Scheduling independent parallel machines with convex programming *†

Stéphane Chrétien‡, Nathalie Herr§, Jean-Marc Nicod§ and Christophe Varnier§
‡Laboratoire de Mathématiques, Université de Franche-Comté and UMR CNRS 6623
16 route de Gray, 25030 Besançon, France
Email: chretien@math.univ-fcomte.fr
§FEMTO-ST/AS2M, UBFC/UFC/CNRS/ENSMM, Besançon, France
Email: [nathalie.herr/jean-marc.nicod/christophe.varnier]@femto-st.fr

Abstract

In the field of production scheduling, this paper adresses the problem of maximizing the production horizon of a set of independent parallel machines. Each machine is considered to be able to provide several throughputs corresponding to different operating conditions and associated to different lifetimes. Convex optimization is used to define the contribution of each machine to a global needed throughput. A Mirror Descent for Saddle Points method is proposed to cope with the assignment problem. The considered model and the resolution method are first detailed. Results based on computational experiments are then provided.

Keywords: Convex optimization l_1 trend filtering Production scheduling Parallel machines

1 Introduction

The problem tackled in this paper concerns the scheduling of M heterogeneous parallel machines M_m $(1 \leq m \leq M)$, performing independent and identical tasks. All the machines are supposed to be of similar type and independent. A subset of machines has to be used in parallel to reach a given needed global throughput $\sigma(t)$. The total provided throughput corresponds to the sum of the contributions of machines that are currently running. All the machines are not supposed to be in use at any time because the target throughput can be reached by using only a subset of the machines within the platform or because some machines are not available. Machines are indeed assumed to suffer from wear and tear. Their lifetime is then limited and maintenance is required. Many reasons justify to postpone maintenance operations as late as possible and to maintain all the machines at the same time. Maintenance can for instance be challenging

*This work has been supported by the Labex ACTION project (contract "ANR-11-LABX-01-01") – Authors in alphabetic order

†This is the author's version of a work that was presented at the Gaspard Monge Program for Optimization - Conference on Optimization & Practices in Industry (PGMO-COPI'14), Paris, France, October 28-31, 2014

and costly [10]. Isolated or embedded equipments can also require to wait for the end of a global task before performing maintenance [1], for example in the aerospace, the railway or the automobile domain. One challenging objective is then to maximize the production horizon of the set of machines between two maintenance periods. This production horizon corresponds to the lifetime of the whole set of machines. This global lifetime depends itself on machines lifetimes, but also on the decisions performed during the scheduling process. Each machine lifetime is indeed assumed to be variable and dependent on its use. A machine lifetime can be extended when the machine is used in less stressful operating conditions than nominal ones. The provided throughput f_m is then decreased, but is available for a longer time (see Figure 1). Available throughputs and their associated lifetime can be determined at each time in a Prognostics and Health Management (PHM) context, in which, based on a machine monitoring, a prognostics phase allows to estimate the Remaining Useful Life (RUL) of machines [11], depending on their past and future usage.

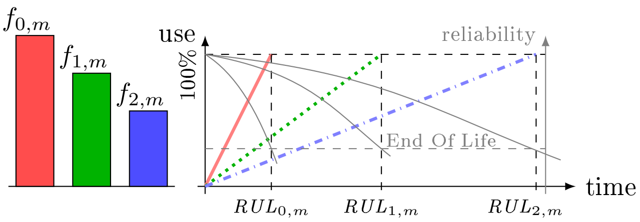

Figure 1: Throughputs and associated Remaining Useful Lives (RUL) for a machine with three different operating conditions

It has been shown in [13], [6] and [7] that a platform useful life can be extended by managing the usage of machines thanks to the knowledge of each machine remaining useful life. In these studies, machines throughputs have been considered to be in a discrete domain. Some complexity results have been proposed in [7] for different classes of the decision problem. The problem can be solved in polynomial time under some restrictive assumptions, while it turns out to be NP-complete in the general case. An optimal formulation through an Integer Linear Program has been detailed in [13]. As the decision problem has been proven to be NP-complete, optimal solutions can be found in limited time only for small size instances considering a very limited number of machines, very few throughput values and short production horizons. For larger problems, many polynomial heuristics have been provided in [6] and [7] to cope with the problem of maximizing a platform useful life under service constraint. Efficiency of these heuristics have been assessed through exhaustive simulations, but results remain suboptimal.

The study proposed here makes use of convex programming to obtain optimal solutions for large instances of a similar decision problem. The objective remains the same, but the model considered for machines is more complex. Each machine M_m is supposed to be able to provide a throughput f_m that can vary continuously and take any value within a given interval $[f_{min,m}; f_{max,m}]$ (see Figure 2). The maximal throughput is more efficient in terms of output, but is associated to a minimal lifetime. This maximal throughput is furthermore supposed to be decreasing with time. A lower throughput is less efficient, but allows to reach a longer operational time. Considering discretized time, the problem consists then in selecting, for each period of time t, a subset of machines to be used and an associated throughput f_m for each of them, with the maximization of the production horizon as objective. Such a model can be applied to fuel cells, which use appears to be of growing interest for power generation [8]. This technology offers indeed a potential alternative to conventional power systems. Several applications can be found in the literature, as the supply of energy to telecommunication stations [15] or to electric vehicles [14, 16]. Regarding this application, a continuous use of machine will be observed during the schedule. Change of throughput will still be authorized, but the number of scheduled shutdowns will be minimized for each machine. Starting and stopping a fuel cell can indeed induce considerable damage [2].

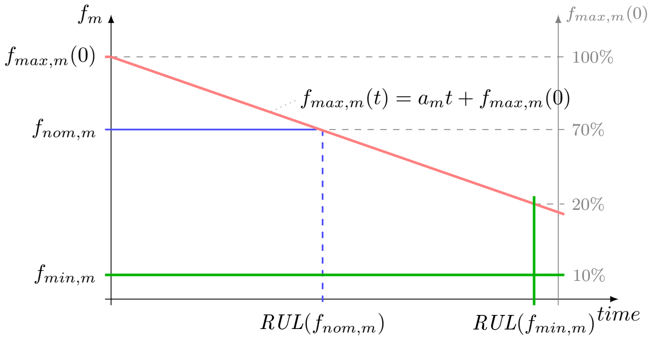

Figure 2: Model considered for the machine throughputs and their associated Remaining Useful Lives (RUL)

In the proposed approach, we look for solutions with small variations satisfying a set of natural constraints on the individual throughput of each machine. For a machine M_m, $m = 1, \ldots, M$, the throughput is decomposed into two components, a first one being assumed to be piecewise constant with rare jumps and a second one assumed to be piecewise affine with small slope and rare slope changes inside the makespan. In order to solve this type of problems in the framework of convex optimization, we introduce a penalized optimization problem which incoporates ℓ_1 and ℓ_∞ norms of the various quantities described above, leading to solutions with sparse first derivative for the first component and sparse second derivative for the second, while uniformly controlling the slopes. This nonsmooth penalized approach is the subject of extensive research in the machine learning, computational statistics and signal processing communities. The ℓ_1 penalization approach was recently advertized for the solution of the sparsest solution of an under-dertermined system of linear equations [5] and for the piecewise affine approximation [9]. These ideas have lead to very important discoveries in the fields of mathematics and computer science in relation to the frontier between P and NP (see [4] and references therein). We exploit this set of tools for our relaxation of the scheduling problem.

The organization of the paper is as follows: a first approach for a similar scheduling problem is first proposed in Section 2. The problem tackled in this article is then formulated as the solution of a convex program in Section 3. The resolution method is detailed in Section 4 and results based on computational experiments are provided in Section 4.5. The work is finally concluded in Section 5.

2 An integer programming model

In this section, we propose a first approach for the scheduling problem with an exact resolution method, based on an Integer Linear Program (ILP). This formulation was proposed in [13] for machines which can be used with n running profiles defined in a discrete throughput domain (see Figure 1).

2.1 Decision problem

The considered decision problem can be described as follows: does exist a schedule to achieve the given service σ during a given number of time periods T, considering the current health state of all the machines, i.e., the value of $\{RUL_{i,m} \text{ s.t. } 0 \leq i \leq n-1 \text{ and } 1 \leq m \leq M\}$? For this first problem, denoted DP_T, Nicod et al. [13] have proposed an Integer Linear Program ($ILP(DP_T)$) which we describe below.

2.1.1 Variables

Let M be the number of machines and T the horizon. We assume that at time $t \in \{1, \dots, T\}$, each machine M_m may be running with n different running profiles associated to throughput levels, denoted by $\rho_{i,m}$. Let $a_{i,m,t}$, $0 \leq i \leq n-1, 1 \leq m \leq M$ and $0 \leq t \leq T$, be the variables of the decision problem. For each (i, m, t), $a_{i,m,t}$ is defined as a binary variable:

$$a_{i,m,t} = \begin{cases} 1 \text{ if equipment } m \text{ is used with the profile } i \text{ during the period } t \\ \\ 0 \text{ otherwise.} \end{cases}$$

Using these variables involves that machines are supposed to be able to jump from any throughput value to any other at each period. Some smoothing can be desirable and will be taken into account in our new proposal described in the next section.

2.1.2 Constraints

The constraints of the decision problem DP_T should expressed first the production throughput required, the limitation of the useful life and the possible control profile for machines.

- The first set of constraints concerns the production throughput. At least the required service σ should be reached for each time period. This can be expressed by the following inequalities:

$$\forall t \in \{0, \dots, T\} \quad \sum_{m=1}^{M} \sum_{i=0}^{n-1} a_{i,m,t} \cdot \rho_{i,m} \geq \sigma \tag{1}$$

- The second set of constraints requires that if a machine is used for a given period, then it runs with only one profile $\rho_{i,m}$:

$$\sum_{i=0}^{n-1} a_{i,m,t} \leq 1,\ m = 1, \ldots, M \text{ and } t = 0, \ldots, T. \tag{2}$$

- Finally, the last set of constraints is due to the remaining useful life for each machine. We can consider that during a given period t, if a machine with index m is used with the running profile $\rho_{i,m}$, then it cuts the remaining useful life by $\delta_T / RUL_{i,m}$, where δ_T is the time length of one period. Consequently, due to the value of the remaining useful life for machine M_m, the following inequalities expressed that each machine could not be used more than its remaining useful life:

$$\sum_{i=0}^{n-1} \frac{\sum_{t=1}^{T} a_{i,m,t} \times \delta_T}{RUL_{i,m}} \leq 1,\ m = 1, \ldots, M. \tag{3}$$

2.2 Associated optimization problem

The previous described Integer Linear Program, denoted $ILP(\rho, \mathcal{M}, T)$ allows without any objective function to answer the question: does exist a configuration of all the machines such that the required throughput σ could be reached during at least T periods?

As presented earlier, we propose to solve the problem where the set of machines is able to produce the throughput demand σ as long as possible. Besides the previous model can compute a solution to reach a demand σ, it is not sufficient since it is designed for a given number of periods T. Nevertheless it can be useful to determine the greatest number of periods during which a given set of machines is able to produce the given throughput σ. First, one can determine two bounds of this number. The first one is an upper bound T_{max}:

$$T_{max} = \left\lfloor \frac{\sum_{m=1}^{M} \max_{i=0,\ldots,n-1} \left(\rho_{i,m} \times RUL_{i,m}\right)}{\sigma} \right\rfloor \tag{4}$$

This equation means that if all the machines are used with their better yield (the running profile that provides the greatest production during the whole remaining useful life) and the global production demand σ is constant in time, then T_{max} is the longest duration for which the throughput σ can be reached. A lower bound T_{min} can also be computed using a heuristics algorithm. Then, the worst lower bound will be 0. If a heuristics can provide a solution, the latter could be considered as a better lower bound. Since one can compute these two bounds, T_{max} and T_{min}, finding the maximum number of periods that can be reached for a given throughput demand σ and a given set of machines can be done using a dichotomy search approach. This approach is detailed in algorithm 1.

2.3 Drawbacks of this combinatorial approach

The combinatorial optimization approach described in this section can be solved for small instances with e.g. standard software for integer linear programs. However, in many applications, one is interested in dealing with problems where the

---

Algorithm 1: Dichotomy search procedure for finding the maximum number of periods

---

Remark: for this algorithm, we call $ILP(\rho, \mathcal{M}, T)$ the integer linear program described in section 2 and $LP(\rho, \mathcal{M}, T)$ the rational relaxation of $ILP(\rho, \mathcal{M}, T)$

$T_{min} \leftarrow$ *lower bound*

$T_{max} \leftarrow$ *upper bound*

while $T_{max} - T_{min} > 0$ **do**

  $T \leftarrow (T_{min} + T_{max})/2$

  if $LP(\rho, \mathcal{M}, T)$ *has a solution* **then**

    if $ILP(\rho, \mathcal{M}, T)$ *has at least one solution* **then**

      $T_{min} \leftarrow T$

    else

      $T_{max} \leftarrow T$

  else

    $T_{max} \leftarrow T$

return T

---

horizon and/or the number of machines may be quite large. In such instances, the Integer Linear Programming approach may be intractable and one may have to resort to heuristics or relaxations. In the next section, we present a convex programming approach based on sparsity promoting penalization functionals for a related scheduling problem. The main advantage of the convex programming approach is that it can be solved in polynomial computational time and thus, can be added various constraints that could have been yet less tractable via the ILP formulation.

3 Formulating the problem as the solution of a convex program

In this section, we introduce a new formulation of the scheduling problem leading to a convex programming problem. The new formulation may not be considered as a relaxation of the approach described in the last section, but is an example of how one may address practical but usually intractable problems more efficiently by completely changing our paradigm.

The scheduling problem and the considered objective remain the same, but the model taken into account for machines is different. While running profiles were defined in a discrete throughput domain in the previous detailed approach (see Section 2), machines are now supposed to be able to provide a throughput f_m that can vary continuously and take any value within a given interval $[f_{min,m}; f_{max,m}]$ (see Figure 2).

3.1 Model

We consider the problem of scheduling a set of M machines to produce enough energy to satisfy a given demand $\sigma(t)$ which evolves over time. For the machine with index m, we define a function $f_m(t)$, $t = 0, \ldots, T$, which is the throughput that the m^{th} machine contributes. The relationship with the combinatorial approach of the previous section is straightforward: $f_m(t) = \sum_{i=0}^{n-1} a_{i,m,t}\rho_{i,m}$.

Our main constraint is that

$$\sum_{m=1}^{M} f_m(t) \geq \sigma(t) \quad \text{for all } t \text{ over the time span } \{0,\dots,T\} \tag{5}$$

Let us assume that each $f_m(t)$, $m = 1,\dots,M$, can be decomposed as follows:

$$f_m(t) = f_{1,m}(t) + f_{2,m}(t) \tag{6}$$

where $f_{1,m}(t)$ is piecewise constant and $f_{2,m}(t)$ is piecewise linear. Each time where $f_{2,m}(t)$ changes its slope will be called a breackpoint. Each function $f_{c,m}$ satisfies

$$f_{c,m}(t) \geq 0 \quad \text{for } c \in \{1,2\} \text{ and for all } t \in \{0,\dots,T\} \tag{7}$$

For each machine M_m, we will also impose the following upper bound:

$$f_m(t) \leq f_{\max,m}(t) \quad \text{for all } m = 1,\dots,M \text{ and for all } t = 0,\dots,T. \tag{8}$$

This upper bound corresponds to a maximal throughput, which typically declines gradually during the use of a machine M_m.

A certain consumption rate constraint is set for each machine M_m $m = 1,\dots,M$ and may be written as:

$$\sum_{t=0}^{T} \Phi(f_m(t)) \leq 1 \tag{9}$$

with Φ a convex function. These consumption constraints express the limited lifetime of each machine.

3.2 Main idea

Our goal is to find the functions $f_{c,m}$, $c = 1,2$ and $m = 1,\dots,M$ using convex optimization, so that the solution can be found in polynomial time. The main idea is to use an approach which was recently promoted in signal processing and computational statistics. In [9], Kim, Koh, Boyd and Gorinevsky showed the practical interest of minimizing the ℓ_1-norm for obtaining sparsity in the context of function modeling over time. More precisely, they showed through multiple experiments that minimizing the ℓ_1-norm of the finite differences of a vector leads, under very mild conditions, to a vector which is piecewise constant. The same idea can be used to obtain polynomially shaped (of any order) vectors which can be interpreted as the discretized version of a polynomial function of time.

The main ingredient in our proposal is to model the functions f_m by a sum $f_{1,m} + f_{2,m}$ of a piecewise constant function and a function which has uniformly controlled slopes. We will impose that the number of jumps be small and the slope of $f_{1,m}(t)$ be small too as well as the number of breakpoints. Let $\Delta : \mathbb{R}^{T+1} \mapsto \mathbb{R}^T$ denote the operator which takes the successive differences, i.e.

$$f = \begin{bmatrix} f(0) \\ \vdots \\ f(T) \end{bmatrix} \qquad \Delta f = \begin{bmatrix} f(1) - f(0) \\ \vdots \\ f(T) - f(T-1) \end{bmatrix} \tag{10}$$

Under the constraints, many throughputs may be feasible for a given horizon T. Optimizing the horizon may provide solutions which have unreasonable shapes with respect to the physical properties of the machines.

In particular,

- All the machines may not be involved at any time. Thus, we expect to see *working windows* occuring in the solution. This is modelled by the *sparsity* of f_m and the first derivative of $f_{1,m}$;
- The solution may be very oscillatory. In order to overcome such bad feature, one may allow small variations of the individual throughputs which can be modelled by imposing that $f_{2,m}$ is piecewise polynomial with em sparse slope changes and small absolute slope.

These problems could be addressed by imposing the sparsity of the various components to be *less than prescribed by certain physical constraints.* A bad news is that sparsity is *not convex* and leads to **NP-hard** feasibility problems.

A simple solution can be incorporated into the approach. The idea is to replace sparsity by a *convex surrogate.* There is a field where such surrogates have proved very efficient: signal processing [5], [9]. In most cases, the corresponding relaxation sums up to minimize the ℓ_1-norm of the quantity whose sparsity is to be controlled.

Using the ℓ_1 penalization approach, one obtains that our problem can be addressed via optimizing the composite function:

$$\phi(F) = \sum_{m=1}^{M} \lambda_{1,m} \|\Delta f_{1,m}\|_1 + \lambda_{2,m} \|\Delta f_{2,m}\|_\infty + \lambda_{2',m} \|\Delta^2 f_{2,m}\|_1 \qquad (11)$$

subject to the constraints (5), (6), (7), (8) and (9). Each term in this objective function is a penalty for imposing a certain sparsity on $\Delta f_{c,m}$ or $\Delta^2 f_{c,m}$, $c \in \{1,2\}$, $m = 1,\ldots,M$. $\|\Delta f_{1,m}\|_1$ is used to minimize the discontinuity of the final solution, $\|\Delta f_{2,m}\|_\infty$ minimizes the slopes of the linear parts of f_m and $\|\Delta^2 f_{2,m}\|_1$ minimizes the number of slope changes.

The main interest in using such penalties is that they are well structured convex and are thus amenable to efficient methods of convex optimization.

In order to enforce that the $f_{c,m}$ equal zero more often than would lead the previous objective, one can propose the following objective function

$$\mathcal{F}(F) = \|F\|_1 + \sum_{m=1}^{M} \lambda_{1,m} \|\Delta f_{1,m}\|_1 + \lambda_{2,m} \|\Delta f_{2,m}\|_\infty + \lambda_{2',m} \|\Delta^2 f_{2,m}\|_1 \qquad (12)$$

4 Resolution method: a saddle point mirror descent algorithm

4.1 Intermediate definitions

Consider first the entropy function $h(x) = \sum_{i=1}^{d} x_i \ln(x_i)$. We have then:

$$\nabla h(x) = \begin{bmatrix} \ln(x_1) + 1 \\ \vdots \\ \ln(x_d) + 1 \end{bmatrix} \tag{13}$$

Let us then introduce the following functions which adequately describe our constraints:

$$\begin{array}{lcl}
\psi_0 : \mathbb{R}^{2M(T+1)} \mapsto \mathbb{R}^{T+1} & : & \psi_0(F) = \sum_{m=1}^{M} f_m - \sigma \\
\psi_{c,m} : \mathbb{R}^{2M(T+1)} \mapsto \mathbb{R}^{T+1} & : & \psi_{c,m}(F) = f_{c,m}, \quad c \in \{1,2\},\ m = 1,\ldots,M \\
\psi_{3,m} : \mathbb{R}^{2M(T+1)} \mapsto \mathbb{R}^{T+1} & : & \psi_{3,m}(F) = f_{\max,m} - f_m \\
\psi_{4,m} : \mathbb{R}^{2M(T+1)} \mapsto \mathbb{R}^{M} & : & \psi_{4,m}(F) = 1 - \sum_{t=0}^{T} \Phi(f_m(t)), \quad m = 1,\ldots,M
\end{array}$$

with

$$F^t = \left[f_{1,1}(0), f_{2,1}(0), f_{1,2}(0), f_{2,2}(0), \ldots, f_{1,M}(0), f_{2,M}(0), \ldots, f_{1,1}(T), f_{2,1}(T), \ldots, f_{1,M}(T), f_{2,M}(T)\right].$$

4.2 Mirror Descent for Saddle Points

A mirror descent method is proposed to cope with the problem of minimizing the objective function proposed in Equation (12).

The mirror descent algorithm is a method first proposed by Nemirovskii and Yudin [12] for convex programming. It has been extensively studied recently and several relationships have been discovered between the mirror descent scheme and Bregman-proximal methods. We refer the interested reader to [3] for a detailed and very pedagogical description of mirror descent algorithms.

4.2.1 Main ideas

Assume for a moment that $\mathcal{F}$ is differentiable. The idea behind mirror descent algorithms is very simple.

The problem can be described as follows:

$$\underset{F \in \mathbb{R}^{2M(T+1)}}{\operatorname{argmin}} \left(\|F\|_1 + \phi(F)\right) \tag{14}$$

such that $\psi_0(F) \geq 0$ and $\psi_{K,m}(F) \geq 0$, $\forall\, K = 1,\ldots,4$ and $\forall\, m = 1,\ldots,M$. Let us denote the constraint set by $\mathcal{C}$. The standard projected gradient algorithm is of the form

$$F^{(l+1)} = P_{\mathcal{C}}\left(F^{(l)} - \lambda^{(l)}\, \nabla\mathcal{F}(F^{(l)})\right) \tag{15}$$

where $P_{\mathcal{C}}$ is the projection operator onto the set $\mathcal{C}$.

A mirror function is a convex function whose gradient is one-to-one and has a defining set which may conveniently incorporate simple constraints. Let θ be such a function. Then, the mirror descent iteration is given by

$$\begin{aligned} \nabla\theta(G^{(l+1)}) &= \nabla\theta(F^{(l)}) - \lambda^{(l)}\, \nabla\mathcal{F}(F^{(l)}), \\ F^{(l+1)} &= P_{\mathcal{C}}(G^{(l+1)}). \end{aligned}$$

As a very useful example, one might consider θ, a mirror map on $R^{2M(T+1)}$, defined by:

$$\theta(F) = \sum_{t=0}^{T}\sum_{m=1}^{M}\sum_{c=1}^{2} F(c,m,t)\ln(F(c,m,t)). \tag{16}$$

In accordance with the gradient of the entropy function defined in Equation (13), we have then:

$$\nabla\theta(F) = \ln(F) + 1 \tag{17}$$

4.2.2 The saddle point mirror descent algorithm

The main difficulty in the mirror descent scheme is that projecting onto the constraint set $\mathcal{C}$ might not be so easy. In order to overcome this problem, one possibility is to consider an algorithm which solves the primal-dual saddle point problem for the Lagrange function. For this purpose, define the Lagrange function as

$$L(F,u) = \|F\|_1 + \phi(F) + \langle u_0, \psi_0(F)\rangle + \sum_{m=1}^{M}\left(\langle u_{3,m}, \psi_{3,m}(F)\rangle + \langle u_{4,m}, \psi_{4,m}(F)\rangle\right) \tag{18}$$

and find a saddle point of this function under the constraint that $u \leq 0$. One can propose the following mirror descent scheme :

$$\nabla\theta_{a,b}(F^{(l+1)}, u^{(l+1)}) = \nabla\theta_{a,b}(F^{(l)}, u^{(l)}) - \eta\left(\begin{bmatrix} \nabla_F L(F^{(l)}, u^{(l)}) \\ -\nabla_u L(F^{(l)}, u^{(l)}) \end{bmatrix}\right) \tag{19}$$

where

$$\begin{aligned} \theta_{a,b}(F,u) &= a\sum_{t=0}^{T}\sum_{m=1}^{M}\sum_{c=1}^{2} F(c,m,t)\ln(F(c,m,t)) \\ &\quad + b\left(\sum_{m=1}^{M}\sum_{t=0}^{T} u_{0,m,t}\ln(u_{0,t}) + (u_{3,m,t})\ln(u_{3,m,t}) + \sum_{m=1}^{M} u_{4,m}\ln(u_{4,m})\right) \end{aligned} \tag{20}$$

4.3 Theoretical garantees

The saddle point mirror descent method has been analyzed by various authors. One interesting result is presented in [3, Theorem 5.1].

Theorem 1. *Assume that L is Lipschitz in both variables with constant L_F in the first variable and constant L_u in the second variable. Assume that there exist some positive constants R_F and R_u such that $\theta_{a,b}$ satisfies*

$$\sqrt{\sup_u \sup_F \theta_{a,b}(F,u) - \inf_F \theta_{a,b}(F,u)} \leq R_F \tag{21}$$

$$\sqrt{\sup_F \sup_u \theta_{a,b}(F,u) - \inf_u \theta_{a,b}(F,u)} \leq R_u. \tag{22}$$

Take $a = L_F/R_F$ and $b = L_u/R_u$ and $\eta = \sqrt{2/t}$. Then

$$\inf_F L\left(F, \frac{1}{t}\sum_{l=1}^{t} u^{(l)}\right) - \sup_u L\left(\frac{1}{t}\sum_{l=1}^{t} F^{(l)}, u\right) \leq (R_F L_F + R_u L_u)\sqrt{\frac{2}{t}}.$$

4.4 Maximizing the horizon

The proposed algorithm allows to search for a solution F with a fixed horizon. Since the main objective is to maximize the production horizon, a dichotomic search approach is used to determine the maximal horizon for which a solution exists.

4.5 Some preliminary simulation results

We ran the algorithm on simple toy instances with $M = 5$ and $T = 100$. An example is plotted in Figure 3 below. In accordance with the model for the machines taken into account in this contribution and illustrated in Figure 2, the function Φ used in the consumption rate constrains (see Equation 9) has been defined as follows:

$$\Phi(f_m(t)) = \frac{a_m}{f_{1,m}(t) + f_{2,m}(t) - f_{max,m}} \quad \text{for all } m = 1, \ldots, M \text{ and for all } t = 0, \ldots, T \tag{23}$$

with a_m the slope of the decreasing function associated to the maximal throughput for machine with index m.

One may easily notice the shapes obtained for the solution achieve the desired goal since the first component has a sparse gradient and the second component has a sparse second order discrete derivative. As such, the first component is piecewise constant with rare jumps and the second component is piecewise affine with rare slope changes. Tuning the relaxation parameters $\lambda_{1,m}$, $\lambda_{2,m}$ and $\lambda_{2'}$, $m = 1, \ldots, M$ can be easily done by simply choosing them independent of m and trying several values until an appropriate shape to the taste of the user is obtained. One might also address this issue using automatic selection methods.

5 Conclusion and future work

In this paper, we proposed a convex optimization method to solve the problem of scheduling M heterogeneous parallel machines M_m ($1 \leq m \leq M$), performing independent and identical tasks. We introduced a simple mirror descent algorithm to find the optimal solution.

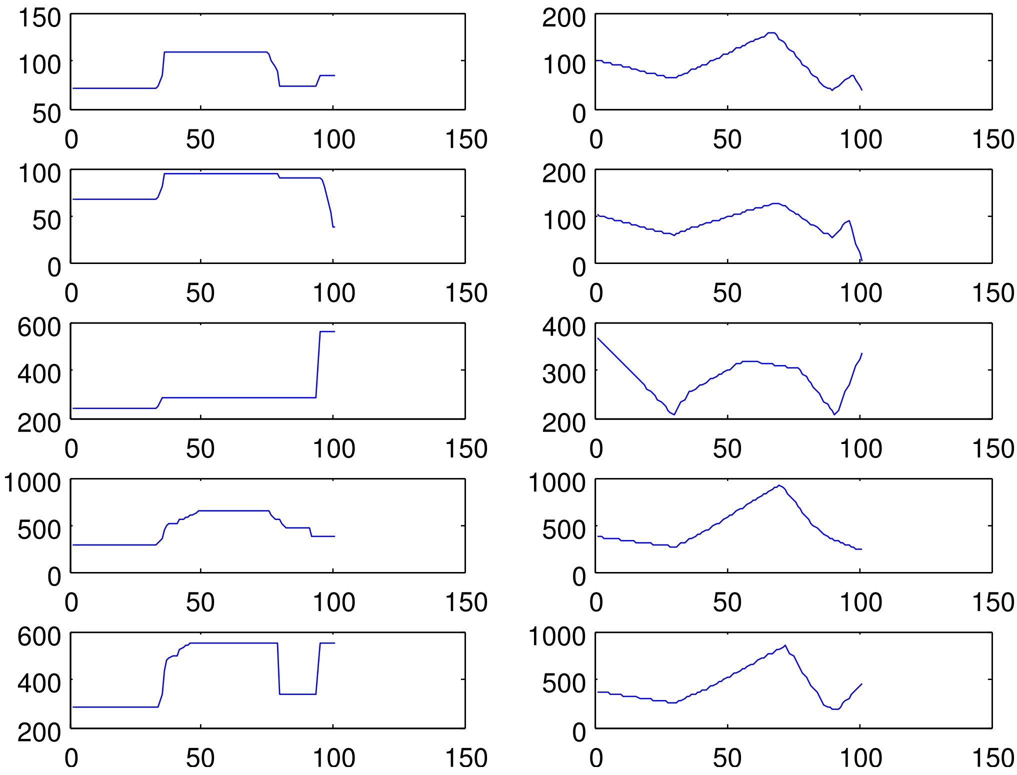

Figure 3: Solutions on a toy example with $M = 5$ and $T = 100$.

Future further work is needed to accelerate the optimization method. One simple approach is to use the variational representation of the various norms involved in the objective function to perform a smarter descent scheme. A relevant approach could be to use a saddle point mirror prox scheme such as introduced by Nemirovskii [3, Section 5.2.3]. Such a method allows to obtain a convergence speed of the order of $1/t$ instead of $1/\sqrt{t}$ for the simpler mirror descent proposed here. Using such an algorithm, we will be able to perform large scale simulations and provide extensive Monte Carlo simulations which will prove the model to be useful far beyond the framework previously handled using combinatorial optimization.

6 Appendix: computational details

The gradient of $F^{(l+1)}$ will be considered to optimize the function defined in Equation 12. Each term will then be considered successively.

In what follow, the subdifferential of a convex function ϕ at F will be denoted by $\partial\phi(F)$. We will also denote by $\nabla\phi(F)$ any subgradient of ϕ at F. Of course, when ϕ is differentiable, $\nabla\phi(F)$ is the usual gradient of ϕ at F.

First, notice that the sub-gradients of $\theta_{a,b}$ are given by the following formulae

$$\begin{aligned}
\nabla\theta_{a,b}(F^{(l)},u) &= a\left(\ln(F^{(l)})+1\right)\\
\nabla\theta_{a,b}(F,u_0^{(l)}) &= -b\left(\ln(-u_0^{(l)})-1\right)\\
\nabla\theta_{a,b}(F,u_{3,m}^{(l)}) &= -b\left(\ln(-u_{3,m}^{(l)})-1\right)\\
\nabla\theta_{a,b}(F,u_{4,m}^{(l)}) &= -b\left(\ln(-u_{4,m}^{(l)})-1\right).
\end{aligned}$$

On the other hand, the gradients of the Lagrange function are given by

$$\begin{aligned}
\nabla_F L(F,u) &= \nabla_F\|F\|_1+\sum_{m=1}^{M}\left(\lambda_{1,m}\nabla_F\|\Delta f_{1,m}\|_1+\lambda_{2,m}\nabla_F\|\Delta f_{2,m}\|_\infty+\lambda_{2',m}\nabla_F\|\Delta^2 f_{2,m}\|_1\right)\\
&\quad+\sum_{t=0}^{T}u_{0,t}\nabla_F\psi_{0,t}(F)+\sum_{m=1}^{M}\sum_{t=0}^{T}u_{3,m,t}\nabla_F\psi_{3,m,t}(F)+\sum_{m=1}^{M}u_{4,m}\nabla_F\psi_{4,m}(F)
\end{aligned}$$

and

$$\begin{aligned}
\nabla_u L(F,u_0) &= \sum_{m=1}^{M} f_m-\sigma\\
\nabla_u L(F,u_{3,m}) &= f_{\max,m}-f_m\\
\nabla_u L(F,u_{4,m}) &= 1-\sum_{t=0}^{T}\Phi(f_m(t))
\end{aligned}$$

Finally, iteration l is completed once we take the inverse of the mirror map:

$$(F^{(l+1)},u^{(l+1)}) = \exp\left(D_{a,b}^{-1}\,\nabla\theta_{a,b}(F^{(l+1)},u^{(l+1)})-1\right), \tag{24}$$

where

$$D_{a,b} = \begin{bmatrix} D(a\cdot e_{c\times M\times(T+1)}) & 0\\ 0 & B(b\cdot e_{M\times(T+1)+M})\end{bmatrix} \tag{25}$$

where for all d, e_d denotes the vector of all ones in $\mathbb{R}^d$.

6.1 A subgradient of $\|F\|_1$

Recall that

$$\|F\|_1 = \left\|\left[f_{1,1},f_{2,1},\ldots,f_{1,M},f_{2,M}\right]^t\right\|_1 = \sum_{\substack{c=1,2\\ m=1,\ldots,M}}|f_{c,m}| \tag{26}$$

One possible gradient for $\|F\|_1$ is defined by the following function:

$$x\mapsto \operatorname{sign}(x)=\begin{cases}-1 & \text{if } x<0\\ 0 & \text{if } x=0\\ 1 & \text{if } x>0.\end{cases} \tag{27}$$

Then,

$$\nabla_{f_{c,m}}\|F\|_1=\left[\operatorname{sign}(f_{1,1}),\operatorname{sign}(f_{2,1}),\ldots,\operatorname{sign}(f_{1,M}),\operatorname{sign}(f_{2,M})\right]^t. \tag{28}$$

6.2 A subgradient of $\|\Delta f_{1,m}\|_1$

$\Delta^t \text{sign}(\Delta f_{1,m})$ is a subgradient in $f_{1,m}$, with Δ the function previously defined in Equation 10.

Since

$$\Delta\,\text{sign}(\Delta f_{1,m}) = \begin{bmatrix} -1 & 1 & 0 & \dots & & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ 0 & \dots & & \dots & & 0 \\ \vdots & & \vdots & & & \vdots \\ 0 & \dots & & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(T+1)\times T}, \tag{29}$$

we have

$$\nabla_{f_{c,m}} \|\Delta f_{1,m}\|_1 = \Delta^t \text{sign}(\Delta f_{1,m}) = \begin{bmatrix} -1 & 0 & 0 & \dots & & 0 \\ 1 & -1 & 0 & & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & & \vdots & & & \vdots \\ 0 & \dots & & 0 & 1 & -1 \\ 0 & & \dots & & 0 & 1 \end{bmatrix} \in \mathbb{R}^{T\times(T+1)}. \tag{30}$$

6.3 A subgradient of $\|\Delta f_{2,m}\|_\infty$

Using the variational formulation of the ℓ_∞-norm and the chain rule of subdifferential calculus, we have

$$\nabla_{f_{c,m}} \|\Delta f_{2,m}\|_\infty = \Delta^t y^*(\Delta f_{2,m}), \tag{31}$$

with

$$y^*(\Delta f_{2,m}) = \left[0, 0, \dots, 0, \text{sign}((\Delta f_{2,m})_{\max}), 0, \dots, 0\right]^t. \tag{32}$$

6.4 A subgradient of $\|\Delta^2 f_{2,m}\|_1$

Using the variational formulation of the ℓ_1 norm, we easily obtain

$$\nabla_{f_{c,m}} \|\Delta f_{2,m}\|_1 = \Delta'^t \Delta^t \text{sign}(\Delta' \Delta f_{2,m}), \quad \text{with} \quad \Delta' \in \mathbb{R}^{T\times(T-1)} \tag{33}$$

6.5 A subgradient of $\|F\|_1 + \phi(F)$

Let $G(F)$ be a part of the objective function and defined by $F \mapsto \|F\|_1 + \phi(F)$. Considering the previous developments, we have:

$$G(F) = \begin{bmatrix} \text{sign}(f_{1,1}) & +\lambda_{1,1}\Delta^t \text{sign}(\Delta f_{1,1}) & \\ \text{sign}(f_{2,1}) & +\lambda_{2,1}\Delta^t y^*(\Delta f_{2,1}) & +\lambda_{2',1}\Delta^t \Delta'^t \text{sign}(\Delta' \Delta f_{2,1}) \\ \text{sign}(f_{1,2}) & +\lambda_{1,2}\Delta^t \text{sign}(\Delta f_{1,2}) & \\ \text{sign}(f_{2,2}) & +\lambda_{2,2}\Delta^t y^*(\Delta f_{2,2}) & +\lambda_{2',2}\Delta^t \Delta'^t \text{sign}(\Delta' \Delta f_{2,2}) \\ \vdots & \vdots & \\ \text{sign}(f_{1,M}) & +\lambda_{1,M}\Delta^t \text{sign}(\Delta f_{1,M}) & \\ \text{sign}(f_{2,M}) & +\lambda_{2,M}\Delta^t y^*(\Delta f_{2,M}) & +\lambda_{2',M}\Delta^t \Delta'^t \text{sign}(\Delta' \Delta f_{2,M}) \end{bmatrix} \in \mathbb{R}^{2M(T+1)} \tag{34}$$

6.6 Computation of $(F^{(l+1)}, u^{(l+1)})$

We have now collected all the necessary information for providing an explicit formula for the successive iterates. Indeed, we have:

$$(F^{(l+1)}, u^{(l+1)}) = \exp\left(D_{a,b}^{-1}\,\nabla\theta_{a,b}(F^{(l+1)}, u^{(l+1)}) - 1\right), \tag{35}$$

where $D_{a,b}$ is given by (25).

References

[1] Edward Balaban, Sriram Narasimhan, Matthew Daigle, José Celaya, Indranil Roychoudhury, Bhaskar Saha, Sankalita Saha, and Kai Goebel. A mobile robot testbed for prognostic-enabled autonomous decision making. In *Annual Conference of the Prognostics and Health Management Society*, 2011.

[2] Rod Borup, jeremy Meyers, Bryan Pivovar, Minoru Inaba, Kenichiro Ota, and Zempachi Ogumi et al. Scientific aspects of polymer electrolyte fuel cell durability and degradation. *Chemical reviews*, 107(10):3904–3951, 2007.

[3] S. Bubeck. *Theory of Convex Optimization for Machine Learning*. 2014.

[4] E. J. Candes. The mathematics of sparsity (and a few other things). In *Proceedings of the International Congress of Mathematicians*, 2014.

[5] D. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of National Academy of Science*, 100:2197–2202, 2003.

[6] Nathalie Herr, Jean-Marc Nicod, and Christophe Varnier. Prognostic decision making to extend a platform useful life under service constraint. In *Proceedings of IEEE International Conference on Prognostics and Health management*, Spokane, Washington, June 22-25, 2014. Best paper Award.

[7] Nathalie Herr, Jean-Marc Nicod, and Christophe Varnier. Prognostics-based scheduling in a distributed platform: Model, complexity and resolution. In *2014 IEEE International Conference on Automation Science and Engineering (CASE), Taipei, Taïwan*, pages 1054–1059, August 18-22 2014.

[8] Marine Jouin, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, and Noureddine Zerhouni. Prognostics and health management of pemfc - state of the art and remaining challenges. *International Journal of Hydrogen Energy*, 38:15307–15317, 2013.

[9] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1-trend filtering. *SIAM Review*, 51(2):339–360, 2009.

[10] Andras Kovacs, Gabor Erdos, Laszlo Monostori, and Zsolt Janos Viharos. Scheduling the maintenance of wind farms for minimizing production loss. In *Proceedings of the 18th IFAC World Congress, Milano (Italy)*, 2011.

[11] M. Lebold and M. Thurston. Open standards for condition-based maintenance and prognostic systems. In *5th Annual Maintenance and Reliability Conferance (MARCON), Gatlinburg, USA*, 2001.

[12] A. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

[13] Jean-Marc Nicod, Léo Obrecht, and Christophe Varnier. Task scheduling to extend platform useful life using prognostics. In *Proceedings of IFAC Conference on Manufacturing Modelling, Management, and Control, Saint Petersburg, Russia*, June 19-21, 2013.

[14] M. Tekin, D. Hissel, M.C. Pera, and J.M. Kauffmann. Energy management strategy for embedded fuel cell system using fuzzy logic. *IEEE Transactions on Industrial Electronics*, 54(1):595–603, 2007.

[15] Fu-Cheng Wang, Po-Chen Kuo, and Hsueh-Ju Chen. Control design and power management of a stationary pemfc hybrid power system. *International Journal of Hydrogen Energy*, 38:5845–5856, 2013.

[16] Keith Wipke, Tony Markel, and Doug Nelson. Optimizing energy management strategy and degree of hybridization for a hydrogen fuel cell suv. In *Proceedings of 18th Electric Vehicle Symposium, Berlin*, 2001.