

An Iterated Min–Max procedure for practical workload balancing on non-identical parallel machines in manufacturing systems

Quentin Christ, Stéphane Dauzere-Peres, Guillaume Lepelletier

▶ **To cite this version:**

Quentin Christ, Stéphane Dauzere-Peres, Guillaume Lepelletier. An Iterated Min–Max procedure for practical workload balancing on non-identical parallel machines in manufacturing systems. European Journal of Operational Research, 2019, 279 (2), pp.419-428. 10.1016/j.ejor.2019.06.007 . emse-02333444

HAL Id: emse-02333444

https://hal-emse.ccsd.cnrs.fr/emse-02333444

Submitted on 25 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S0377221719304825
Manuscript_b903b930571c4758129422e06eb33c22

An Iterated Min-Max Procedure for Practical Workload Balancing on Non-Identical Parallel Machines in Manufacturing Systems

Quentin Christ[1,2] Stéphane Dauzère-Pérès[1,3] Guillaume Lepelletier[2]

[1]Mines Saint-Etienne, Univ Clermont Auvergne
CNRS, UMR 6158 LIMOS
CMP, Department of Manufacturing Sciences and Logistics
F-13541 Gardanne, France
E-mail: quentin.christ@emse.fr, dauzere-peres@emse.fr

[2]STMicroelectronics Crolles
F-38926 Crolles, France
E-mail: Guillaume.Lepelletier@st.com

[3]Department of Accounting, Auditing and Business Analytics
BI Norwegian Business School
0484 Oslo, Norway

Abstract

This paper presents an original approach for a practical workload balancing problem on non-identical parallel machines in manufacturing systems. After showing the limitations of an initial model, in particular to support relevant decisions, the min-max fairness workload balancing problem is motivated and positioned in the literature. The Iterated Min-Max (IMM) procedure is then presented, with its properties, and illustrated. The IMM consists in solving a succession of linear programs using information from dual variables obtained at each iteration. Computational results on industrial instances show the relevance of the approach when compared to the initial model. The current use of the IMM procedure in an industrial tool is discussed.

Keywords: Manufacturing, Linear Programming, Duality, Min-Max Fairness, Workload Balancing

1. Introduction

An important problem in many systems with multiple resources, such as manufacturing systems that are considered in this paper, is the balancing of the workload, i.e. product quantities to process or tasks to perform, on the various resources. This is in particular critical in capital intensive industries, such as the semiconductor manufacturing industry, where machine usage should be maximized. Moreover, in semiconductor manufacturing systems, machines in the same workshop often have different qualifications (Johnzén et al., 2011; Rowshannahad et al., 2015), i.e. not all products can be processed on all machines or equivalently not all machines are qualified (also called eligible in the literature) to process all products. Moreover, the process time per unit of a given product might differ from one qualified machine to another. In this case, the optimal allocation of product quantities to machines (called workload

© 2019 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

balancing in this paper) for a given criterion is usually not a trivial problem.

The resolution of workload balancing problems have multiple purposes in production and capacity planning (see for instance Mönch et al. (2018)). In particular, optimizing the workload balance helps to define bottleneck (often also called critical) machines, that are usually defined as the machines that are the most loaded. Process improvements should then prioritize focusing on these machines. However, it is also important to characterize the machines that are the less loaded, since they should be made eligible for additional products. In the literature, the focus is often only on minimizing the maximum workload on any machine. As shown in an example in Section 3.2, this may lead to wrong decisions by capacity planners. Because this was observed in an industrial planning tool, an alternative approach has been developed and implemented which is described and discussed in this paper. We are considering workload balancing at tactical level, i.e. continuous product quantities can be assigned to non-identical parallel machines. Also, since hundreds of workload balancing problems are solved by the proposed approach in each run of the industrial planning tool, solutions times are of critical importance. Note that the identical machine case is trivial since a perfect workload balancing (i.e. where all machines have the same workload) can always be found.

This paper is structured as follows. The workload balancing problem in manufacturing systems we are considering is defined and motivated in Section 2. Then, the model initially used in the industrial planning tool to solve the problem, as well as its limits, is presented in Section 3. Section 4 recalls the concept of Min-Max Fairness with the associated literature, and presents its application to our workload balancing problem. Section 5 introduces our Iterated Min-Max (IMM) procedure and, based on the work of Nace and Orlin (2007), shows that it determines solutions with the expected properties. Computational results on industrial instances are discussed in Section 6. Conclusions are drawn, with a short discussion on how the the IMM procedure is used in practice, in Section 7. Future research directions are also provided.

2. Problem Definition

Let us consider a set of products $\mathscr{P} = \{1,\dots,P\}$, and a quantity $q_p \in \mathbb{R}^+$ for each product $p \in \mathscr{P}$, to be processed on a set of non-identical parallel machines $\mathscr{M} = \{1,\dots,M\}$. The quantity of a product can be split on multiple machines. Moreover, the machines on which product p is processed have to be selected in a subset of machines $\mathscr{M}_p \subseteq \mathscr{M}$, with a strictly positive process time $a_{p,m}$ (defined in time units per unit of product) on machine $m \in \mathscr{M}_p$. Each machine m has a capacity c_m (in time units) also strictly positive. Let $X_{p,m}$ be the quantity of product p that is allocated to machine $m \in \mathscr{M}_p$. The workload of machine m is defined as:

$$W_m = \frac{\sum_{p\in\mathscr{P};\ m\in\mathscr{M}_p} a_{p,m}X_{p,m}}{c_m} \tag{1}$$

Note that the workload W_m takes the capacity of machine m into account. Let us define $X = \{X_{p,m};\ \forall(p,m) \in \mathscr{P}\times\mathscr{M}_p\}$ as a workload balancing solution. The goal is therefore to determine the quantity of each product to process on each machine, in order to optimize a certain objective. The

problem (P) can be modeled as follows:

$$\min f(X) \tag{2}$$

$$\sum_{m \in \mathscr{M}_p} X_{p,m} = q_p \quad p = 1, \ldots, \mathscr{P} \tag{3}$$

$$X_{p,m} \in \mathbb{R}^+ \qquad p = 1, \ldots, \mathscr{P},\ m = 1, \ldots, \mathscr{M}_p \tag{4}$$

The objective function $f(.)$ takes a balancing solution as input. Constraints (3) ensure that, for each product p, the whole quantity q_p is allocated to the machines in $\mathscr{M}_p$. Preemption is allowed since variables $X_{p,m}$ are continuous. Note that $f(.)$ depends on the production criteria that are optimized. Also, machine capacities are considered in the workload definition but not as constraints. Hence, the balancing solution may induce a workload for a machine that is larger than its capacity. In this case, the workload is larger than 1 and the machine is considered as overloaded.

Note that it is important for us that the time to solve (P) is very small, since hundreds of problems (see experiments with industrial data in Section 6) are solved for each run of our industrial production planning tool that is used daily in a semiconductor manufacturing facility (with hundreds of products to be processed on hundreds of machines). This is why the function $f(.)$ is usually linear and decision variables $X_{p,m}$ are in $\mathbb{R}^+$.

3. Initial Model and its Limitations

3.1. Initial Model

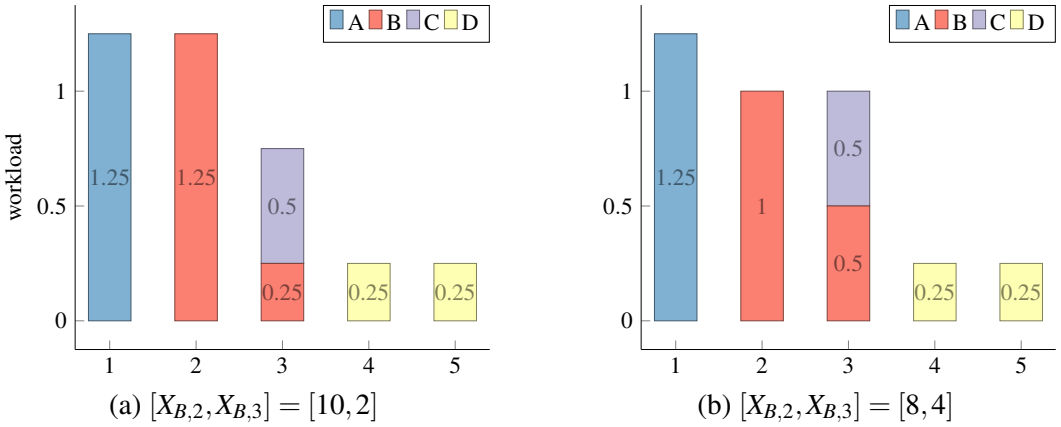
In the model initially implemented in our planning tool, three positive weights (α, β, γ) are used to balance between three criteria in the following objective function $f_c(.)$ which is used in (P):

$$f_c(X) = \alpha \max_{m \in \mathscr{M}} W_m - \beta \min_{m \in \mathscr{M}} W_m + \gamma \sum_{m \in \mathscr{M}_p} c_m W_m$$

$$= \alpha \max_{m \in \mathscr{M}} \frac{\sum_{p \in \mathscr{P}} a_{p,m} X_{p,m}}{c_m} - \beta \min_{m \in \mathscr{M}} \frac{\sum_{p \in \mathscr{P}} a_{p,m} X_{p,m}}{c_m} + \gamma \sum_{p \in \mathscr{P}} \sum_{m \in \mathscr{M}_p} a_{p,m} X_{p,m} \tag{5}$$

This objective function is rather natural in manufacturing systems. The first and primary criterion aims at minimizing the workload of the most loaded machines, i.e. of the bottleneck machines. The second criterion aims at maximizing the workload of the less loaded machines. Combining these two criteria helps to reduce the workload variation between machines, and thus indirectly to better balance the workload among all machines. Finally, the third criterion tries to reduce the total process time among all machines, in order to avoid selecting solutions that increase the process times on machines by selecting slower machines for products. This is particular true in semiconductor manufacturing where, in some workshops, a machine can be faster to process one unit of product 1 than one unit of product 2, while the opposite is true for another machine.

In our industrial case, weights are chosen so that a lexicographical order is satisfied from the first criterion to the third criterion, i.e. $\alpha >> \beta >> \gamma$. Minimizing the workload of bottleneck machines is



(a) $[X_{B,2}, X_{B,3}] = [10, 2]$ (b) $[X_{B,2}, X_{B,3}] = [8, 4]$

Figure 1: Two "equivalent" workload balancing solutions

considered to have the highest priority, followed by maximizing the minimum workload on machines, and finally minimizing the total process time. Another realistic lexicographical order would be to prioritize the total process time over the minimum workload on machines, i.e. $\alpha >> \gamma >> \beta$. This case is also analyzed in the numerical experiments of Section 6.

3.2. Limitations on an illustrative example

Let us illustrate on an example the type of questions we would like to answer, and the limitations of the initial model. It will also be used later to explain the mechanism of our new approach. This instance is composed of 5 machines and 4 products A, B, C and D. To simplify the problem, let us consider identical process times ($a_{p,m} = 1 \; \forall (p,m) \in \mathscr{P} \times \mathscr{M}_p$) and capacities ($c_m = 8 \; \forall m \in \mathscr{M}_p$). Besides, let us set weights $(\alpha, \beta, \gamma) = (1, 1, 0.01)$ to normalize the problem. Quantities for each product are $\{q_A, q_B, q_C, q_D\} = \{10, 12, 4, 4\}$. All machines cannot process all products. Machines 4 and 5 can only process product D, machines 2 and 3 can process products B, C and D, while any product can be processed by machine 1.

Figure 1 presents two possible workload balancing solutions. The two solutions only differ in the allocation of product B on machines 2 and 3. In solution (a), machine 2 takes 10 units of product B for a total process time of 10 hours, and therefore is balanced with machine 1. In contrast, in solution (b), 2 units of product B are moved from machine 2 to machine 3 to balance the workload between machines 2 and 3. The two solutions lead to the same value for the objective function. Indeed, in both cases, the maximum workload is set by machine 1 which, as it is the only one that can process product A, has a workload which is equal to $(a_{A,1} X_{A,1})/c_1 = (1 \times 10)/8 = 1.25$. On the opposite, machines 4 and 5 are only qualified to process product D. Each machine takes 2 units of product D, which leads to the minimal workload, which is equal to $(1 \times 2)/8 = 0.25$. Finally, as process times are identical for all machines, the second term does not depend on the allocated quantities and is equal to $(q_A + q_B + q_C + q_D) = (10 + 12 + 4 + 4) = 30$. Therefore, the objective function of both solutions (a) and (b) is equal to $f_c = \alpha \times 1.25 + \beta \times 0.25 + \gamma \times 30 = 1.80$. However, the two solutions do not provide the same information for the user and solution (b) provides more relevant information than solution (a).

Indeed, based on the allocation of solution (a), machines 1 and 2 appear to have a workload of 1.25, meaning that these machines are critical and should be analyzed. Based on this information, capacity

planners would be tempted to take measures such as, for instance, delaying preventive maintenance operations to provide additional capacity to the temporarily overloaded resource. However, solution (b) shows that only machine 1 is really critical as machine 2 can be balanced with machine 3. Therefore, providing additional capacity to machine 2 would be an unnecessary and costly measure. On the opposite, because the workload of machine 3 in solution (a) is equal to 0.75, capacity planners would conclude that machine 3 does not require any specific focus, and even that a productivity loss (i.e a decrease of the capacity parameter c_m) for machine 3 would not be critical. However, solution (b) underlines the fact that machine 3 is actually important because it can be balanced with machine 2 since both machines are qualified to process product B. Hence, a productivity loss on machine 3 would lead to a workload larger than 1 for both machines 2 and 3, meaning that they would be overloaded and thus not be able to handle the production plan.

This example illustrates the multiple risks of inaccurate forecasting of critical or under-loaded machines. Unnecessary decisions might be taken and the importance of some machines might be underestimated. Furthermore, some workload balancing solutions might not point out the relevant interactions between machines such as solution (a) for machines 2 and 3. These problems were observed in the solutions provided by our industrial production planning tool when the initial objective function (5) was used. When analyzing in detail the results, planners were sometimes complaining that they did not understand the proposed workload on some machines.

To differentiate solutions, adding new terms in the objective function is not necessarily a good alternative, in particular because the use of a linear objective function combining different criteria leads to difficulties in the tuning of weights and loss of clarity. This is why we decided to develop a new and more relevant approach.

4. The Min-Max Fairness Workload Balancing (MMFWB) Problem

In this section, let us introduce the Min-Max Fairness (MMF) problem, well studied, in particular in various areas of networking, and then explain the interest to extend this problem to our workload balancing problem for manufacturing systems by giving several properties on the provided solution.

4.1. The Min-Max Fairness Problem

Generally speaking, min-max fairness is applicable in situations where it is desirable to achieve an equitable distribution of some resources, shared by competing demands (Nace and Pióro, 2008). Intuitively, a min-max fair (respectively max-min fair) solution is a solution where decreasing (respectively increasing) the resources allocated to a demand necessarily leads to an increase (respectively decrease) of the resources allocated to already larger (respectively lower) or equally allocated demands. Note that the min-max (respectively max-min) fairness problem was originally defined as the lexicographic minimax (respectively maximin) problem and some papers use this formulation. Although these terms are only equivalent in case of convex attainable sets (Radunovic and Le Boudec, 2007), as it is the case for our problem, we refer in the remainder of this section both to research on min-max fairness and lexicographic minimax problems. Besides, in the remainder of this paper, we use the acronym MMF to mention both Min-Max and Max-Min Fair problems.

The concept of MMF has been largely studied in a variety of settings, notably for network and communication problems (Bertsekas et al., 1987; Radunovic and Le Boudec, 2007; Nace and Pióro, 2008;

Yaakob and Khalil, 2016; Sadeghi et al., 2018; Zhu et al., 2018). For more information on applications of MMF in network problems, readers are referred to Ogryczak et al. (2014). The concept of MMF has also been applied in other domains such as public services with fair water resource allocation (Wang et al., 2008) or in air transport problems as in Murça (2018) for fair air traffic flow management. More recently, Qi (2016) developed a new performance measure for assignment design problems in the context of outpatient clinics to describe dissatisfaction of both doctors and patients. She then uses a lexicographical minimax approach to improve the design of appointment systems. In a discrete optimization setting, workload balancing has also been considered in the health care literature, see for instance Bredström and Rönnqvist (2008), Lanzarone and Matta (2014) and Yalçındağ et al. (2016). Two recent literature reviews on patient assignment problems in home health care are proposed in Cissé et al. (2017) and Fikar and Hirsch (2017).

Considering applications to manufacturing problems, Luss and Smith (1986) develop a polynomial time algorithm which can be used in production planning to balance the weighted deviation from given product demands. Tang (1988) studies the application of the min-max fair approach to solve Material Requirement Planning (MRP) problems to minimize the penalty cost when demands are not satisfied. He also presents an application of max-min fairness to decide when to produce to maximize the time between two production triggers. King (1989) presents an industrial application of the algorithm of Luss and Smith (1986). He uses the lexicographic minimization procedure to develop a decision support tool to help planners to choose alternative production plans when the initial production plan becomes unfeasible due to variability in the manufacturing facility such as machine breakdowns or changes in customer orders. The original algorithm is extended to a multi-period production problem, and a heuristic is used to determine integer production values. Luss (1999) reviews a variety of resource allocation problems using the lexicographic minimax approach. He notably underlines the interest to use this method in production planning to fairly allocate component to products in high-tech product manufacturing. More recently and at a higher scale, Liu and Papageorgiou (2013) exploit the lexicographic minimax method to solve a multi-objective supply chain optimization problem. Then, Liu and Papageorgiou (2018) also use a MMF approach to fairly balance the profit among actors of a three-echelon supply chain using transfer prices. To the best of our knowledge, there is no reference to applications of min-max fairness that deal with operational capacity planning and workload balancing on machines in manufacturing systems.

Several definitions have been proposed to characterize a min-max fair solution (Nace and Pióro, 2008; Radunovic and Le Boudec, 2007). The latter is used for the definition of a min-max fair solution below. Let us consider a set $\chi \subset \mathbb{R}^N$ $(N \in \mathbb{N})$ and a vector $x \in \chi$. The vector x is said to be *min-max fair* if and only if:

$$\forall\, y \in \chi \quad \exists s \in (1,\ldots,N) \quad y_s < x_s \implies \exists\, t \in (1,\ldots,N) \quad s.t. \quad y_t > x_t \geq x_s \tag{6}$$

This means that decreasing x_s necessarily leads to the increase of another element x_t that is equal or larger.

To connect this definition with our workload balancing problem, let us consider $x \in \mathbb{R}^M$ as a workload balancing solution where each component x_m is the workload on machine m and $\chi \subset \mathbb{R}^M$ is the set of all feasible allocations. Then, x is a min-max fair solution if it is not possible to reduce the workload on a machine without increasing the workload of another machine already more or equally loaded.

Let us define the search of the min-max fair solution for our workload balancing problem as the Min-Max Fair Workload Balancing (MMFWB) problem and define an optimal solution of this problem as a min-max fair workload balancing solution.

Considering again the example in Figure 1, the workload balancing proposed in solution (a) is not a MMFWB solution. Indeed, it is possible to decrease the workload of machine 2 without increasing the workload of machine 1. This re-allocation only increases the workload of machine 3 which is initially less loaded. In contrast, it is not possible to perform such a workload reduction in solution (b). In fact, this solution is min-max fair, which is proved in section 5.

4.2. Properties of MMFWB Solutions

Some properties of MMFWB solutions are presented that can be derived from the structure of min-max fair solutions.

4.2.1. Detection of critical machines

Let us recall the definition used to characterize critical machines.

Definition 4.1. A machine is **critical** if it is impossible to reduce its workload without increasing the workload of another machine with a larger than or equal workload.

Thus, let us state the following proposition.

Proposition 1. *Any machine in a MMFWB solution is critical.*

This property is directly derived from the definition of a min-max fair solution, as it is impossible to reduce one component without increasing another component that is already larger or equal. Thus, in a MMFWB solution, for any machine, it is not possible to reduce its workload without increasing the workload of another machine with an equivalent or larger workload. Because of this property, capacity planners can determine critical machines (and specifically those with the largest workload) more accurately, and thus better plan preventive actions.

4.2.2. Grouping of balanced machines

Let us present a property of machines with equal workload in a MMFWB solution.

Proposition 2. *In a MMFWB solution, if two machines $m1$ and $m2$ are processing at least one common product, i.e. $\exists p$ such that $X_{p,m1} > 0$ and $X_{p,m2} > 0$, then they have the same workload, i.e. $W_{m1} = \frac{\sum_{p\in\mathscr{P};\ m1\in\mathscr{M}_p} a_{p,m1}X_{p,m1}}{c_{m1}} = W_{m2} = \frac{\sum_{p\in\mathscr{P};\ m2\in\mathscr{M}_p} a_{p,m2}X_{p,m2}}{c_{m2}}$.*

The proof can be conducted by contradiction. Consider a MMFWB solution with two machines $m1$ and $m2$ with different workloads, that are processing at least one common product p. Let us assume that machine $m1$ is more loaded than machine $m2$. It is always possible to transfer the workload of $m1$ to the workload of $m2$ by reducing $X_{p,m1}$ and increasing $X_{p,m2}$. Thus, it is possible to reduce the workload of a machine without increasing the workload of a machine with a workload which is larger or equal, which contradicts the basic property of MMFWB solutions.

Because of Proposition 2, in a MMFWB solution, there is no unbalance between machines where a workload can be transferred from one machine to another machine with a smaller workload. In the illustrative example of Section 3.2, solution (a) in Figure 1 is not a MMFWB solution whereas solution (b) is a MMFWB solution. Providing MMFWB solutions gives more confidence to capacity planners.

4.2.3. Existence and Uniqueness

In their book, Bertsekas et al. (1987) propose an important property for the min-max fair vector.

Lemma 1. *If a min-max fair vector exists on a given set, then it is unique.*

Then, in addition to provide a general framework to define Max-Min Fair problems, Radunovic and Le Boudec (2007) also underline a sufficient condition for the existence a min-max fair vector. A simplified version is given below.

Lemma 2. *If the set $\chi \subset \mathbb{R}^N$ is convex and compact, then a min-max fair vector exists on χ.*

Using these two properties, it is possible to write the following proposition.

Proposition 3. *For a given instance of the MMFWB problem, there exists a MMFWB solution and it is unique.*

Let us sketch the proof which is available in the appendix. The key is to prove that, for any instance of the problem, the set of possible allocations χ is always compact and convex. This is based on the fact that the set of feasible solutions ψ of the problem (P) is compact and convex, and by using a linear application ϕ as $\chi = \phi(\psi)$. Because of the linearity properties of ϕ, it is possible to conclude that the set of possible allocations χ is compact and convex, and thus that there always exists a unique optimal solution for the MMFWB problem.

The important result in Proposition 3 guarantees that, for any MMFWB problem, it is possible to find a min-max fair solution that satisfies the properties presented in this section. It also guarantees that the MMFWB solution is unique, in contrast to the initial model in Section 3 for which multiple solutions are optimal.

5. The Iterated Min-Max (IMM) procedure

After motivating the relevance of min-max fair solutions and showing their properties for the workload balancing problem, let us now introduce our method to determine MMFWB solutions. The Iterated Min-Max (IMM) procedure is based on the iterative resolution of a constrained version of the MMFWB problem as a linear program and on the use of the complementary slackness theorem, which is an important property of the Linear Programming theory. Based on Nace and Orlin (2007), we prove that the IMM procedure determines the optimal solution of the MMFWB problem. Then, our illustrative example is used again.

5.1. Description

To introduce the IMM procedure, let us rewrite the MMFWB problem. Let $\mathscr{B} \subseteq \mathscr{M}$ be a subset of machines, and let $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_m\} \in \mathbb{R}^M$ be a given workload vector where γ_m is the workload

assigned to machine m. Then, the linear program $P(M,B,\gamma)$ is written:

$$
\begin{aligned}
min \quad & S && && (7)\\
s.c. \quad & W_m \leq S && \forall m \in \mathscr{M} \setminus \mathscr{B} && (8)\\
& W_m = \gamma_m && \forall m \in \mathscr{B} && (9)\\
& W_m = \sum_{p \in \mathscr{P};\ m \in \mathscr{M}_p} \frac{a_{p,m}}{c_m} X_{p,m} && \forall m \in \mathscr{M} && (10)\\
& \sum_{m \in \mathscr{M}_p} X_{p,m} = q_p && \forall p \in \mathscr{P} && (11)\\
& X_{p,m} \geq 0 && \forall p \in \mathscr{P},\ \forall m \in \mathscr{M}_p && (12)
\end{aligned}
$$

Constraints (11) and (12) are respectively the quantity and the non-negativity constraints. The goal is to minimize the workload of a subset of machines, while the workloads of the other machines are fixed. Variable S is the workload of the most loaded machines in $\mathscr{M} \setminus \mathscr{B}$ and is determined through Constraints (8). Note that S is positive since variables W_m are positive. Constraints (9) impose the workload for machines in $\mathscr{B}$. Constraints (10) define how the workload is allocated to the machines.

At the beginning of the IMM procedure, $\mathscr{B} = \emptyset$ and the maximum workload of all machines is minimized. Then the workload vector γ is iteratively constructed so that, at the end of the procedure, the workload assigned to each machine in γ is the workload in the MMFWB solution. Let us define λ_m as the dual variable associated to Constraint (8) for machine m. Algorithm 1 summarizes the IMM procedure.

Algorithm 1: The IMM Procedure

Data: An instance with M machines and P products
Result: The MMFWB solution γ
$\mathscr{B} := \emptyset$, $\gamma_m = 0\ \forall m \in \mathscr{M}$, $S^* = 0$;
while $\mathscr{B} \neq \mathscr{M}$ **do**
  Solve $P(\mathscr{M},\mathscr{B},\gamma)$ and determine S^* **for** $m \in \mathscr{M} \setminus \mathscr{B}$ **do**
    if $\lambda_m < 0$ **then**
      $\gamma_m := S^*$;
    end
  end
  Set $\mathscr{B} := \mathscr{B} \cup \{m \in \mathscr{M} \setminus \mathscr{B}; \lambda_m > 0\}$;
end

The algorithm takes as input an instance of the workload balancing problem with a set of M machines and P products. Then, the linear program $P(\mathscr{M},\mathscr{B},\gamma)$ is solved to determine the optimal objective function S^*. Next, Constraints (8) that are binding are checked, i.e. corresponding to the machines that limit S^* to its current value. This is done by looking at the dual values λ of Constraints (8), and by using the complementary slackness theorem which states that: For an optimal solution, either the slack variable or the dual variable is equal to 0. Thus, if $\lambda_m < 0$, then Constraint (8) associated to machine m is binding, i.e. it is not possible to reduce the workload of m without degrading S^* by increasing the workload of other machines in $\mathscr{M} \setminus \mathscr{B}$. Once Constraints (8) that are binding are identified, the workload of the

corresponding machines is set to the current optimal objective function S^* (moving from Constraints (8) to Constraints (9)), and the workload vector γ is updated. Hence, the linear program at the next iteration minimizes the maximum workload on the remaining machines while preventing the increase of the fixed workloads of machines in $\mathscr{B}$. The procedure is repeated until the workloads of all machines have been fixed. Note that, when solving $P(\mathscr{M},\mathscr{B},\gamma)$, S^* is such that $S^* < \gamma_m$, $\forall m \in \mathscr{B}$.

Note that the Max-Min Fairness Workload Balancing problem, which can be seen as a dual version of the MMFWB, can be solved by adapting the IMM procedure, which then becomes the Iterated Max-Min procedure. More precisely, in Algorithm 1, the linear program $P(M,B,\gamma)$ is modified by maximizing S in (7) and replacing (8) by $W_m \geq S$.

5.2. Proof of correctness

Various papers in the literature on MMF problems (or their lexicographic equivalent) have proposed solution methods. Luss and Smith (1986) and Tang (1988) propose polynomial time algorithms to solve special production planning problems. Other authors use the resolution of linear programs iteratively. Bertsekas et al. (1987) also propose an algorithm to solve the max-min fair problem. Radunovic and Le Boudec (2007) show that a special case of the max-min fairness problem can be solved very fast by a Water Filling algorithm, and propose a procedure using linear programs iteratively to solve the general problem. Moreover, Behringer (1981) details the use of a simplex based method to solve the lexicographically extended maximin problem. In these two last papers, there is no mention of the use of dual variables. However, some other research works explicitly consider duality. In his book, Luss (2012) considers dual variables to detect saturated constraints, notably in the case of non separable objective functions. Nace and Pióro (2008) develop a linear programming procedure to solve max-min fair routing in communication networks and also mention the use of the min-max fair concept to lexicographically balance the load in a given network. Finally, Nace and Orlin (2007) introduce what they call lexicographically minimum load linear programming problems for applications in capacitated multicommodity networks. They present a linear programming based procedure and give a proof of its correctness. Thus, although we could not find an algorithm to solve our MMFWB problem, the analogy between lexicographic minimization and min-max fair problems is strong. This is why we rely on Nace and Orlin (2007) to state the proposition below.

Proposition 4. *The solution provided by the IMM procedure for the MMFWB problem is optimal and is obtained in polynomial time by solving at most $|\mathscr{M}|$ linear programs, where $\mathscr{M}$ is the set of machines.*

The proof of Proposition 4 follows the proof of correctness in Nace and Orlin (2007), with nevertheless a difference for the polynomial time analysis. First, let us underline the analogy between the approach in Nace and Orlin (2007) and the IMM procedure. First, the linear program (P_1) in Nace and Orlin (2007) can be transformed into problem $P(\mathscr{M},\mathscr{B},\gamma)$, by considering that Constraints (1) in (P_1) are Constraints (8) in $P(\mathscr{M},\mathscr{B},\gamma)$ and Constraints (2) in (P_1) are Constraints (9) and (11) in $P(\mathscr{M},\mathscr{B},\gamma)$. Besides, Step 1 in the algorithm of Nace and Orlin (2007), in which a linear program is solved, corresponds to solving $P(\mathscr{M},\mathscr{B},\gamma)$ in Algorithm 1. Then Step 2 in the algorithm of Nace and Orlin (2007), which aims at finding binding constraints and updating the new linear program, corresponds to the remaining steps in Algorithm 1. Finally, the two algorithms end when no inequality constraint remains, and the resulting load vector γ is min-max fair (leximax minimal in Nace and Orlin (2007)).

Nace and Orlin (2007) also show that their algorithm is polynomial by relying on two main points: (1) The linear problem can be solved in polynomial time and (2) At most $2|\mathscr{M}|-1$ linear problems must be solved. The first point brings no difficulty as there are numerous methods to solve linear programs in polynomial time (Cook et al. (1995)).

For the second point, let us go a little further than Nace and Orlin (2007), who guarantee that the linear program is solved at most $|\mathscr{M}|$ times when using a solution method that provides a strictly complementary solution. They cite for example the central trajectory based interior point method of Freund and Mizuno (2000). To overcome the use of a method that does not guarantee to provide strictly complementary solutions, Nace and Orlin (2007) propose an additional step, leading to the resolution of at most $2|\mathscr{M}|-1$ linear programs. However, we claim that, using the IMM, at most $|\mathscr{M}|$ linear programs are solved, regardless of the solution method. To prove it, it is necessary to show that, at each iteration, at least one dual value is strictly negative. This hypothesis is not obvious with a complementary optimal solution. as the complementary slackness states that, for a given constraint, "at least" the slack variable or the dual variable associated is equal to 0. Thus, although a given constraint is binding, the associated dual variable may be equal to 0, and if all dual variables are equal to zero, then the algorithm may cycle without changing $\mathscr{B}$. However, the unrestricted primal variable S in Constraints (8) leads to the associated constraint in the dual problem $\sum_{m\in\mathscr{M}\setminus\mathscr{B}}\lambda_m=-1$. For more detail, a similar reasoning can be found in Luss (2012) (Chapter 3, Page 115). Therefore, since $\lambda_m\leq 0$, $\forall m\in\mathscr{M}\setminus\mathscr{B}$, the dual constraint implies that $\lambda_m<0$ for at least one machine m at each iteration. Consequently, there are at most M linear programs to solve.

5.3. An illustrative example

To conclude this section, let us illustrate the IMM procedure using the example in Section 3.2. Let us recall that the instance includes 5 machines, each with a capacity $c_m=8$, and 4 products A, B, C and D. Moreover, to simplify, all process times are assumed to be identical. Quantities to process for each product are $\{q_A,q_B,q_C,q_D\}=\{10,12,4,4\}$, and all machines cannot process all products. Machines 4 and 5 can only process product D, machines 2 and 3 can only process products B, C and D, while all products can be processed by machine 1.

- **Initialization**: $\mathscr{B}:=\emptyset$ and $\gamma=\{0,\ldots,0\}$. Thus, let us minimize the maximum workload of all machines.
- **Step 1.1** Solve the linear program $P(\mathscr{M},\emptyset,\{0,\ldots,0\})$. The optimal objective function value $S^*=10/8=1.25$ with solution $[q_{A,1},q_{B,2},q_{B,3},q_{C,3},q_{D,4},q_{D,5}]=[10,10,2,4,4,0]$. Therefore, machines 1 and/or 2 seem to be critical.
- **Step 1.2** Analyze the dual values associated to Constraints (8). $[\lambda_1,\ldots,\lambda_5]=[-12.5,0,0,0,0]$. Therefore, although workload constraints associated to machines 1 and 2 seem to be blocking, the analysis of the associated dual values shows that only machine 1 is actually critical.
- **Step 1.3** Set $\gamma_1=S^*=1.25$ and $\mathscr{B}:=\{1\}$. The workload of machine 1 is set to 1.25, and machine 1 is no longer considered in the maximum workload minimization.

- **Step 2.1** Solve the new linear program $P(\mathscr{M}, \{1\}, \{1.25, 0, 0, 0, 0\})$. The optimal objective function is $S^* = 8/8 = 1$ with solution $[q_{A,1}, q_{B,2}, q_{B,3}, q_{C,3}, q_{D,4}, q_{D,5}] = [10, 8, 4, 4, 4, 0]$. Compared to the previous solution, some quantity of product B was transferred from machine 2 to machine 3, allowing these two machines to balance each other with a common workload of 8/8=1.

- **Step 2.2** Analyze the dual values associated to Constraints (8). $[\lambda_2, \ldots, \lambda_5] = [-6.25, -6.25, 0, 0]$. Machines 2 and 3 have a strictly negative dual value associated to their workload constraint, which means that they both are blocking.

- **Step 2.3** Set $\gamma_1 = \gamma_2 = S^* = 1$ and $\mathscr{B} := \{1, 2, 3\}$. The workload of machines 2 and 3 is set to 1, and machines 2 and 3 are no longer considered in the maximum workload minimization.

- **Step 3.1** Solve $P(\mathscr{M}, \{1, 2, 3\}, \{1.25, 1, 1, 0, 0\})$. The optimal objective function is $S^* = 2/8 = 0.25$ with solution $[q_{A,1}, q_{B,2}, q_{B,3}, q_{C,3}, q_{D,4}, q_{D,5}] = [10, 8, 4, 4, 2, 2]$. Machines 4 and 5 can only process product D, and thus share q_D to balance each other, leading to a small workload of 0.25.

- **Step 2.2** Analyze the dual values associated to Constraints (8). $[\lambda_4, \lambda_5] = [-6.25, -6.25]$. Machines 4 and 5 have a strictly negative dual value associated to their workload constraint, which means that both machines are blocking.

- **Step 2.3** Set $\gamma_4 = \gamma 5 = S^* = 0.25$ and $\mathscr{B} := \{1, 2, 3, 4, 5\}$ The workload of machines 4 and 5 is set to 0.25, and machines 4 and 5 are no longer considered in the maximum workload minimization.

- **End**, as $\mathscr{B} = \mathscr{M}$. Return $\gamma = [1.25, 1, 1, 0.25, 0.25]$ with the corresponding quantity allocation $[q_{A,1}, q_{B,2}, q_{B,3}, q_{C,3}, q_{D,4}, q_{D,5}] = [10, 8, 4, 4, 2, 2]$. All machines have a fixed workload.

6. Computational Experiments

The previous sections showed that the characteristics of the min-max fair solutions lead to useful properties for our problem, and that the IMM procedure provides min-max fair solutions, and thus workload balancing solutions with the desired properties. In this section, our goal is to experimentally evaluate the performance of the IMM procedure compared to the initial balancing model using (i) Equation (5) with $\alpha >> \gamma >> \beta$ in $f_c(.)$ and (ii) Equation (5) with $\alpha >> \beta >> \gamma$ in $f_c(.)$. The three approaches are, respectively, referred as *AvgFirst* (minimizing total process time prioritized over maximizing minimum workload), *MinFirst* (maximizing minimum machine workload prioritized over minimizing total process time) and IMM. We use a set of 30 real industrial instances taken from the most advanced manufacturing facility of STMicrolectronics, where the number of machines is about 350 and the number of products ranges from 4,000 to 8,000.

The performance indicators to compare the initial balancing model and the IMM procedure are discussed in Section 6.1. The actual comparison is conducted in Section 6.2.

6.1. Performance Indicators

The indicator *Nb of Unnecessary Loaded Machines* aims at quantifying the ratio of machines for which the initial model gives a larger workload than the one determined by the IMM procedure. Given the properties of min-max fair solutions, we know that, if a machine has a larger workload than that of the

MMF solution, then it should be possible to decrease this workload by only increasing the workload of the less loaded machines, which is preferable. This indicator helps to evaluate the ratio of unnecessarily loaded machines prevented by the IMM procedure. Using the example in Figure 1, the solution provided by the initial model and presented in (a) shows machine 2 as being unnecessarily overloaded. Indeed, machine 2 has a larger workload than in (b). However, according to the properties of the IMM procedure, we know that it is possible to reduce the workload of machine 2 (from solution (a) to solution (b)), by only increasing the workload of less loaded machines, i.e. machine 3 in our case. In this example, we have one machine out of five that are unnecessarily overloaded, which leads to a ratio of 20%. Note that machine 3 has a lower workload when using the initial method (solution (a)) than when using the IMM procedure (solution (b)). However, this does not mean that solution (a) is better, because we know that, by property of MMFWB solutions (provided by the IMM procedure), it would be possible to increase the workload of machine 3 to reduce the workload of an already more loaded machine (here machine 2), which is always preferable.

Then, as shown in previous sections, the IMM procedure determines independent sets of machines and products. The indicator *Nb of Unbalanced Machines* aims at analyzing balanced machines in the IMM solution, and then evaluates the ratio of these machines whose balance was broken. These cases of broken balance are non desired as they do not provide information on the possible "mutual supply" relationship between machines. Again, by illustrating with the example in Figure 1, note that the workloads of machines 2 and 3 are not the same in solution (a) and in solution (b). But we do know, by property of MMFWB solutions, that if a method provides a solution with a different workload for a machine, then there are two possible cases: (1) The machine has a larger workload than in the MMFWB solution, i.e it is possible to transfer some of its workload to a less loaded machine (machine 2 in the example), or (2) The machine has a lower workload than in the MMFWB solution, i.e. it is possible to add some of the workload of a more loaded machine (machine 3 in the example). In both cases, these machines are not well balanced. In the example, there are two unbalanced machines, i.e. a ratio of 40%.

The indicator *Average Machine Workload* is used to evaluate the impact of the IMM procedure on the total process time on the machines. Indeed, we observed that, in the initial balancing model, the second term in the objective function (5) aims at minimizing the total process time. The goal is to avoid that smoothing the workload between machines (minimizing the maximum workload and maximizing the minimum workload) impacts too negatively the total process time. Because it does not explicitly consider the total process time, it is interesting to analyze the impact of the IMM procedure on this indicator, expressed as the average machine workload.

Finally, since the IMM procedure generally solves several linear programs against only one with the initial balancing model, it it also interesting to evaluate the impact on computational times. The column *Diff.* shows the difference in percentage between the time required using the IMM procedure and the fastest method between *AvgFirst* and *MinFirst*, i.e $Diff. = \frac{CPU_{IMM} - Min(CPU_{AvgFirst}, CPU_{MinFirst})}{Min(CPU_{AvgFirst}, CPU_{MinFirst})}$.

6.2. Comparison with Initial Balancing Models

Each instance was run with our planning tool, and the results are summarized in Tables 1 and 2. The number of MMFWB problems solved for each instance can be found in the second column of Table 2. Note that 365 MMFWB problems were solved on average for the 30 instances. For each MMFWB problem, *AvgFirst*, *MinFirst* and the IMM procedure are run and compared using the indicators introduced

in the previous section. Each balancing problem includes several non-identical parallel machines (8 on average but varying between 1 and 20 machines depending on the problem) and several dozen different products.

Table 1: Proportion of non desirable balancing cases with initial model and IMM procedure

Instance	Nb of Unnecessary Loaded Machines			Nb of Unbalanced Machines		
	AvgFirst	*MinFirst*	IMM	*AvgFirst*	*MinFirst*	IMM
1	12.1%	14.9%	0.0%	16.7%	8.8%	0.0%
2	12.5%	17.3%	0.0%	16.8%	8.6%	0.0%
3	12.8%	15.6%	0.0%	17.5%	7.8%	0.0%
4	11.3%	14.6%	0.0%	15.1%	7.4%	0.0%
5	9.9%	13.1%	0.0%	12.1%	6.2%	0.0%
6	11.2%	13.3%	0.0%	10.8%	5.0%	0.0%
7	10.5%	13.4%	0.0%	14.6%	7.1%	0.0%
8	13.4%	16.1%	0.0%	18.1%	10.6%	0.0%
9	12.5%	16.0%	0.0%	18.0%	8.4%	0.0%
10	13.9%	17.2%	0.0%	14.9%	9.7%	0.0%
11	13.9%	19.6%	0.0%	16.0%	11.5%	0.0%
12	13.2%	19.5%	0.0%	16.4%	11.6%	0.0%
13	12.8%	14.8%	0.0%	14.1%	8.9%	0.0%
14	16.0%	17.9%	0.0%	16.4%	12.1%	0.0%
15	17.5%	18.7%	0.0%	16.5%	12.7%	0.0%
16	16.4%	18.1%	0.0%	15.1%	11.6%	0.0%
17	16.7%	18.5%	0.0%	16.0%	12.0%	0.0%
18	17.2%	19.8%	0.0%	16.8%	11.1%	0.0%
19	17.0%	18.0%	0.0%	14.2%	9.2%	0.0%
20	17.4%	18.1%	0.0%	16.4%	11.3%	0.0%
21	16.8%	19.6%	0.0%	15.3%	8.6%	0.0%
22	17.6%	18.7%	0.0%	14.6%	10.2%	0.0%
23	17.2%	18.1%	0.0%	16.4%	11.5%	0.0%
24	17.9%	19.6%	0.0%	18.1%	12.8%	0.0%
25	17.6%	21.3%	0.0%	17.0%	10.5%	0.0%
26	17.5%	20.0%	0.0%	17.0%	11.0%	0.0%
27	19.5%	21.1%	0.0%	17.4%	11.3%	0.0%
28	17.5%	18.2%	0.0%	17.6%	11.7%	0.0%
29	18.1%	21.4%	0.0%	17.8%	11.4%	0.0%
30	18.7%	20.3%	0.0%	16.8%	9.8%	0.0%
Avg	15.2%	17.8%	0.0%	16.0%	10.0%	0.0%
Max	19.5%	21.4%	0.0%	18.1%	12.8%	0.0%
Min	9.9%	13.1%	0.0%	10.8%	5.0%	0.0%

Several remarks can be made when analyzing Table 1. First, as expected, the columns corresponding to the IMM procedure only have zeroes. Then, note that there are differences between the solutions provided with the initial model depending on the criterion that is prioritized. The *Nb of Unnecessary*

Loaded Machines is equal to 15.2% on average with *AvgFirst* while it increases to 17.8% with *MinFirst*. This shows that a significant number of machines (55 on average, at least 36 in each instance) has a workload that could be reduced without impacting the workload of more loaded machines.

Again, an important remark must be made on machines for which the IMM procedure appears to increase the workload compared to the initial balancing model. Indeed, there are some machines for which the average machine workload determined by the IMM procedure is larger than the average machine workload determined by the initial model. This type of situations is possible but in theory implies that in return the IMM procedure is able to reduce the workload of other machines that are already more loaded. Our industrial computational experiments show that, every time a machine is more loaded in the IMM solution than in the initial model, then the opposite phenomenon is observed for an already more loaded machine.

The IMM procedure can thus significantly impact the analysis of the plant capacity, because planners may have an incorrect view of the workload of dozens of machines. This may lead to inappropriate decisions.

Let us now consider the second indicator, the *Nb of Unbalanced Machines*, which corresponds to the ratio of machines that are balanced with each other in a Min-Max Fair solution but are not in the solutions provided by the initial model. The results show that on average 16% of machines with *AvgFirst* and 10% of machines with *MinFirst* are not balanced as they should. This type of unbalance is similar to the one presented in our illustrative example, where machines 2 and 3 are not balanced in solution (a) of Figure 1 although they could be. These results show that the use of the IMM procedure to replace the initial balancing model allows the interdependence relationships to be highlighted for a significant number of machines.

The results on the average machine workload summarized in Table 2 show that *AvgFirst*, as expected and because it prioritizes the total process time over the minimum workload, provides solutions with a lower average machine workload. However, note that the difference remains relatively small with *MinFirst* and the IMM procedure. In addition, note that some of the solutions determined by the IMM procedure are better than the ones of the initial model. Hence, the benefits of the IMM procedure do not come at the expense of the average machine workload.

Furthermore, despite the large size of the instances and because only continuous variables are used, the three approaches are running very quickly, with CPU times generally of the order of a few seconds to solve hundreds of workload balancing problems. A slight increase of the computational time of 3.9% on average is observed with the IMM procedure. Hence, because it is viable and efficient, the IMM procedure is used to solve industrial problems of very large sizes.

7. Conclusions and Perspectives

We addressed the problem of optimally balancing the workload of different products on non-identical parallel machines in manufacturing systems, which occurs in semiconductor manufacturing in particular. We then recalled the notion of Min-Max Fair solution and showed that, applied to our problem, it can provide comprehensible and meaningful solutions for planners, especially for machine capacity management. The Iterated Min-Max (IMM) procedure is proposed and, based on the work of Nace and Orlin (2007), is proved to determine optimal solutions for our Min-Max Fair Workload Balancing Problem.

Table 2: Average machine workload and total computational time with initial model and IMM procedure

Instance	Nb of MMFWB Problems	Average Machine Workload			Total CPU Time (ms)			
		AvgFirst	*MinFirst*	IMM	*AvgFirst*	*MinFirst*	IMM	*Diff.*
1	386	0.391	0.398	0.397	130	120	130	8%
2	384	0.418	0.419	0.418	130	130	130	0%
3	384	0.318	0.320	0.320	120	130	120	0%
4	380	0.302	0.303	0.303	130	120	130	8%
5	379	0.346	0.344	0.344	130	130	120	-8%
6	379	0.207	0.206	0.206	150	110	120	9%
7	380	0.306	0.310	0.309	120	120	140	17%
8	378	0.314	0.321	0.321	120	110	110	0%
9	376	0.332	0.340	0.339	120	120	110	-8%
10	375	0.329	0.333	0.330	120	120	120	0%
11	372	0.262	0.267	0.265	120	110	110	0%
12	375	0.263	0.266	0.264	130	110	120	9%
13	376	0.300	0.303	0.302	120	120	120	0%
14	368	0.231	0.234	0.237	370	370	380	3%
15	335	0.251	0.254	0.258	380	390	390	3%
16	349	0.239	0.243	0.247	370	370	380	3%
17	369	0.308	0.312	0.317	380	380	420	11%
18	365	0.246	0.247	0.251	370	370	380	3%
19	382	0.250	0.250	0.251	380	400	450	18%
20	368	0.278	0.283	0.286	370	370	370	0%
21	339	0.215	0.217	0.219	120	110	120	9%
22	345	0.252	0.253	0.255	370	400	390	5%
23	365	0.264	0.263	0.265	430	450	420	-2%
24	378	0.298	0.299	0.301	420	510	430	2%
25	344	0.249	0.253	0.256	430	410	430	5%
26	345	0.278	0.282	0.285	400	450	430	8%
27	348	0.296	0.300	0.303	430	470	450	5%
28	345	0.329	0.336	0.340	430	460	440	2%
29	354	0.324	0.332	0.335	440	440	450	2%
30	339	0.246	0.248	0.250	420	430	440	5%
Avg	365	0.288	0.291	0.293	271.6	277.6	278.3	3.9%
Min	335	0.207	0.206	0.206	120	110	110	-8.3%
Max	384	0.418	0.419	0.418	440	510	450	18.4%

The procedure has been implemented in a Decision Support System for operational production planning in an advanced wafer semiconductor manufacturing facility. The IMM procedure is ran hundreds of times for each calculation of the production plan. Relevant information on the planned workload of each machine is provided to the users. Moreover, the IMM procedure determines groups of "connected" machines and products that are used in other applications.

Our future research is focusing on solving the workload balancing problem on two periods, by allowing some product quantities to be assigned to either one of the two periods. Also, for some applications, although a unique solution is determined by the IMM procedure for the workload per machine, there are usually many ways to balance the quantity of a given product between machines. Hence, for operational decisions, we are working on algorithms to propose various product dispatching from a single workload balancing solution. This might require to study how to consider other criteria in the IMM procedure.

Acknowledgements

This work has been partially financed by the ANRT (Association Nationale de la Recherche et de la Technologie) through the PhD number 2016/0421 with CIFRE funds and a cooperation contract between STMicroelectronics and Mines Saint-Etienne.

References

Behringer, F. A., 1981. A simplex based algorithm for the lexicographically extended linear maxmin problem. European Journal of Operational Research 7 (3), 274–283.

Bertsekas, D. P., Gallager, R. G., Humblet, P., 1987. Data networks. Vol. 2. Prentice-hall Englewood Cliffs, NJ.

Bredström, D., Rönnqvist, M., 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. European journal of operational research 191 (1), 19–31.

Cissé, M., Yalçındağ, S., Kergosien, Y., Şahin, E., Lenté, C., Matta, A., 2017. OR problems related to home health care: A review of relevant routing and scheduling problems. Operations Research for Health Care 13, 1–22.

Cook, W., Lovász, L., Seymour, P. D., et al., 1995. Combinatorial optimization: papers from the DIMACS Special Year. Vol. 20. American Mathematical Soc.

Fikar, C., Hirsch, P., 2017. Home health care routing and scheduling: A review. Computers & Operations Research 77, 86–95.

Freund, R. M., Mizuno, S., 2000. Interior point methods: current status and future directions. In: High performance optimization. Springer, pp. 441–466.

Johnzén, C., Dauzère-Pérès, S., Vialletelle, P., 2011. Flexibility measures for qualification management in wafer fabs. Production Planning and Control 22 (1), 81–90.

King, J. H., 1989. Allocation of scarce resources in manufacturing facilities. Bell Labs Technical Journal 68 (3), 103–113.

Lanzarone, E., Matta, A., 2014. Robust nurse-to-patient assignment in home care services to minimize overtimes under continuity of care. Operations Research for Health Care 3 (2), 48–58.

Liu, S., Papageorgiou, L. G., 2013. Multiobjective optimisation of production, distribution and capacity planning of global supply chains in the process industry. Omega 41 (2), 369–382.

Liu, S., Papageorgiou, L. G., 2018. Fair profit distribution in multi-echelon supply chains via transfer prices. Omega 80, 77–94.

Luss, H., 1999. On equitable resource allocation problems: A lexicographic minimax approach. Operations Research 47 (3), 361–378.

Luss, H., 2012. Equitable Resource Allocation: Models, Algorithms and Applications. Vol. 101. John Wiley & Sons.

Luss, H., Smith, D. R., 1986. Resource allocation among competing activities: A lexicographic minimax approach. Operations Research Letters 5 (5), 227–231.

Mönch, L., Uzsoy, R., Fowler, J. W., 2018. A survey of semiconductor supply chain models part iii: master planning, production planning, and demand fulfilment. International Journal of Production Research 56 (13), 4565–4584.

Murça, M. C. R., 2018. Collaborative air traffic flow management: Incorporating airline preferences in rerouting decisions. Journal of Air Transport Management 71, 97–107.

Nace, D., Orlin, J. B., 2007. Lexicographically minimum and maximum load linear programming problems. Operations research 55 (1), 182–187.

Nace, D., Pióro, M., 2008. Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial. IEEE Communications Surveys & Tutorials 10 (4).

Ogryczak, W., Luss, H., Pióro, M., Nace, D., Tomaszewski, A., 2014. Fair optimization and networks: A survey. Journal of Applied Mathematics 2014.

Qi, J., 2016. Mitigating delays and unfairness in appointment systems. Management Science 63 (2), 566–583.

Radunovic, B., Le Boudec, J.-Y., 2007. A unified framework for max-min and min-max fairness with applications. IEEE/ACM Transactions on networking 15 (5), 1073–1083.

Rowshannahad, M., Dauzere-Peres, S., Cassini, B., 2015. Capacitated qualification management in semiconductor manufacturing. Omega 54, 50–59.

Sadeghi, M., Björnson, E., Larsson, E. G., Yuen, C., Marzetta, T. L., 2018. Max–min fair transmit precoding for multi-group multicasting in massive mimo. IEEE Transactions on Wireless Communications 17 (2), 1358–1373.

Tang, C. S., 1988. A max-min allocation problem: its solutions and applications. Operations Research 36 (2), 359–367.

Wang, L., Fang, L., Hipel, K. W., 2008. Basin-wide cooperative water resources allocation. European Journal of Operational Research 190 (3), 798–817.

Yaakob, N., Khalil, I., 2016. A novel congestion avoidance technique for simultaneous real-time medical data transmission. IEEE journal of biomedical and health informatics 20 (2), 669–681.

Yalçındağ, S., Matta, A., Şahin, E., Shanthikumar, J. G., 2016. The patient assignment problem in home health care: using a data-driven method to estimate the travel times of care givers. Flexible Services and Manufacturing Journal 28 (1-2), 304–335.

Zhu, X., Jiang, C., Yin, L., Kuang, L., Ge, N., Lu, J., 2018. Cooperative multigroup multicast transmission in integrated terrestrial-satellite networks. IEEE Journal on Selected Areas in Communications.

AppendixA. Proof of the existence and uniqueness of min-max fair workload balancing solution

Let us recall the property based on Bertsekas et al. (1987) and Radunovic and Le Boudec (2007).

Proposition 5. *Since any subset of $\mathbb{R}^N$ is convex and compact, then there exists a min-max fair vector and it is unique.*

The idea is to prove that, for any instance of the MMFWB problem, the set of possible allocations $\chi \subset \mathbb{R}^M$, for which the vector γ corresponds to the workloads of the machines for a given feasible workload balancing solution, is always compact and convex. However, this property is not obvious in our case, and we first have to define the set of feasible workload balancing solutions ψ of Problem (P).

A solution is defined by a vector:

$$X \in \mathbb{R}^{+(P\times M)} = \{\ldots, X_{p,m}, \ldots\}$$

which summarizes the product quantities assigned to the machines. Let us then define the set of feasible solutions $\psi \subset R^{+(P\times M)}$, i.e. the set of solutions that satisfy the product quantity balance and qualification constraints:

$$\begin{aligned}\psi = \{\ & X \in \mathbb{R}^{+(P\times M)} \\ & \text{s.t.} \quad \forall p \in \mathscr{P}, \sum_{m\in\mathscr{M}_p} X_{p,m} = q_p \quad \wedge \quad \forall p \in \mathscr{P}, \forall m \notin \mathscr{M}_p, X_{p,m} = 0\}\end{aligned} \tag{A.1}$$

As it does not use strict inequality relations and because the intersection of closed sets is closed, it is possible to state that the set ψ is closed. Since $\inf(\psi) = 0$ and $\sup(\psi) = \max_{p\in\mathscr{P}} q_p$, the set ψ is also bounded. Based on the Borel-Lebesgue theorem, in a $\mathbb{R}^N$ topology, all closed and bounded sets are also compact. Therefore ψ is compact. Besides, ψ is convex. Indeed, if $(x,y) \in \psi^2$ then, for any vector $z = \lambda x + (1-\lambda)y$ with $\lambda \in [0,1]$, is also in ψ.

Let us then define the application ϕ, that takes as input a vector of product quantities and gives the load vector γ that includes the workloads of all machines:

$$\phi : \quad \psi \subset R^{+(P\times M)} \quad \rightarrow \quad \phi(\psi) \subset \mathbb{R}^{+M}$$

$$X = \{\ldots, X_{p,m}, \ldots\} \quad \mapsto \quad \gamma = \{\ldots, \sum_{p\in\mathscr{P};\, m\in\mathscr{M}_p} \frac{a_{p,m}}{c_m} X_{p,m}, \ldots\}$$

The application ϕ is linear since, $\forall(\lambda,\mu) \in \mathbb{R}^2$ and $\forall(x,y) \in \chi$, $\phi(\lambda x + \mu y) = \lambda\phi(x) + \mu\phi(y)$. The set χ is defined as follows: $\chi = \phi(\psi)$. Since ψ is convex and ϕ is linear, χ is also convex. Also, $\mathbb{R}^{+(P\times M)}$ and $\mathbb{R}^{+M}$ are finite dimension spaces, and thus the linear application ϕ preserves compacity. Since ψ is compact, then χ is also compact.

It is thus possible to conclude that χ is a convex and compact set. Based on Proposition 5, it is possible to assert that a min-max fair solution exists in the set, and that this solution is unique.