifying a number of total hours generated for instance by the fast energy-optimal schedule generator, then the production gets distributed in a more balanced way to all available machines, resulting in a better minimum to be reached for the makespan, as shown in Fig. 9. In the specific case of parts 10033 and 10052, the scheduler is not able to distribute the work because of the machine-part compatibility constraint according to the historical logs available.

Each optimization algorithm is adapted to the specific purpose for which it was designed. The energy-optimal
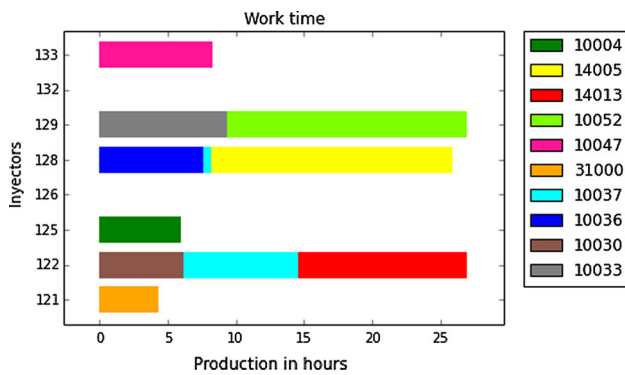
Fig. 8 Cumulative time in hours for the production of specific parts in different injectors—energy-optimal scheduling with a 27 h maximum time constraint
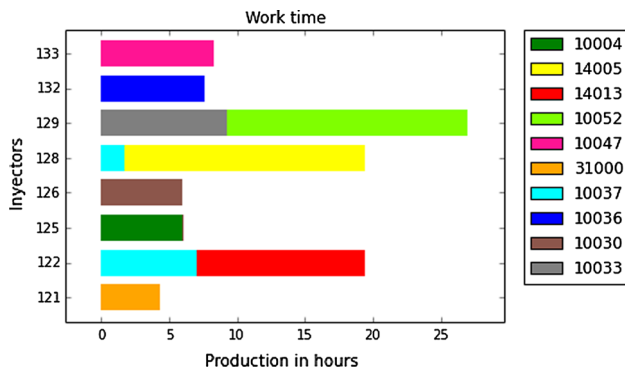

Fig. 9 Cumulative time in hours for the production of specific parts in different injectors—time-optimal scheduling with a 27 h maximum time constraint. In the specific case of parts 10,033 and 10,052, the scheduler is not able to distribute the work because of the machine-part compatibility constraint, based on the historical production logs available

scheduling algorithm is fast and converges to a global optimum for its objective function. The time-optimal scheduling algorithm by itself only converges to a local minimum. Introducing in its formulation a maximum production time constraint based on the results from the fast energy optimizer makes it converge to a lower minimum. This composition of the two methods makes it possible to rapidly obtain schedules that combine low energy footprint with low makespans for the production.

9 Interactive schedule visualization

The design and implementation of a Graphical User Interface front-end to the scheduling system makes its interactive usage possible. Planners can introduce the parts that they need produced, and any maximum production time constraints that are deemed relevant. The tool reads this input and the historical production logs, then calculates an optimal schedule that is then presented in a Gantt chart. Figure 10 shows a screen-
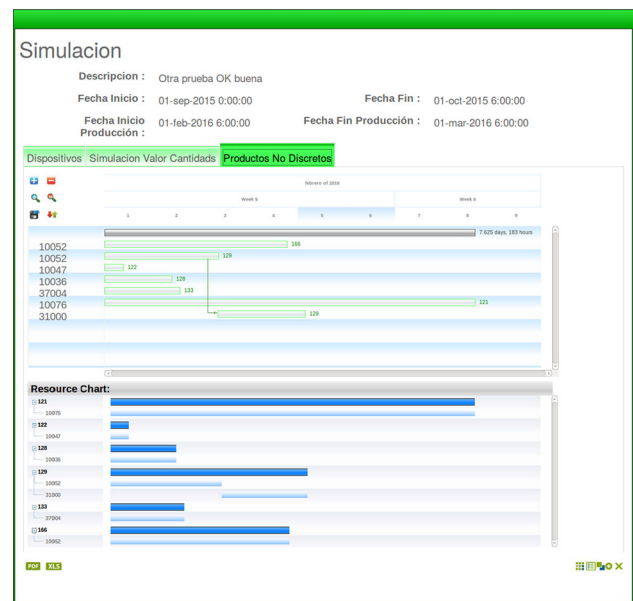

Fig. 10 Schedule representation in a Gantt chart

shot of the interface, which includes an interactive schedule visualization which is created based on the planning results.

10 Conclusions

The sensorization of manufacturing in Industry 4.0 can be leveraged for automated energy- and time-optimal production scheduling across multiple production lines.

We present an methodology aimed at this objective that focuses methodologically on optimization by linear programming and Bagged Binary Knapsack. Combining the results from an efficient energy-optimizing scheduler with a secondary time-optimizing method yields the best results.

The methodology is demonstrated and evaluated on real data from a sensorized manufacturing plant operating by metal injection molding procedures.

References

1. Abel, O., et al.: Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints. J. Process Control **10**(4), 351–362 (2000)
2. Adam, L., et al.: Application of high-performance computing to a bolt static tensile test. Int. J. Inter. Design Manufact. (IJIDeM) **6**(3), 195–203 (2012)
3. Amaya, Andrés Felipe Duque, Torres, Adalberto Gabriel Dıaz, Maya, Diego Andrés Acosta: First and second thermodynamic law analyses applied to ignition engines modeling and emission prediction. Int. J. Interact. Des. Manuf (2014)
4. Biskup, Dirk, Herrmann, Jan, Gupta, Jatinder N.D.: Scheduling identical parallel machines to minimize total tardiness. Int. J. Prod. Econ. **115**(1), 134–142 (2008)

5. Borghetti, Alberto, et al.: Optimal scheduling of a multiunit hydro power station in a short-term planning horizon. Int. Series Operat. Res. Manage. Sci. **212**, 167–181 (2015)

6. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)

7. Caccetta, L., Kulanoot, A.: Computational aspects of hard knapsack problems. Nonlinear Anal. **47**(8), 5547–5558 (2001). doi:10.1016/S0362-546X(01)00658-7

8. Cascini, Gaetano, et al.: Systematic design through the integration of triz and optimization tools. Procedia Eng. **9**, 674–679 (2011)

9. Chekuri, Chandra, Khanna, Sanjeev: A polynomial time approximation scheme for the multiple knapsack problem. SIAM J. Comp. **35**(3), 713–728 (2004)

10. Cheng, T.C.E., Sin, C.C.S.: A state-of-the-art review of parallel-machine scheduling research. Euro. J. Operat. Res. **47**(3), 271–292 (1990)

11. Cherifi, Ahmed, et al.: Methodology for innovative eco-design based on triz. Int. J. Inter. Design Manufact. (IJIDeM) **9**(3), 167–175 (2015)

12. Cousins, S., Kahn, M.: The visual display of temporal information. Art Intell Med 3 **6**(3), 341–357 (1991)

13. De Vroom, H.: Organization for economic cooperation and development (OECD). Encyclopedia of Statistical Sciences

14. Gen, Mitsuo, Lin, Lin: Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. J. Intell. Manufact. **25**(5), 849–866 (2014)

15. Helal, Magdy, Rabadi, Ghaith, Al-Salem, Ameer: A tabu search algorithm to minimize the makespan for the unrelated parallel machines scheduling problem with setup times. Int. J. Operat. Res. **3**(3), 182–192 (2006)

16. Khoury, Ibrahim, et al.: Towards an optimization methodology of a rough forged part taking into account ductile damage. Int. J. Inter. Design Manufact. (IJIDeM) **5**(4), 213–225 (2011)

17. Lin, Chi-Wei, Lin, Yang-Kuei, Hsieh, Han-Ting: Ant colony optimization for unrelated parallel machine scheduling. Int. J. Adv. Manufact. Technol. **67**(1–4), 35–45 (2013)

18. Lin, Yang-Kuei, Pfund, Michele E., Fowler, John W.: Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. Comp. Operat. Res. **38**(6), 901–916 (2011)

19. Manohar Priyadarshan, A., Acharya, Sushil, Wu, Peter: Enhancing manufacturing process education via computer simulation and visualization. J. Edu. Learn. **3**(3), 172–182 (2014)

20. Garey Michael, R., Johnson David, S.: Computers and intractability: a guide to the theory of np-completeness, WH Free. Co., San Fr (1979)

21. Nash, Stephen G.: A history of scientific computing. ACM, (1990)

22. Chee Khiang, Peng, et al.: Intelligent diagnosis and prognosis of industrial networked systems, vol. 44. CRC Press, Boca Raton (2011)

23. Rajesh, J.K., et al.: Livegantt: interactively visualizing a large manufacturing schedule. Chem. Eng. Sci. **56**(3), 999–1010 (2001)

24. Rajesh, J.K., et al.: Multi-objective optimization of industrial hydrogen plants. Chem. Eng. Sci. **56**(3), 999–1010 (2001)

25. Salonitis, Konstantinos, Ball, Peter: Energy efficient manufacturing from machine tools to manufacturing systems. Procedia CIRP **7**, 634–639 (2013)

26. Shen, W.M., Wang, L.H., Hao, Q.: Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. Systems, man, and cybernetics, part C: applications and reviews. IEEE Trans **36**(4), 563–577 (2006)

27. Srinivasan, Bala, Palanki, Srinivas, Bonvin, Dominique: Dynamic optimization of batch processes: I. characterization of the nominal solution. Comp. Chem. Eng. **27**(1), 1–26 (2003)

28. Bertrand Thompson, C.: The taylor system of scientific management. Hive Publishing Company, Easton, Maryland (1974)

29. Wilson, James M.: Gantt charts: a centenary appreciation. Euro. J. Operat. Res. **149**(3), 430–437 (2003)