Les cahiers du laboratoire Leibniz

Comparison of objective functions for the JIT problemf

Vassilissa Lebacque, Vincent Jost, Nadia Brauner

Laboratoire Leibniz-IMAG, 46 av. Félix Viallet, 38000 GRENOBLE, France -
ISSN : 1298-020X

n° 110
Sep. 2004

Site internet : http://www-leibniz.imag.fr/LesCahiers/

Comparison of objective functions for the JIT problem

Vassilissa Lebacque, Vincent Jost, Nadia Brauner

15 septembre 2004

Abstract

Just-in-time production models have been developed over recent years in order to reduce costs of diversified small-lot production. Those methods aim at matching the exact demand of each product and therefore holding inventory and shortage costs as small as possible. Different ways of measuring the slack between a given schedule and the ideal no-inventory no-shortage production have been considered in the literature. This note compares the three most studied objective functions and refutes several conjectures that have been developed in the last few years.

1 Introduction

Just-in-time (JIT) manufacturing environments have been developed in order to reduce costs of diversified small-lot production. This model aims at matching the exact demand of each product and therefore holding inventory and shortage costs as small as possible.
Monden [10] states that the most important goal of a JIT system is to keep the schedule as balanced as possible, i.e. to keep the production rate of each type of product per unit of time as smooth as possible. Different ways of measuring the deviation of the real production from the ideal perfectly balanced schedule have been considered in the literature. The main objective of this document is to compare the three most studied objective functions and to refute several conjectures that have been developed in the last few years. It also describes tools and models that characterise instances with solutions which optimise several objective functions simultaneously.

We use two different formulations of the JIT scheduling problem. The first one, a linear program, has been introduced by Miltenburg in [9]. The second one is a reformulation by Kubiak and Sethi [7] as an assignment problem.

Section 2 reviews previous research on JIT manufacturing. Section 3 gives the constraints of the problem and the notations used. It also provides a proof of the equivalence between the three objective functions for the two-part type case. In Section 4, description of the maximum deviation JIT problem as a perfect matching problem [11] is presented and the main results of reformulation [7] are given ; utilisation of those results to prove optimality of min-sum sequences is explained and graph representation in this document is described and justified. Section 5 contains a comparison of the total and maximum deviation problems when deviations are absolute values. Section 6 deals with 1-bounded total deviation problems, i.e. total deviation problems such that the maximum deviation is lower or equal to 1. In Section 7 we give some results on maximum deviation problems and total deviation problems with square functions as deviations. Section 8 is a remark on total deviation problems. In Section 9, we describe the linear programs employed in the tests.

2 Literature review

In this section, we review previous research on JIT manufacturing. The description by Monden [10] of Toyota's production system arouses first interest in levelled scheduling. By proposing an optimisation formulation of the problem, Miltenburg [9] has lead to a considerable amount of research. This formulation aims at minimising the *total deviation* or sum of all deviations of the real production from the ideal but rational production. When the deviations are convex, non-negative functions, Kubiak and Sethi [6, 7] prove that this optimisation model can be reformulated as an assignment problem. In this formulation, deviations from the ideal are represented by penalties for placing parts earlier or tardier than their location in the ideal sequence. Inman and Bulfin [3] give a pseudo-polynomial heuristic for total deviation problems with slightly different penalty functions. They consider that the due date of a part is its ideal instant of production and solve the problem with an Earliest Due Date Rule.

An other class of objective functions has been widely considered in the literature. The constraints remain the same and the goal is to minimise the *maximum deviation* of the real production from the ideal. Steiner and Yeomans [11] show that when considered as a one-machine scheduling problem with release and due dates, the model could be reduced to a perfect matching problem in a bipartite convex graph. From this approach, they obtain a pseudo-polynomial time algorithm. Brauner and Crama [2] revise those results and show that the maximum deviation JIT problem is in Co-NP. Note that it is even difficult to know whether or not total deviation or maximum deviation problems are in NP as the output is not polynomial in the size of the input.

3 Notations and constraints of the problem

3.1 Constraints

A diversified small-lot production consists of n part types with a demand $d_i \in \mathbb{N}$ for part type $i = 1, 2, ..n$. Each part is produced in one time period. Let $D = \sum_{i=1}^{n} d_i$ be the total demand. A schedule will be called *uniformly levelled* if, at each time period k, the line has assembled $k\, d_i/D$ parts of type i. The proportion $r_i = d_i/D$ is called the *ideal production rate* and a JIT schedule tries to keep the effective production rate as close as possible to that ideal. Monden [10] states that it is the main goal of Toyota's JIT systems.
In order to formulate this problem as an optimisation problem, we denote $x_{i,k}$, for $i = 1, 2, .. n$; $k = 1, 2, .. D$ the number of parts of type i produced in the time periods 1 to k.

Miltenburg [9] formulated the constraints of the problem as follows :

$$\begin{array}{lll} \sum_{i=1}^{n} x_{i,k} = k, & k = 1, 2, .. D & (1.a) \\ x_{i,D} = d_i, & i = 1, 2, .. n & (1.b) \\ 0 \leq x_{i,k} - x_{i,k-1}, & i = 1, 2, .. n;\ k = 2, 3, .. D & (1.c) \\ x_{i,k} \in \mathbb{N}, & i = 1, 2, .. n;\ k = 1, 2, .. D & (1.d) \end{array} \quad (1)$$

Equality (1.a) indicates that k parts have to be produced in the first k time periods ; equality (1.b) means that all demands have to be satisfied at time period D ; inequality (1.c) states that the number of parts of a given type i produced must not decrease with time. Hence, (1.a) and (1.c) together imply that exactly one part is produced per time period.

3.2 Objective functions

The objective function of the problem must describe the fact that we want to keep the effective production 'as close as possible' to the ideal and therefore minimise the distance between a feasible sequence and the ideal production. There is no consensus on which distance is the most adequate and many objective functions have been studied in the literature.
In this document, we consider *max-abs*, *sum-abs* and *sum-sqr* problems, which are the most widely studied.

Some authors consider that the main objective is to minimise the maximum of deviations with an objective function F_{max} of the form $F_{max} =$

$\max_{1\le i\le n,\,1\le k\le D} F_i(x_{i,k}-kr_i)$. For instance in [11, 5, 2], $F_i(x)=|x|$, for $i=1,2,..n$ and hence the objective function is $F_{max}=\max_{i,k}|x_{i,k}-kr_i|$ which leads to the *max-abs* problem. Note that the set of optimal solutions is the same for any identical pair functions F_i increasing on $[0,+\infty[$ (see [4]).
In Section 4.1, the max-abs problem is formulated as a perfect matching problem in a bipartite convex graph as in [11].

Other authors consider that the main objective is to minimise the total deviation with an objective function of the form $F_{sum}=\sum_{k=1}^{D}\sum_{i=1}^{n}F_i\left(x_{i,k}-kr_i\right)$. The problem is then denoted *total deviation* problem or *min-sum* problem. In [7], the deviations F_i, for $i=1,2,..,n$ are convex functions verifying

$$F_i\left(0\right)=0,\ i=1,2,..n \quad \text{and} \quad F_i\left(y\right)>0 \text{ for } y\neq 0,\ i=1,2,..n \tag{2}$$

For the min-sum problems, the most considered values for F_i are $F_i(x)=|x|$ and $F_i(x)=x^2$ [9, 7, 5]. In the first case, $F_{sum}=\sum_{k=1}^{D}\sum_{i=1}^{n}|x_{i,k}-kr_i|$, and we obtain the *sum-abs* problem. In the second case, $F_{sum}=\sum_{k=1}^{D}\sum_{i=1}^{n}\left(x_{i,k}-kr_i\right)^2$ and we refer to *sum-sqr* problems.
In section 4.2, JIT problems with total deviation are formulated as assignment problems [7].

3.3 Notations

We use the following notations.
- The set of all integers $i\in[a,\,b]$ is denoted $[a..b]$;
- The ceiling $\lceil a\rceil$ of a real number a is the smallest integer greater or equal to a ;
- The flooring $\lfloor a\rfloor$ of a real number a is the greatest integer smaller or equal to a ;
- The rounding $[a]$ of a real number a is the unique integer such that $a-\frac{1}{2}\le[a]<a+\frac{1}{2}$;

3.4 Two-part type problems

For the two-part type case, max-abs, sum-abs and sum-sqr objective functions can be optimised simultaneously, i.e. for any instance, there exists a sequence that is optimal for the 3 objective functions. For completeness, we prove this result that has been mentioned in [2]. It can be deduced from the following proposition :

Proposition 1 *For a two-part type problem with production rates r_1 and $r_2=1-r_1$, the solution x^* of the JIT problem defined by*

$$x^*_{1,k}=[kr_1] \qquad k=1,2,..D$$
$$x^*_{2,k}=k-[kr_1] \qquad k=1,2,..D$$

is optimal for all objective functions F_{max} or F_{sum} such that F_1 and F_2 are nonnegative functions increasing on $[0,+\infty[$ and verifying

$$\forall i = 1,2,\ \forall y \in \mathbb{R}, \qquad F_i\left(y\right) = F_i\left(-y\right) \tag{3}$$

Proof
Let x be a feasible solution of the JIT problem.

$$\forall k = 1,\ 2,..\,D,\ \ x_{1,k} = j \Rightarrow x_{2,k} = k - j$$

Hence, if $x_{1,k} = j$ we have

$$\begin{aligned} x_{2,k} - kr_2 &= k - j - kr_2 \\ &= k\left(1 - r_2\right) - j \\ &= kr_1 - j \\ &= -\left(x_{1,k} - kr_1\right) \end{aligned}$$

Therefore $\forall k = 1,\ 2,..\,D,\ \ |x_{1,k} - kr_1| = |x_{2,k} - kr_2|$
By definition of $x^*_{i,k}$, one has $|x^*_{i,k} - kr_i| \leq \frac{1}{2}$ for any $i = 1,2$ and $k = 1,\ 2,..\,D$.
Let x be a feasible solution of the JIT problem that differs from x^* at time period l. As shown in [2], such a solution verifies $|x_{1,l} - lr_1| \geq \frac{1}{2}$
Hence $\forall k = 1,\ 2..\,D, \forall i = 1,2,$, one has $|x_{i,k} - kr_i| \geq |x^*_{i,k} - kr_i|$.
The functions F_1 and F_2 verify $\forall i = 1,2,\ \forall y \in \left]-\infty,+\infty\right[,\ \ F_i\left(y\right) = F_i\left(|y|\right)$ and both functions are increasing on $[0,+\infty[$.
Therefore, $\forall k = 1,\ 2,..\,D,\ \ \forall i = 1,2,\ \ F_i\left(x_{i,k} - kr_i\right) \geq F_i\left(x^*_{i,k} - kr_i\right)$
Hence, for any objective function $F = F_{max}$ or $F = F_{sum}$ such that F_1 and F_2 are pair nonnegative functions increasing on $[0,+\infty[$, x^* is an optimal solution of the $F-$JIT problem. □

Remark This statement holds also if the functions F_1 and F_2 are identical, non-negative, decreasing on $]-\infty,0]$ and increasing on $[0,+\infty[$. The statement does not hold if the functions are different, non-negative, decreasing on $]-\infty,0]$, increasing on $[0,+\infty[$ and not pair. For the instance $d_a = 1$ and $d_b = 3$ and a total deviation objective function with $F_a(-\frac{1}{2}) + F_b(\frac{1}{2}) < F_a(\frac{1}{2}) + F_b(-\frac{1}{2})$, x^* is not optimal as the sequence $S = (b,\ a,\ b,\ b)$ is strictly better.

4 Just in time scheduling formulations

4.1 Maximum deviation and bipartite graphs

The max-abs problem, also denoted by MDJIT (maximum deviation JIT problem) in [2], has been analysed by Steiner and Yeomans [11] and Brauner and Crama [2]. Consider the recognition version of problem (1) with a max-abs objective function :

max-abs decision problem :
Input :
– $n \in \mathbb{N}$: number of part types
– $d_i \in \mathbb{N}$: demand for part type i, $i = 1, 2, .. n$
– $B \in \mathbb{Q}$: a bound
Question : Does there exist an $n \times D$ matrix $x = (x_{i,k})$ such that :

$$\begin{array}{ll} \max_{1\le i\le n,\, 1\le k\le D} |x_{i,k} - kr_i| \le B & \\ & \\ \sum_{i=1}^{n} x_{i,k} = k, & k = 1, 2, .. D \\ x_{i,D} = d_i, & i = 1, 2, .. n \\ 0 \le x_{i,k} - x_{i,k-1}, & i = 1, 2, .. n;\ k = 2, 3, .. D \\ x_{i,k} \in \mathbb{N}, & i = 1, 2, .. n;\ k = 1, 2, .. D \end{array} \tag{4}$$

We denote (i, j) the j^{th} part of type i. Consider an instance $(n, d_1, d_2, .. d_n, B)$ of the max-abs decision problem. Define the earliest time $E(i, j)$ and the latest time $L(i, j)$ to produce part (i, j) by

$$E(i,j) = \left\lceil \frac{j-B}{r_i} \right\rceil \text{ and } L(i,j) = \left\lfloor \frac{j-1+B}{r_i} + 1 \right\rfloor \tag{5}$$

In any feasible schedule, part (i, j) lies into the interval $[E(i, j)..L(i, j)]$ [2].

The max-abs decision problem can be formulated as a perfect matching problem in the bipartite graph $G = (V_1 \cup V_2, E)$ as follows [11]. The vertex set $V_1 = [1..D]$ represents the production time periods and V_2 is the set of all parts (i, j). An edge $(k,(i, j))$ is in E if and only if part (i, j) can be produced in time period k, i.e., if and only if $k \in [E(i, j)..L(i, j)]$.

The following statement describes a necessary and sufficient condition for the feasibility of a solution :

Proposition 2 [2] *The max-abs decision problem has a feasible solution if and only if the graph G has a perfect matching.*

From the previous proposition, we can trivially deduce :

Proposition 3 *The optimal objective value of the max-abs problem is the smallest B such that the corresponding graph G has a perfect matching*

This statement will be used in Section 5 to show optimality results for max-abs problems.

Remark When considering possible optimal value for the max-abs objective function, one can restrict oneself to $B = \frac{q}{D}$ with $q \in [1..D]$ since deviations are multiples of $\frac{1}{D}$ and since maximum deviation is always lower than 1 (see [11, 2]).

4.2 Assignment problem and costs matrix

This section describes the formulation of problems with total deviation (like sum-abs and sum-sqr) as assignment problems [7]. This formulation allows to find and prove optimality for those problems.

In [7], the authors show that, for any convex function F_i, $i = 1, 2, .. n$ verifying condition (2), problem (1) with the objective of minimising the total deviation F_{sum} can be reformulated as an assignment problem.
The ideal location $Z^*_{i,j} = \lceil k_{ij} \rceil$ of (i, j) is the ceiling of the unique instant k_{ij} satisfying

$$F_i\,(j - k_{ij}\, r_i) = F_i\,(j - 1 - k_{ij}\, r_i) \tag{6}$$

Let $C_{i,j,k}$ be the costs induced by placing (i, j) in the k^{th} position :

$$C_{i,j,k} = \begin{cases} \sum_{p=k}^{Z^*_{ij}-1} \psi_{ijp} & k < Z^*_{ij} \\ 0 & k = Z^*_{ij} \\ \sum_{p=Z^*_{ij}}^{k-1} \psi_{ijp} & k > Z^*_{ij} \end{cases} \tag{7}$$

where ψ_{ijp} is defined by

$$\psi_{ijp} = |F_i\,(j - p\, r_i) - F_i\,(j - 1 - p\, r_i)\,| \tag{8}$$

If (i, j) is produced before the ideal instant Z^*_{ij}, then ψ_{ijp} represent *inventory costs* and when (i, j) is produced after the ideal instant Z^*_{ij}, they represent *shortage costs*.

The assignment variables are defined as

$$y_{i,j,k} = \begin{cases} 1 & \text{if } (i, j) \text{ is produced in the time period } k \\ 0 & \text{otherwise} \end{cases}$$

We can now describe the assignment problem as :

$$\begin{array}{ll} & \mathit{minimise} \sum_{k=1}^{D} \sum_{i=1}^{n} \sum_{j=1}^{d_i} C_{i,j,k}\, y_{i,j,k} \\ \text{st} & \sum_{i=1}^{n} \sum_{j=1}^{d_i} y_{i,j,k} \;=\; 1, \quad \forall k = 1, 2, .. D \\ & \sum_{k=1}^{D} y_{i,j,k} \;=\; 1, \qquad\quad \forall i = 1, 2, .. n,\ \forall j = 1, 2, .. d_i \end{array} \tag{9}$$

The constraints of this problem indicate that only one part is produced at the time period k and that part (i, j) is produced only one time.

4.3 Optimality certificate for min-sum problems

In order to prove the optimality of the solutions proposed in Sections 6 and 7 for problems with total deviation, we will consider the assignment problem in term of graphs. The objective is to find a smallest perfect matching in the weighted-bipartite graph $G_w = (V_1 \cup V_2, E)$ where, as in Section 4.1, the vertex set $V_1 = [1..D]$ represents the production time periods and V_2 is the set of all parts (i, j).
In problem (9), there is no constraint regarding the maximum deviation. However we will afterward consider min-sum problems with bounded maximum deviation, in which case the edge set E can be described as before : an edge $(k, (i, j))$ is in E if and only if part (i, j) can be produced in time period k, i.e. if and only if $k \in [E(i, j)..L(i, j)]$.

Such a maximum deviation bounded total deviation problem will be denoted *B-bounded min-sum* problem.

Note that in a classical total deviation problem, a part (i, j) can be produced at any instant k. Therefore G_w is a complete bipartite graph. An edge $e = (k, (i, j)) \in E$ has the weight $w_e = C_{i,j,k}$. Denote A the vertex-edge incidence matrix of the graph G_w (i.e. the $2\,D \times D^2$ matrix such that $a_{l,m} = 1$ if the l^{th} vertex is incident to the m^{th} edge and 0 otherwise).

We can rewrite the problem as

$$\begin{array}{rl} & min\ wy \\ s.t. & Ay = 1 \\ & y \geq 0 \end{array} \tag{10}$$

Its dual is

$$\begin{array}{rl} & max\ 1z \\ s.t. & zA \leq w \\ & z \end{array} \tag{11}$$

This dual problem (11) can be interpreted as finding vertex weights z such that the sum of z_l for all $l \in V_1 \cup V_2$ is maximal and such that for all edge $(k, (i, j))$, one has $z_k + z_{i,j} \leq C_{i,j,k}$. Therefore, one has the following statement :

Proposition 4 *The objective value s of a total deviation problem is optimal if and only if there is a perfect matching M and a weight function $z : V_1 \cup V_2 \rightarrow \mathbb{R}$ such that for all edges $(k, (i, j))$, one has*

$$z_k + z_{i,j} \leq C_{i,j,k} \tag{12}$$

$$and \quad \sum_{k \in V_1} z_k + \sum_{(i,j) \in V_2} z_{i,j} = \sum_{(k,(i,j)) \in M} C_{i,j,k} = s \tag{13}$$

Proof

Consider a min-sum problem and his associated graph G_w. As the graph G_w is bipartite, the matrix A is totally unimodular. Therefore, the polyhedron $\{y;\, Ay = 1,\, y \geq 0\}$ is integer and the polyhedron $\{z; zA \leq w\}$ is rational as w is in $\mathbb{Q}$. Note that the vertices of $\{y;\, Ay = 1,\, y \geq 0\}$ are the incidence vectors of the perfect matchings of G_w. By duality, if the polytope $\{y;\, Ay = 1,\, y \geq 0\}$ is non empty, the problems (10) and (11) have optimal integer and rational solutions respectively. Furthermore, any couple of those optimal solutions (y^*, z^*) verifies $wy^* = 1z^*$. Denote M^* the perfect matching whose vector of incidence is y^*. One has

$$wy^* = \sum_{(k,(i,j))\in E} C_{i,j,k} y^*_{i,j,k} = \sum_{(k,(i,j))\in M^*} C_{i,j,k}$$

Therefore,

$$\sum_{k\in V_1} z^*_k + \sum_{(i,j)\in V_2} z^*_{i,j} = \sum_{(k,(i,j))\in M^*} C_{i,j,k}$$

Hence, if the min-sum problem has an optimal finite value s, then s verifies condition (13).

Let us prove the reverse implication. For all perfect matching M and all weight function $z : V_1 \cup V_2 \to \mathbb{R}$ verifying condition (12), one has

$$\sum_{(k,(i,j))\in M} C_{i,j,k} \geq \sum_{k\in V_1} z_k + \sum_{(i,j)\in V_2} z_{i,j}$$

Therefore,

$$\min_{M\, perfect\, matching} \sum_{(k,(i,j))\in M} C_{i,j,k} \geq \max_{z\, verifying\, (12)} (\sum_{k\in V_1} z_k + \sum_{(i,j)\in V_2} z_{i,j})$$

Hence, if there is is a perfect matching M^* and a weight function $z^* : V_1 \cup V_2 \to \mathbb{R}$ verifying conditions (12) such that

$$\sum_{k\in V_1} z^*_k + \sum_{(i,j)\in V_2} z^*_{i,j} = \sum_{(k,(i,j))\in M^*} C_{i,j,k}$$

then z^* and the incidence vector of y^* of M^* are optimal solution of the problems (11) and (10) respectively. □

Proposition 4 is used in Sections 6 and 7 to prove optimality of sum-abs and sum-sqr sequences by the means of graphs where matching, edge costs and corresponding vertex weights are represented.

4.4 Representation of graphs and f-factors

Most of the examples used in this document are of the same type : they consist in a number m of part types with demand $d_1 = d_2 = .. = d_m = 1$ and a number $n - m$ of part types with demand $d_{m+1} = d_{m+2} = .. = d_n$. In order to draw smaller graphs, we represent the vertices $(i, 1)$, $i = 1, 2, .. m$ as one vertex denoted $(a..\alpha, 1)$ (α being the m^{th} letter of the alphabet) and for each $j = 1, 2, .. d_n$ we represent the vertices (i, j) , $i = m+1, m+2, .. n$ as one vertex denoted $(\beta..\gamma, j)$ (β and γ being respectively the $(m+1)^{th}$ and n^{th} letters of the alphabet).

This transformation of the graph is justified by the notion of *f-factors*. Consider a graph $G = (V, E)$ and a function $f : V \longrightarrow \mathbb{N}$. A *$f$-factor* is a subgraph H of G such that $deg_H(v) = f(v), \forall v \in V$. It comes easily that a graph has a 1-factor if and only if it has a perfect matching. For further reading concerning f-factors theory, see [8, 1].

Consider an instance of the max-abs decision problem (n, d_1, d_2, $..d_n$, B) and the associated graph $G = (V_1 \cup V_2, E)$. We denote D the sum of all demands. Suppose that $d_{i_1} = d_{i_2}$ for some part types i_1 and i_2. Let $j \in [1 .. d_{i_1}]$. By definition (5) of the earliest and latest completion times, $E(i_1, j) = E(i_2, j)$ and $L(i_1, j) = L(i_2, j)$. Therefore, $\delta((i_1, j)) = \delta((i_2, j))$ with $\delta(v)$ the set of all edges incident to the vertex v. Hence, our transformation can be described by the following algorithm :

Algorithm *Simplify*
Initialisation
$L = \{1, 2, .. n\}$
$G = (V_1 \cup V_2, E)$
$f : V = V_1 \cup V_2 \longrightarrow \mathbb{N}, \quad f(v) = 1 \quad \forall v \in V$

While there are i_1, $i_2 \in L$ such that $d_{i_1} = d_{i_2}$ **Do**
For all $j \in 1, 2, .. d_{i_1}$ **Do**
$V_2 = V_2 \setminus (i_2, j)$
$f((i_1, j)) = f((i_1, j)) + f((i_2, j))$
$E = E \setminus \{((i_2, j), k) \in \delta((i_2, j))\}$
end
$L = L \setminus \{i_2\}$
end

The proof of the following proposition is rather easy but laborious and is only given out of completeness.

Proposition 5 *The graph G has a perfect matching, if and only if the graph G' obtained at the end of the algorithm has a f-factor.*

Proof
Remark first that the algorithm leaves invariant the set V_1 and all values of f over V_1.

Denote $G^p = (V_1 \cup V_2^p, E^p)$ the graph and f^p the function at the beginning of the p^{th} iteration. Since a graph has a perfect matching if and only if it has a 1-factor, we will consider the execution of the $(p+1)^{th}$ step of the algorithm and show that the graph G^p has a f^p-factor if and only if the graph G^{p+1} has a f^{p+1}-factor.

Suppose that this graph G^p has a f^p-factor H^p and that L^p contains i_1 and i_2 such that $d_{i_1} = d_{i_2}$. Construct the graph G^{p+1} and the function f^{p+1} of the $(p+1)^{th}$ iteration using those two elements of L^p. Let us show that G^{p+1} has an f^{p+1}-factor.
Consider the subgraph H^{p+1} of G^{p+1} such that $V(H^{p+1}) = V(G^{p+1})$ and $((i,j),k)$ is an edge of H^{p+1} if and only if
– $(i,j) \in V_2^{p+1}$ and $((i,j),k)$ is an edge of H^p.
– $i = i_1$ and $((i_2,j),k)$ is an edge of H^p
All the edges of H^{p+1} are indeed included in E^{p+1} since no edge of the form $((i_2,j),k)$ is in the set of edges of H^{p+1}. Note that H^p and H^{p+1} have the same number of edges.

Let $k \in V_1$. According to the definition of the edges of H^{p+1}, we have $deg_{H^{p+1}}(k) = deg_{H^p}(k)$. H^p is a f^p-factor. Therefore, $deg_{H^p}(k) = f^p(k)$. Furthermore, as remarked previously, $f^{p+1}(k) = f^p(k)$. Therefore, $deg_{H^{p+1}}(k) = deg_{H^p}(k) = f^p(k) = f^{p+1}(k)$.

Let $(i,j) \in V_2^{p+1}$ such that $i \neq i_1$. According to the definition of the edges of H^{p+1}, $deg_{H^{p+1}}((i,j)) = deg_{H^p}((i,j))$ and hence $deg_{H^{p+1}}((i,j)) = f^p((i,j))$ as H^p is a f^p-factor. f^p and f^{p+1} differ on V_2^{p+1} only in (i_1,l), $l \in 1, 2, .. d_{i_1}$. Therefore, $f^{p+1}((i,j)) = f^p((i,j))$. Hence, one has $deg_{H^{p+1}}((i,j)) = f^{p+1}((i,j))$

Let $j \in 1, 2, .. d_{i_1}$. Since H^p is a f-factor, $deg_{H^p}((i_1,j)) = f^p((i_1,j))$ and $deg_{H^p}((i_2,j)) = f^p((i_2,j))$. According to the definition of H^{p+1}, $deg_{H^{p+1}}((i_1,j)) = deg_{H^p}((i_1,j)) + deg_{H^p}((i_2,j))$. Therefore, $deg_{H^{p+1}}((i_1,j)) = f^p((i_1,j)) + f^p((i_2,j))$. Since $f^{p+1}((i_1,j)) = f^p((i_1,j)) + f^p((i_2,j))$ we have $deg_{H^{p+1}}((i_1,j)) = f^{p+1}((i_1,j))$.

Hence, all vertex v of H^{p+1} verify $deg_{H^{p+1}}(v) = f^{p+1}(v)$. Therefore, H^{p+1} is a f^{p+1}-factor of G^{p+1}.

Let us prove the reverse implication. Suppose that G^{p+1} has f^{p+1}-factor H^{p+1}. Denote F^{p+1} the set of the edges of H^{p+1}. For all $j = 1, 2, .. d_{i_1}$ consider a partition of the set $\{((i_1,j),k) \in F^{p+1}\}$ into 2 subsets F_j^1 and F_j^2 of respective sizes $f^p((i_1,j))$ and $f^p((i_2,j))$. Note that

$$|\{((i_1,j),k) \in F^{p+1}\}| = f^{p+1}((i_1,j))$$

since $deg_{H^{p+1}}((i_1, j)) = f^{p+1}((i_1, j))$. Therefore,

$$| \left\{ ((i_1, j), k) \in F^{p+1} \right\} | = f^p((i_1, j)) + f^p((i_2, j))$$

and hence the partition is possible.
We define the following subgraph H^p of G^p. All vertices of G^p are in H^p and an edge $((i, j), k)$ of G^p is in the edge set of the subgraph H^p if and only if
– $i \notin \{i_1, i_2\}$ and $((i, j), k) \in E^{p+1}$
– $i = i_1$ and $((i, j), k) \in F_j^1$
– $i = i_2$ and $((i, j), k) \in F_j^2$
In way similar to that of the implication, one can prove that for any vertex $k \in V_1^p$, one has $deg_{H^p}(k) = f^p(k)$ and that for any vertex (i, j) with $i \notin \{i_1, i_2\}$, one has $deg_{H^p}((i, j)) = f^p((i, j))$.

By definition of the subgraph H^p, one has for any $j \in [1..d_{i_1}]$, $deg_{H^p}((i_1, j)) = f^p((i_1, j))$ and $deg_{H^p}((i_2, j)) = f^p((i_2, j))$.

Therefore, H^p is a f^p-factor of G^p.

Hence, G^p has a f^p-factor if and only if G^{p+1} has a f^{p+1}-factor and the proposition holds. □

Remark From the proof of Proposition 5, one can deduce how to obtain a f-factor of the transformed graph from a perfect matching of the initial graph and reciprocally.

Example
Consider the instance $d = (1, 1, 4, 4)$ where $d_a = d_b = 1$ and $d_c = d_d = 4$. Figure 1 represents the corresponding graph for $B = \frac{7}{10}$. The thick edges are a perfect matching of the graph. When the part types with identical

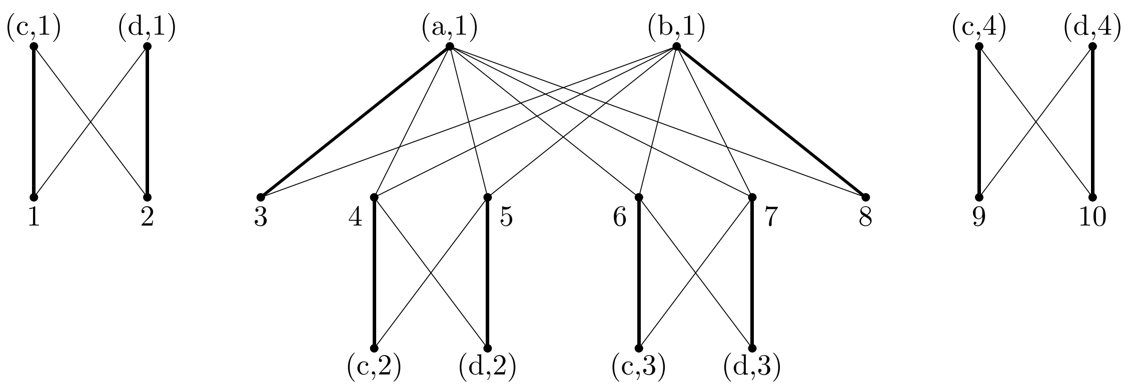

FIG. 1 – Bipartite graph for $d_a = d_b = 1$, $d_c = d_d = 4$ and $B = \frac{7}{10}$

demand are grouped, one obtains the graph of Figure 2.

Remark The thick edges of the graph Figure 2 are an f-factor obtained by

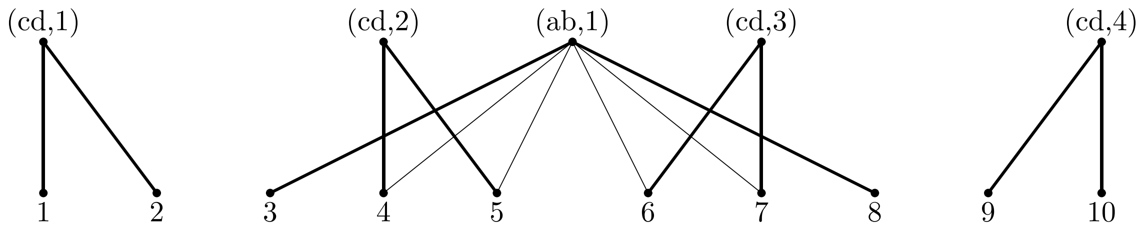

FIG. 2 – Vertices-simplified bipartite graph for $d_a = d_b = 1$, $d_c = d_d = 4$ and $B = \frac{7}{10}$

applying Algorithm Simplify to the subgraph of the graph Figure 1 induced by its perfect matching. □

To the transformation of Algorithm Simplify, we add the following : if the solution is symmetric, then only the first half of the graph will be represented. Proposition 6 and 7 imply that the graph is symmetric and some of the examples proposed have symmetric matchings.

Proposition 6 *Consider a $B-$bounded problem. Then, for all $(i, j) \in V_2$ we have*

$$E(i, d_i - j + 1) = D - L(i, j) + 1 \tag{14}$$
$$L(i, d_i - j + 1) = D - E(i, j) + 1 \tag{15}$$

Proof
From (5) one has

$$\begin{aligned} E(i, d_i - j + 1) &= \left\lceil \frac{d_i - j + 1 - B}{r_i} \right\rceil \\ &= \left\lceil D - \frac{j - 1 + B}{r_i} \right\rceil \\ &= D - \left\lfloor \frac{j - 1 + B}{r_i} \right\rfloor \\ &= D - \left\lfloor \frac{j - 1 + B}{r_i} + 1 \right\rfloor + 1 \\ &= D - L(i, j) + 1 \end{aligned}$$

The equality (15) is obtained by replacing j by $d_i - j + 1$ in (14).

Therefore, the proposition holds. □

We shall now prove that the costs are symmetric. We will first consider 2 lemmas.

Lemma 1 *Let F_i be a pair convex function verifying condition (2). The ideal location $Z^*_{i,j}$ of part (i,j) is such that*

$$Z^*_{i,j} = \begin{cases} D - Z^*_{i,d_i-j+1} & \text{if } k_{i,j} \text{ is integer} \\ D - Z^*_{i,d_i-j+1} + 1 & \text{otherwise} \end{cases}$$

Proof

For any pair convex function F_i verifying condition (2), one has $F_i(x) = F_i(x-1)$ if and only if $x = \frac{1}{2}$. Therefore, according to the definition (6) of the ideal location $k_{i,j}$ of part (i,j), we have $k_{i,j} = \frac{2j-1}{2r_i}$.

Consider a triplet (i,j,k). Let q be an integer and α a rational such that $k_{i,j} = q + \alpha$ and $\alpha \in [0,1[$. One has

$$\begin{aligned} Z^*_{i,d_i-j+1} &= \lceil k_{i,d_i-j+1} \rceil \\ &= \left\lceil \frac{2(d_i - j + 1) - 1}{2r_i} \right\rceil \\ &= \left\lceil D - \frac{2j-1}{2r_i} \right\rceil \\ &= \lceil D - q - \alpha \rceil \\ &= D - q \end{aligned}$$

Therefore, if $k_{i,j}$ is integer, i.e if $Z^*_{i,j} = k_{i,j} = q$ one has $Z^*_{i,d_i-j+1} = D - Z^*_{i,j}$ and if $k_{i,j}$ is not integer, i.e. if $Z^*_{i,j} = q+1$, one has $Z^*_{i,d_i-j+1} = D - Z^*_{i,j} + 1$ Hence, the lemma holds. □

Lemma 2 *Let F_i be a pair convex function verifying condition (2). For all integer p in $[1..D-1]$ one has*

$$\psi_{i,j,p} = \psi_{i,d_i-j+1,D-p}$$

Proof

Let m be an integer. From (8), one has

$$\begin{aligned} \psi_{i,d_i-j+1,D-p} &= |F_i(d_i - j + 1 - (D-p)r_i) \\ &\quad - F_i(d_i - j + 1 - (D-p)r_i - 1)| \\ &= |F_i(-(-d_i + j - 1 + d_i - pr_i)) \\ &\quad - F_i(-(-d_i + j + d_i - pr_i))| \\ &= |F_i(-(j - 1 - pr_i)) - F_i(-(j - pr_i))| \end{aligned}$$

Since F_i is pair, we have $\psi_{i,d_i-j+1,D-p} = |F_i(j-1-pr_i) - F_i(j-pr_i)|$ and therefore

$$\psi_{i,d_i-j+1,D-p} = \psi_{i,j,p}$$

□

Proposition 7 *Let F_i be pair convex functions verifying condition (2). For all triplet (i,j,k), one has*

$$C_{i,j,k} = C_{i,d_i-j+1,D-k+1}$$

Proof

Consider a triplet (i,j,k). Let us show that $C_{i,j,k} = C_{i,d_i-j+1,D-k+1}$. According to Lemma 1, one has $Z^*_{i,j} = D - Z^*_{i,d_i-j+1} + \epsilon$ with $\epsilon \in \{0,1\}$. Denote $\bar{\epsilon}$ the integer $1-\epsilon$.

Suppose that $k < Z^*_{i,j}$. From the Lemma 2 we deduce :

$$\begin{aligned}\sum_{p=k}^{Z^*_{i,j}-1} \psi_{i,j,p} &= \sum_{p=k}^{D-Z^*_{i,d_i-j+1}-\bar{\epsilon}} \psi_{i,d_i-j+1,D-p} \\ &= \sum_{p=Z^*_{i,d_i-j+1}+\bar{\epsilon}}^{D-k} \psi_{i,d_i-j+1,p}\end{aligned}$$

If $\bar{\epsilon} = 1$, then $k_{i,j}$ is integer and hence we can deduce from the proof of Lemma 1 that k_{i,d_i-j+1} is also integer. Therefore, together with the definition (6) of k_{i,d_i-j+1} one has $\psi_{i,d_i-j+1,Z^*_{i,d_i-j+1}} = 0$. Hence, one has

$$\sum_{p=k}^{Z^*_{i,j}-1} \psi_{i,j,p} = \sum_{p=Z^*_{i,d_i-j+1}}^{(D-k+1)-1} \psi_{i,d_i-j+1,p}$$

Therefore, according to the definition (7) of costs, one has

$$C_{i,j,k} = C_{i,d_i-j+1,D-k+1}$$

Suppose that $k = Z^*_{i,j}$. If $k_{i,j}$ is integer, k_{i,d_i-j+1} is also integer and equal to $D-k$ (see Lemma 1). Therefore, we have $C_{i,d_i-j+1,D-k+1} = \sum_{p=Z^*_{i,d_i-j+1}}^{D-k+1-1} \psi_{i,d_i-j+1,p} = \psi_{i,j,Z^*_{i,d_i-j+1}} = 0$. Since $k = Z^*_{i,j}$, we have $C_{i,j,k} = 0$. Hence

$$C_{i,j,k} = C_{i,d_i-j+1,D-k+1}$$

Suppose that $k > Z^*_{i,j}$. From the Lemma 2 we deduce :

$$\begin{aligned}\textstyle\sum_{p=Z^*_{i,j}}^{k-1} \psi_{i,j,p} &= \textstyle\sum_{p=D-Z^*_{i,d_i-j+1}+\epsilon}^{k-1} \psi_{i,d_i-j+1,D-p} \\ &= \textstyle\sum_{p=D-k+1}^{Z^*_{i,d_i-j+1}-\epsilon} \psi_{i,d_i-j+1,p}\end{aligned}$$

If $\epsilon = 0$, then k_{i,d_i-j+1} is integer and $\psi_{i,d_i-j+1,Z^*_{i,d_i-j+1}} = 0$. Therefore one has

$$\sum_{p=Z^*_{i,j}}^{k-1} \psi_{i,j,p} = \sum_{p=D-k+1}^{Z^*_{i,d_i-j+1}-1} \psi_{i,d_i-j+1,p}$$

Hence, according to the definition (7) of costs, one has

$$C_{i,j,k} = C_{i,d_i-j+1,D-k+1}$$

Therefore, the proposition holds. □

Proposition 6 and 7 imply that edges and costs are symmetric, but not necessarily the optimal matching. Nonetheless, in many cases, this matching is symmetric and we adopt a representation where only the first half of the graph is drawn. Dots then indicate that the graph is not completely represented (see Figure 3).

Example
We consider the preceding example with $d = (1, 1, 4, 4)$ where $d_a = d_b = 1$ and $d_c = d_d = 4$. The graph of Figure 2 is symmetric. Therefore we can simplify it to obtain the graph of Figure 3. □

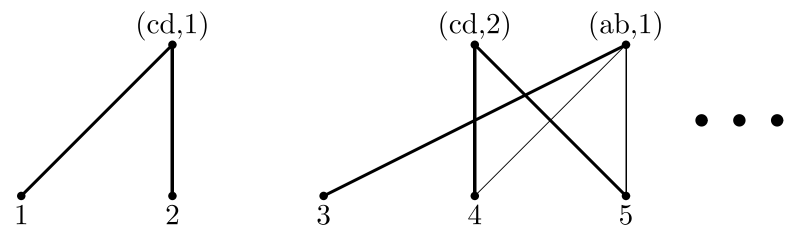

FIG. 3 – Vertices and symmetry simplified bipartite graph for $d_a = d_b = 1$, $d_c = d_d = 4$ and $B = \frac{7}{10}$

Remark For a given statement T, we denote *D-smallest example* an example of T such that no instance with a smaller D is example of the statement T. A *n-smallest example* is defined similarly.

5 Max-abs and sum-abs problems

This section is motivated by a conjecture proposed by Kubiak [6] that compares max-abs and sum-abs problems. They only consider the case where the deviations are the absolute values of the differences between the ideal and effective production.

Conjecture 1 [6] *For any instance, there is a minimal sum-abs sequence that is minimal for the max-abs problem.*

In [5], the authors state that this conjecture does not hold but they do not give any of the counter-examples found. Here we give the D-smallest and the n-smallest together with an easy argument to verify that they are indeed.

Note first that not all optimal max-abs sequences are optimal for the sum-abs problem. Consider for example the instance $d_a = 1$ and $d_b = d_c = 3$. The sequence $S = (c, b, c, a, b, c, b)$ is optimal for the max-abs objective function with a maximum deviation of $\frac{5}{7}$. However, it is not optimal for the sum-abs objective function as the sequence $S = (c, b, a, c, b, c, b)$ is strictly better.

Proposition 8 *The instance $d = (1, 1, 4, 4)$ is a D-smallest counter-example for Conjecture 1.*

Proof
For $B = \frac{6}{10}$, we obtain the graph of Figure 4 which has no perfect matching. The graph of Figure 2 obtained for $B = \frac{7}{10}$ has a perfect matching. Therefore the optimal value of the max-abs objective function is $B^* = \frac{7}{10}$ (see Proposition 3). The graph of Figure 2 has a unique perfect matching up to the

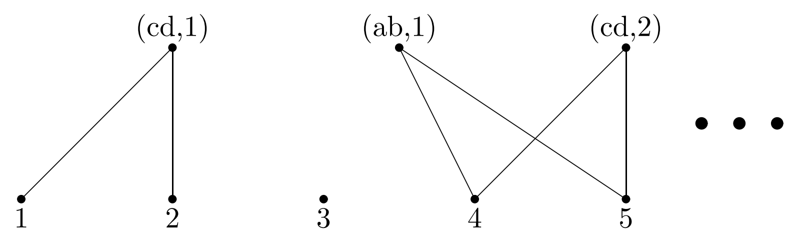

FIG. 4 – Bipartite graph for $d_a = d_b = 1$, $d_c = d_d = 4$ and $B = \frac{6}{10}$

permutation of parts corresponding to the same vertex of the graph. It corresponds to the sequence $S = (c,\ d,\ a,\ c,\ d,\ d,\ c,\ b,\ d,\ c)$. For this sequence, the deviations are given in Table 1. Only the first half of the table has been represented as for $k = 6,\ 7, ..\ 9$ one has $|x_{a,k} - kr_a| = |x_{b,D-k} - (D-k)r_b|$ and $|x_{b,k} - kr_b| = |x_{a,D-k} - (D-k)r_a|$ and $|x_{c,k} - kr_c| = |x_{c,D-k} - (D-k)r_c|$ and $|x_{d,k} - kr_d| = |x_{d,D-k} - (D-k)r_d|$ and hence the total deviation at time $k = 6,\ 7, ..\ 9$ is the same as the total deviation at time $10 - k$. The total deviation for this unique sequence is $11, 8$. It is therefore the optimal value for the B^*-bounded sum-abs problem.

The sequence $S = (c,\ d,\ c,\ d,\ a,\ b,\ d,\ c,\ d,\ c)$ has a total deviation of $11, 4$. Its deviations are symmetric with regard to time period $k = 5$ and therefore only the first half of the table has been represented on Table 2. The sum of deviations of this sequence is lower than the optimal sum of deviations for the B^*-bounded sum-abs problem. Therefore the max-abs and sum-abs objective functions cannot be optimised simultaneously : no optimum schedule for the max-abs problem is optimal for the sum-abs problem.

Exhaustive search with $D \leq 9$ proves that $d = (1,\ 1,\ 4,\ 4)$ is indeed a D-smallest counter-example of Conjecture 1. $\square$

TAB. 1 – Deviations for the sequence $S = (c, d, a, c, d, d, c, b, d, c)$ of the instance $d_a = d_b = 1$ and $d_c = d_d = 4$

time period		1	2	3	4	5
production		c	d	a	c	d
a	$x_{a,k}$	0	0	1	1	1
	deviation	0.1	0.2	0.7	0.6	0.5
	ideal	0.1	0.2	0.3	0.4	0.5
b	$x_{b,k}$	0	0	0	0	0
	deviation	0.1	0.2	0.3	0.4	0.5
	ideal	0.1	0.2	0.3	0.4	0.5
c	$x_{c,k}$	1	1	1	2	2
	deviation	0.6	0.2	0.2	0.4	0
	ideal	0.4	0.8	1.2	1.6	2
d	$x_{d,k}$	0	1	1	1	2
	deviation	0.4	0.2	0.2	0.6	0
	ideal	0.4	0.8	1.2	1.6	2

TAB. 2 – Deviations for the sequence $S = (c, d, c, d, a, b, d, c, d, c)$ of the instance $d_a = d_b = 1$ and $d_c = d_d = 4$

time periode		1	2	3	4	5
production		c	d	c	d	a
a	$x_{a,k}$	0	0	0	0	1
	deviation	0.1	0.2	0.3	0.4	0.5
	ideal	0.1	0.2	0.3	0.4	0.5
b	$x_{b,k}$	0	0	0	0	0
	deviation	0.1	0.2	0.3	0.4	0.5
	ideal	0.1	0.2	0.3	0.4	0.5
c	$x_{c,k}$	1	1	2	2	2
	deviation	0.6	0.2	0.8	0.4	0
	ideal	0.4	0.8	1.2	1.6	2
d	$x_{d,k}$	0	1	1	2	2
	deviation	0.4	0.2	0.2	0.4	0
	ideal	0.4	0.8	1.2	1.6	2

The preceding example is not a n-smallest counter-example of Conjecture 5

Proposition 9 *For $n = 3$ there are instances of the JIT problem such that no sequence optimises sum-abs and max-abs objective functions simultaneously.*

Proof
For $n = 3$, the $D-$smallest example is the instance $d = (2, 7, 17)$. For this instance, the minimum of the maximum deviation is $B^* = \frac{16}{26}$ and we have the graph of Figure 5 where only the left-hand side of the graph has been represented. The thick edges are in any perfect matching of the graph. Therefore the graph has exactly two perfect matchings that corres-

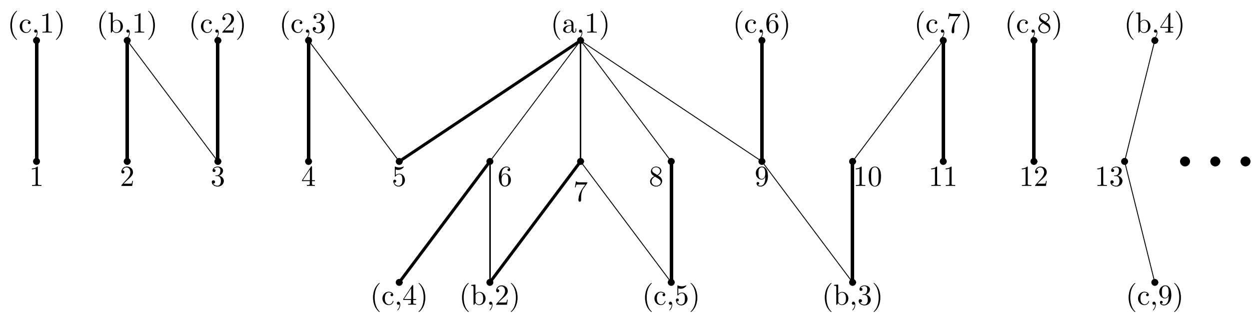

FIG. 5 – Bipartite graph for $d_a = 2$, $d_b = 7$, $d_c = 17$ and $B = \frac{16}{26}$

pond to the same sum of deviations :$\frac{572}{26}$ (which is met for instance for the sequence $S = (c, b, c, c, a, c, b, c, c, b, c, c, b, c, c, c, b, c, c, b, c, a, c, c, b, c)$). The sequence $S = (c, b, c, c, b, c, c, a, c, b, c, c, b, c, c, c, b, c, a, c, c, b, c, c, b, c)$ has a sum-abs objective function equal to $\frac{556}{26}$. Hence, no sequence is minimal for both functions. □

Remark $n = 3$ is the smallest n such that there are instances with no sequence optimising sum-abs and max-abs objective functions simultaneously. (see Proposition 1)

6 1-bounded-sum-abs and sum-sqr problems

The set of all possible solutions of 1-bounded problems is considerably smaller than the one of its unbounded version. Considering 1-bounded problems instead of unbounded ones therefore allows computational improvements on exact min-sum methods such as computing the smallest perfect matching, which can be done in $O(nD^2 \, log \, D)$ for a 1-bounded min-sum problem instead of $O(D^3 \, log \, D)$ for its unbounded version (see [12]). For all instance, it is possible to find a sequence with maximum deviation lower than 1. Hence, it is interesting to know if such a 1-bounded sequence can always be found that optimises total deviation problems, i.e., if the 1-bounded min-sum problems have the same optimal objective value that their unbounded versions. This section shows that it is not the case for sum-abs and sum-sqr problems and compares their solutions for bounded and unbounded problems.

Let P be the polyhedron of feasible solutions of the balanced schedule problem such that the maximum deviation is lower than 1.

$$\begin{array}{rcll} P & = & \{y; \sum_{i=1}^{n} \sum_{j=1}^{d_i} y_{i,j,k} = 1, & \forall k = 1, 2, .. D \\ & & \sum_{k=1}^{D} y_{i,j,k} = 1, & \forall i = 1, 2, .. n, \forall j = 1, 2, .. di \\ & & |\sum_{m=1}^{k} \sum_{j=1}^{d_i} y_{i,j,m} - kr_i| \leq 1, & \forall k = 1, 2, .. D, \forall i = 1, 2, .. n\} \end{array} \tag{16}$$

Theorem 1 *Any optimal sum-abs sequence y, such that $y \in P$, is an optimal sequence for the sum-sqr problem.*

In other words, if there is an optimal solution of the sum-abs problem such that the maximum deviation is lower than 1 then this solution is also optimal for the sum-sqr problem.

In order to prove this result, we will prove the following lemma that compares the costs for the sum-abs and sum-sqr problems.

Lemma 3 *For any solution $y \in P$, one has*

$$\forall i = 1, 2, .. n, \forall j = 1, 2, .. d_i, \forall k = 1, 2, .. D, y_{i,j,k} = 1 \Rightarrow C^a_{i,j,k} = C^s_{i,j,k}$$

where C^a is the cost matrix calculated for the sum-abs problem and C^s is the cost matrix calculated for the sum-sqr problem.

Proof

Denote $\lceil k^a_{ij} \rceil$ and $\lceil k^s_{ij} \rceil$ the ideal locations of (i, j) in the production sequence for the sum-abs problem and the sum-sqr problem respectively. Since $|x| = |x - 1|$ if and only if $x = \frac{1}{2}$ and $x^2 = (x-1)^2$ if and only if $x = \frac{1}{2}$, one has $k^a_{ij} = k^s_{ij}$. Therefore, we will denote $Z^*_{ij} = \lceil k^a_{ij} \rceil = \lceil k^s_{ij} \rceil$ the ideal location of (i, j) in the production sequence for both problems.

Let $C^a_{i,j,k}$ and $C^s_{i,j,k}$ be the costs induced by placing (i, j) in the k^{th} position and ψ^a_{ijp} and ψ^s_{ijp} the inventory and shortage costs for each problem. The function $f(x) = |x^2 - (x-1)^2| - ||x| - |x-1||$ is equal to zero if and only if $x \in [0, 1]$. Therefore we have the following statement :

$$\psi^a_{ijp} = \psi^s_{ijp} \text{ if and only if } j - p\, r_i \in [0, 1]$$

Let's prove that for any $y \in P$, if (i,j,k) is such that $y_{i,j,k} = 1$ then $C^a_{i,j,k} = C^s_{i,j,k}$.

Consider a solution $y \in P$ and i,j,k such that $y_{i,j,k} = 1$.
One has $|j - kr_i| \leq 1$ and hence $j - kr_i \leq 1$. We shall consider 3 cases depending on the respective positions of k and $Z^*_{i,j}$.

<u>Case 1 :</u> $k < Z^*_{ij}$

For all $p = k,\ k+1, .. Z^*_{i,j} - 1$, one has

$$j - p\, r_i > \frac{1}{2}$$

since the function $g\,(x) = j - xr_i$ is decreasing and $g\,(k_{i,j}) = \frac{1}{2}$. Therefore, one has $0 \leq j - p\, r_i \leq 1$ and hence

$$\forall p = k,\ k+1, .. Z^*_{i,j} - 1,\ \ \psi^a_{ijp} = \psi^s_{ijp}$$

<u>Case 2 :</u> $k > Z^*_{ij}$

Since $y_{i,j,k} = 1$ one has $x_{i,k-1} = j - 1$ and together with $y \in P$ one obtains

$$|j - 1 - (k-1)\, r_i| \leq 1$$

Hence, $-1 \leq j - 1 - (k-1)\, r_i$ and therefore

$$0 \leq j - (k-1)\, r_i$$

As the function g is decreasing, for all $p = Z^*_{i,j},\ Z^*_{i,j} + 1, .. k - 1$, we have $j - pr_i \in [0,1]$. Hence one has

$$\forall p = Z^*_{i,j},\ Z^*_{i,j} + 1, .. k - 1, \qquad \psi^a_{ijp} = \psi^s_{ijp}$$

<u>Case 3 :</u> $k = Z^*_{ij}$

In this case, both costs are zero.

Therefore, according to the definition of the costs, one has

$$C^a_{ijk} = C^s_{ijk}$$

in all the three cases. Hence, the lemma holds. □

Proof of Theorem 1
The function $f(x) = |x^2 - (x-1)^2| - ||x| - |x-1||$ is non negative. We therefore obtain

$$\forall i,j,k, \qquad C^a_{i,j,k} \leq C^s_{i,j,k}$$

Hence, for any feasible solution y of the JIT problem

$$\sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^s_{i,j,k}\, y_{i,j,k} \geq \sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^a_{i,j,k}\, y_{i,j,k} \tag{17}$$

Let $y^* \in P$ be an optimal solution of the sum-abs problem. Since for any $y \in P$, $y_{i,j,k} = 1 \Rightarrow C^s_{i,j,k} = C^a_{i,j,k}$ one has

$$\sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^s_{i,j,k}\, y^*_{i,j,k} = \sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^a_{i,jk}\, y^*_{i,j,k} \leq \sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^a_{i,jk}\, y_{i,j,k} \tag{18}$$

Combining (17) and (18), we obtain for any feasible solution y,

$$\sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^s_{i,j,k}\, y_{i,j,k} \geq \sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^s_{i,j,k}\, y^*_{i,j,k}$$

and hence y^* is an optimal solution of the sum-sqr problem. □

Corollary 1 *Any instance such that a pair of optimal solutions* (y^*, Y^*) *of the sum-abs and sum-sqr problems respectively verifies*

$$\sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^s_{i,j,k}\, Y^*_{i,j,k} > \sum_{i=1}^{n}\sum_{j=1}^{d_i}\sum_{k=1}^{D} C^a_{i,j}\, y^*_{i,j,k}$$

is such that the sum-abs problem has no optimal solution with maximum deviation lower or equal to 1.

Proof
Directly from the proof of Theorem 1. □

The following statement proves that the converse implication of Theorem 1 is not true : one cannot reverse the roles of sum-abs and sum-sqr problems. It also refutes the following conjecture of Steiner and Yeomans :
Conjecture 2 [12] *The 1-bounded sum-abs problem and the sum-abs problem are equivalent.*

Proposition 10 *There are instances of the JIT problem such that the sum-sqr problem has an optimal solution in* P *and the sum-abs problem has no optimal solution in* P.

Example
For the instance $d = (1,1,1,1,1,1,1,1,1,6,6,6,6)$ the sum-sqr problem has a solution in P. This solution is represented on the graph Figure 6. The thick edges are the matching solution edges, the italic numbers are the costs of the edges as defined in Section 4.2, the thick numbers are the values of a dual solution verifying conditions (12) and (13). Note that the costs and the dual solution have been multiplied by D in order to obtain integers. All edges have not been represented, but all important ones are drawn. Proposition

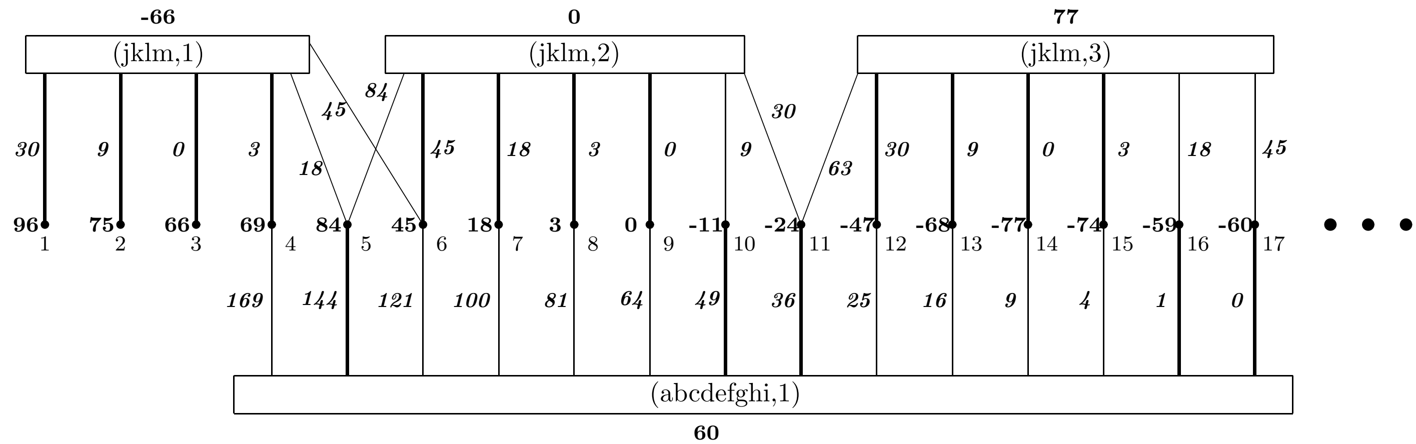

FIG. 6 – Bipartite graph for $d_a = d_b = ...d_i = 1$, $d_j = d_k = ...d_m = 6$ and weights for the min-sum-sqr problem

4 implies that this solution is indeed optimal. Denote that solution Y^*. It is in P as maximum of the deviations is $\frac{30}{33}$. Consider the sequence $S = (j,k,l,m,j,k,l,m,a,b,c,j,k,l,m,d,e,f,j,k,l,m,g,h,i,j,k,l,m,j,k,l,m)$ and the corresponding assignment variable y. One has

$$\sum_{i=1}^{n} \sum_{j=1}^{d_i} \sum_{k=1}^{D} C^s_{i,j,k}\, Y^*_{i,j,k} > \sum_{i=1}^{n} \sum_{j=1}^{d_i} \sum_{k=1}^{D} C^a_{i,j}\, y_{i,j,k}$$

Therefore, for any optimal solution y^* of the sum-abs problem, one has

$$\sum_{i=1}^{n} \sum_{j=1}^{d_i} \sum_{k=1}^{D} C^s_{i,j,k}\, Y^*_{i,j,k} > \sum_{i=1}^{n} \sum_{j=1}^{d_i} \sum_{k=1}^{D} C^a_{i,j}\, y^*_{i,j,k}$$

Hence by Corollary 1, no optimal solution of the sum-abs problem is in P.
□

The following proposition proves that Theorem 1 in [5] is not exact. However this fact has been notified to us by one of the authors who sent us a counter example of this theorem with a total demand $D = 100$. Here we give an instance with $D = 33$ and we explain how we found that counter-example.

Proposition 11 *The sum-abs and sum-sqr objective functions cannot be optimised simultaneously for all instance of the JIT problem.*

Proof
To show that the set of sum-sqr and sum-abs optimal solutions are disjoint for the instance $d = (1, 1, 1, 1, 1, 1, 1, 1, 1, 6, 6, 6, 6)$, we first obtain the optimal objective value for both problems, say S_s and S_a and then add in the sum-sqr problem a constraint fixing the sum-abs objective to its optimal value. If the optimal objective value of the problem

$$\begin{array}{lll} & minimise \sum_{k=1}^{D} \sum_{i=1}^{n} \sum_{j=1}^{d_i} C^s_{i,j,k}\, y_{i,j,k} & \\ \text{st} & \sum_{k=1}^{D} \sum_{i=1}^{n} \sum_{j=1}^{d_i} C^a_{i,j,k}\, y_{i,j,k} = S_a & \\ & \sum_{i=1}^{n} \sum_{j=1}^{d_i} y_{i,j,k} = 1, & \forall k = 1,\, 2, ..\, D \\ & \sum_{k=1}^{D} y_{i,j,k} = 1, & \forall i = 1,\, 2, ..\, n,\ \forall j = 1,\, 2, ..\, d_i \end{array} \tag{19}$$

is S_s, then sum-abs and sum-sqr problems can be optimised simultaneously. As it is not the case for the instance $d = (1, 1, 1, 1, 1, 1, 1, 1, 1, 6, 6, 6, 6)$, it is indeed a counter-example. □

Theorem 1 proves that sum-abs and sum-sqr problems are simultaneously optimised if an optimal solution of the sum-abs problem is in P. The following proposition states that it is not a necessary condition.

Proposition 12 *There are instances such that sum-abs and sum-sqr objective functions can be simultaneously optimised and the sum-abs and sum-sqr problems have no optimal solution in P.*

Example
For a bound $B = 1$ of the maximum deviation, the graph corresponding to the instance $d = (1, 1, 1, 1, 1, 1, 1, 6, 6, 6, 6)$ is on Figure 7. As the costs for sum-abs and sum-sqr problems are identical for $B \leq 1$, the solution represented is optimal for each problem. This solution is not optimal for any of the unbounded total deviation problems. An optimal solution for both problems and their dual is represented on the graphs Figure 8 and 9.

□

7 Sum-sqr and max-abs problems

As the optimal objective value of the max-abs problem is always lower than 1 (as proved in [11, 2]), Theorem 1 implies the following statement :

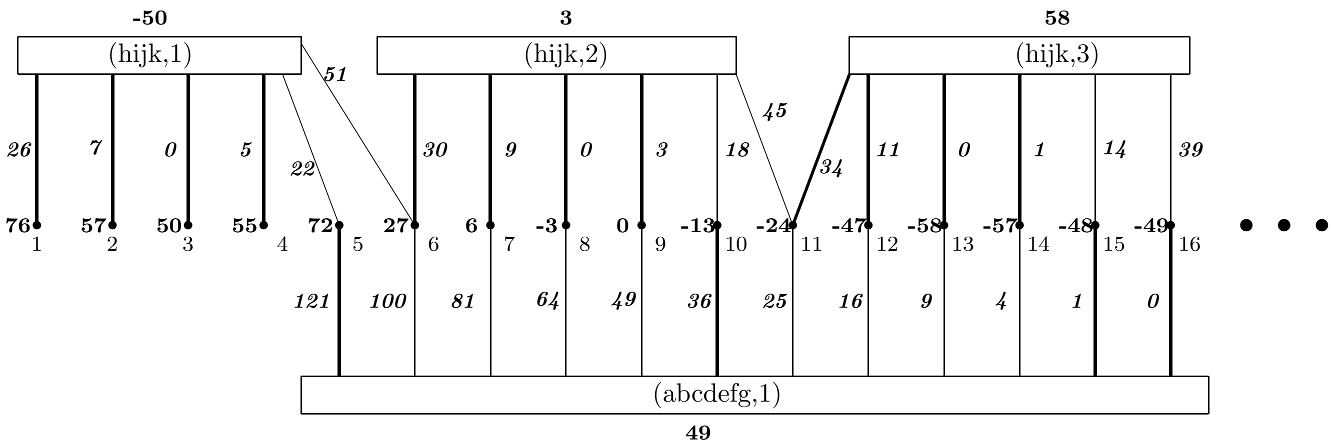

FIG. 7 – Bipartite graph for $d_a = d_b = ...d_g = 1$, $d_h = .. = d_k = 6$ and weights for the min-sum problems with B=1

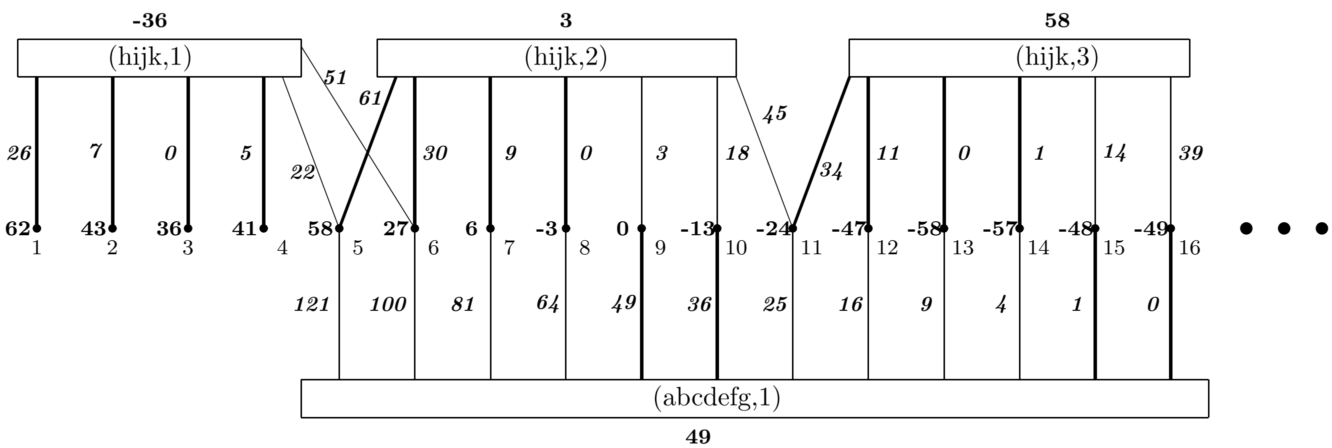

FIG. 8 – Bipartite graph for $d_a = d_b = ...d_g = 1$, $d_h = .. = d_k = 6$ and weights for the min-sum-abs problem

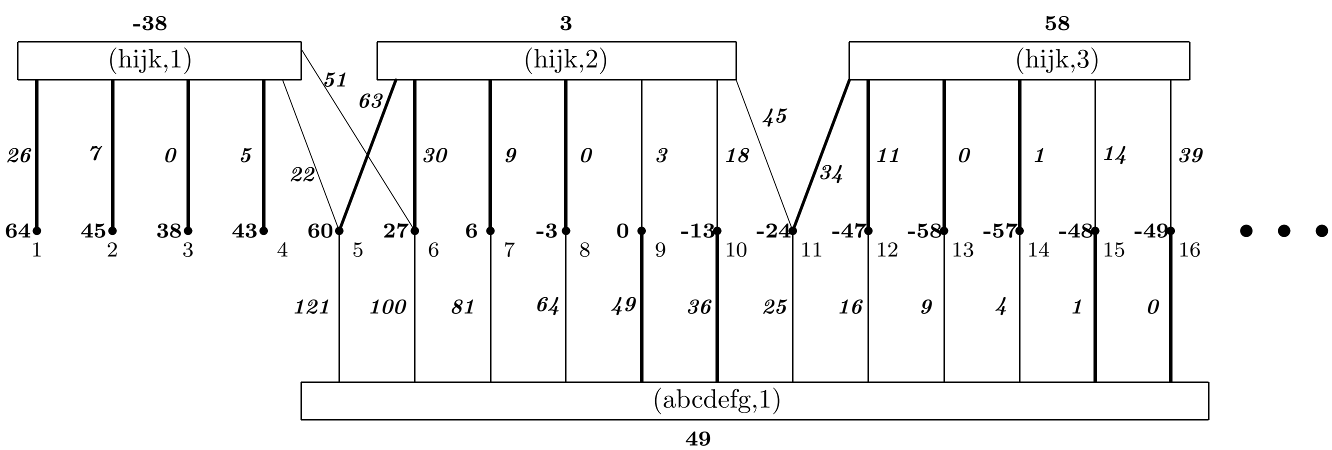

FIG. 9 – Bipartite graph for $d_a = d_b = ...d_g = 1$, $d_h = .. = d_k = 6$ and weights for the min-sum-sqr problem

Proposition 13 *If there is a solution x^* that is optimal for the max-abs and the sum-abs objective functions then x^* is also optimal for the sum-sqr objective function.*

The converse implication does not hold.

Proposition 14 *There are instances such that the max-abs and the sum-sqr objective functions can be optimised simultaneously but the sum-abs and the max-abs objective functions cannot be optimised simultaneously.*

Proof
Consider the instance $d = (1, 1, 1, 1, 1, 1, 1, 1, 1, 7, 7, 7)$. For this instance the optimal value for the max-abs problem is $B^* = \frac{26}{30}$. Using Proposition 4, one can verify that the matching on the graph on Figure 10 represents an optimal solution denoted Y^* of the sum-sqr problem. For this solution the maximum deviation is B^*. An optimal solution y^* of the sum-abs problem is represented on the graph Figure 11. One has

$$\sum_{i=1}^{n} \sum_{j=1}^{d_i} \sum_{k=1}^{D} C^s_{i,j,k}\, Y^*_{i,j,k} > \sum_{i=1}^{n} \sum_{j=1}^{d_i} \sum_{k=1}^{D} C^a_{i,j}\, y^*_{i,j,k}$$

Hence, according to Corollary 1, the sum-abs problem has no optimal solution such that the maximum deviation is lower than 1. Therefore the sum-abs problem has no solution such that the maximum deviation is B^*. □

Note that this result implies Proposition 10.

The example of the proof of Proposition 14 allows to prove several statements on the set of all solutions of the sum-sqr problems. First, remark that the optimal sum-abs sequence represented on Figure 11 is also optimal for the sum-sqr problem. It leads to the following statement :

Proposition 15 *There are instances such that the set of all optimal solutions of the sum-abs and sum-sqr problems have common elements but are not identical.*

Proof
Consider the instance $d = (1, 1, 1, 1, 1, 1, 1, 1, 1, 7, 7, 7)$. An optimal solution y^* of the sum-abs problem is represented on the graph Figure 11. This solution is also optimal for the sum sqr problem and therefore, the sets of all optimal solutions of sum-abs and sum-sqr problems have at least a common element. The optimal solution of sum-sqr problem represented on Figure 10 is optimal for the sum-sqr problem, but not for the sum-abs problem. Hence,

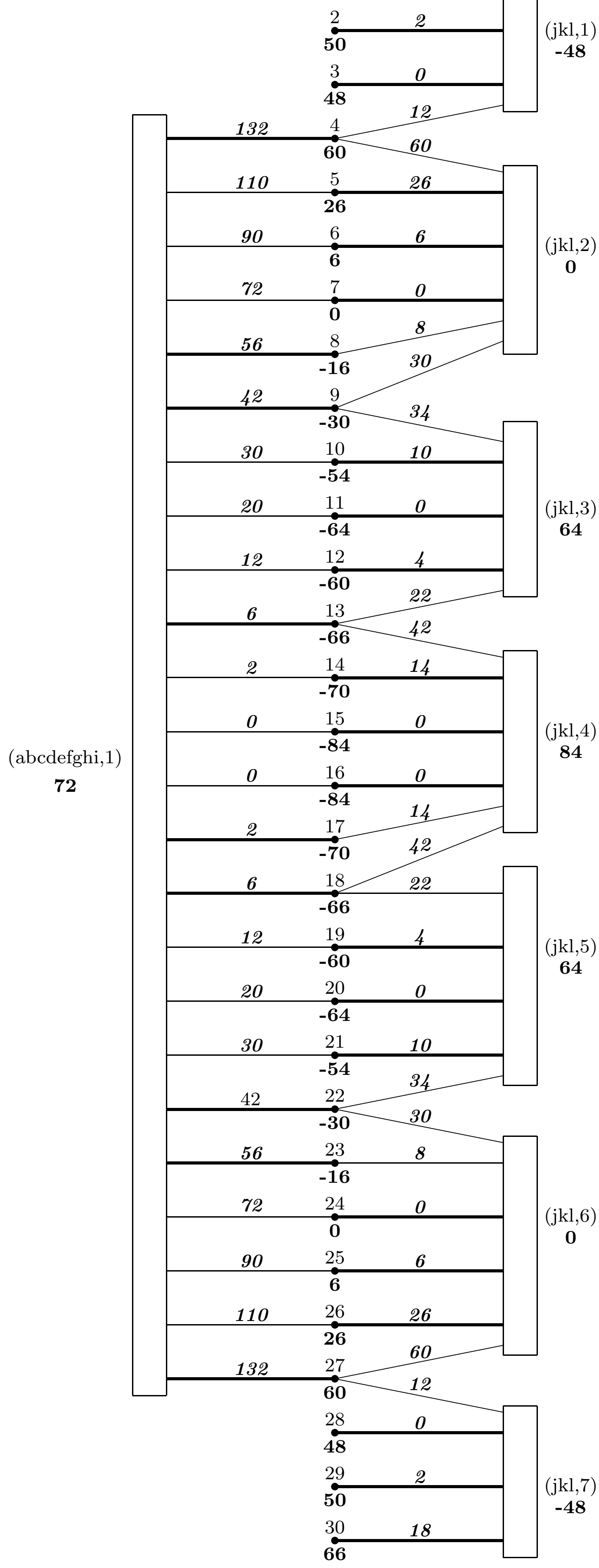

FIG. 10 – Bipartite graph for $d_a = d_b = ...d_i = 1$, $d_j = d_k = d_l = 7$ and weights for the min-sum-sqr problem

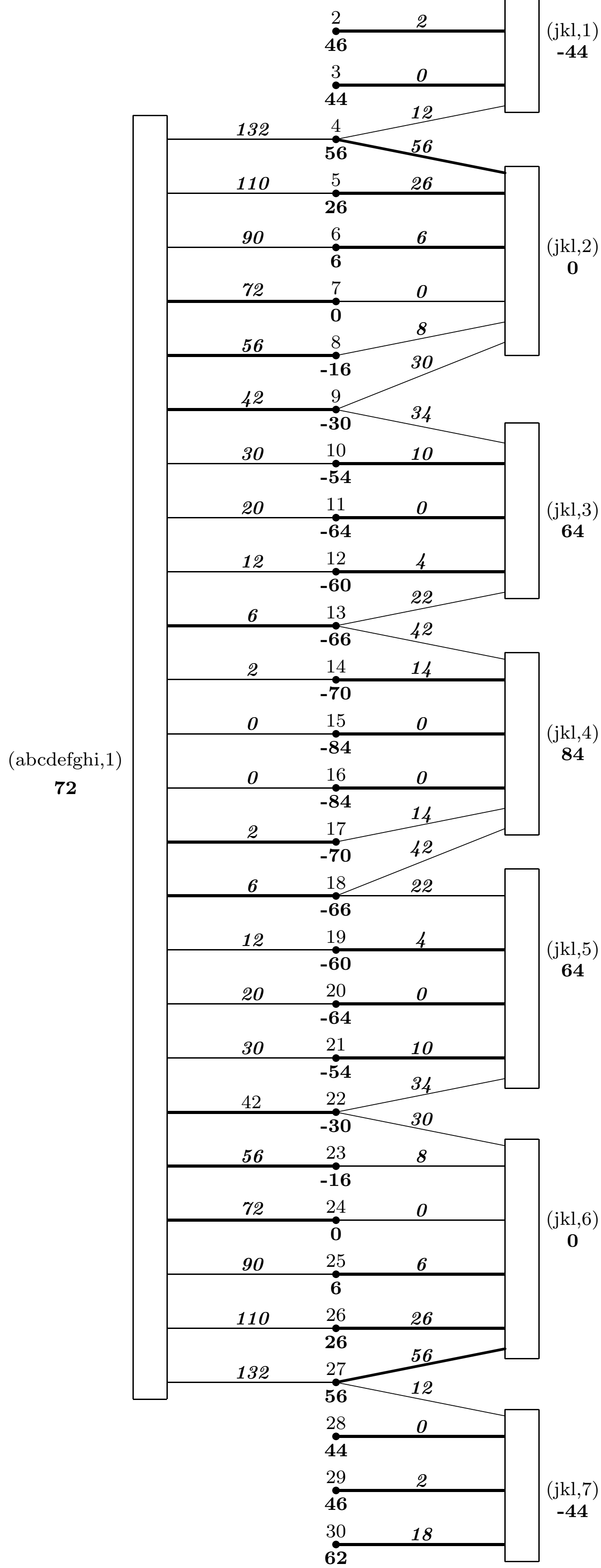

FIG. 11 – Bipartite graph for $d_a = d_b = ...d_i = 1$, $d_j = d_k = d_l = 7$ and weights for the min-sum-abs problem

the sets of optimal solutions of the sum-abs and sum-sqr problems are not identical. □

The sequence represented on Figure 11 has a maximum deviation higher than 1 and that of Figure 10 a maximum deviation lower than 1. As both are optimal solutions of the sum-sqr problem, one has the following statement :

Proposition 16 *There are instances such that an optimal solution of sum-sqr problem is in P but not all optimal solutions of sum-sqr problem are in P.*

8 Convex non-negative objective functions

When the max-abs and the sum-abs objective functions can be optimised simultaneously, the sum-sqr objective function is also optimised by the common solution of both problems. We have tested whether it was true for other objective functions.

Proposition 17 *There are instances such that the max-abs and the sum-abs objective functions can be optimised simultaneously, but not all pair convex functions can be optimised simultaneously to both objective functions.*

Proof
Consider the objective function $F_i(x) = x^4$. One can prove, using the method described in the proof of Proposition 11, that for $d = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 8, 8)$, sequences that simultaneously optimise max-abs and sum-abs are not optimal for this function. □

Remark Denote a, b, .. l the part types. The sequence $S = (k, l, a, k, l, b, c, k, l, d, k, l, e, f, k, l, g, k, l, h, i, k, l, j, k, l)$ optimises max-abs and $F_i(x) = x^4$. Our testing did not allow us to find an example where max-abs and sum-abs objective functions where simultaneously optimised and not max-abs and the min-sum objective function with $F_i(x) = x^4$. It is nonetheless likely than some exist.

9 Practical resolution

To find the examples of this document, we have used exhaustive testing of all instances for given values of the total demand D or all instances for given values of D and of the number of part types n. We have used the software CPLEX to solve the linear programs and we have adopted for them efficient formulations.

9.1 Resolution of total deviation problems

The CPLEX program has no difficulty to solve the total deviation problems as they need not to be handled as integer linear programs. In fact, the polyhedron (20) of the assignment constraints is integer.

$$
\begin{array}{rl}
P = & \{y; \\
 & \sum_{i=1}^{n} \sum_{j=1}^{d_i} y_{i,j,k} \; = \; 1, \quad \forall k = 1,\, 2, .. \, D \\
 & \sum_{k=1}^{D} y_{i,j,k} \; = \; 1, \qquad\qquad \forall i = 1,\, 2, .. \, n, \, \forall j = 1,\, 2, .. \, d_i\}
\end{array}
\tag{20}
$$

Therefore, in our programs, y was real most of the time and we obtained the optimal value of sum-sqr or sum-abs objective functions for the assignment problems, but not the optimal sequences.

9.2 Resolution of max-abs problem

The max-abs problem can be formulated as follow :

$$
\begin{array}{lll}
 & minimise \quad \max_{i,k} |x_{i,k} - kr_i| & \\
\text{st} & \sum_{i=1}^{n} x_{i,k} = k, & k = 1,\, 2, .. \, D \\
 & x_{i,D} = d_i, & i = 1,\, 2, .. \, n \\
 & 0 \leq x_{i,k} - x_{i,k-1}, & i = 1,\, 2, .. \, n; \; k = 2,\, 3, .. \, D \\
 & x_{i,k} \in \mathbb{N}, & i = 1,\, 2, .. \, n; \; k = 1,\, 2, .. \, D
\end{array}
\tag{21}
$$

This problem is non-linear but is easily transformed in the following integer linear program :

$$
\begin{array}{llll}
 & minimise \quad B & & \\
\text{st} & \sum_{i=1}^{n} x_{i,k} = k, & k = 1,\, 2, .. \, D & (22.a) \\
 & x_{i,D} = d_i, & i = 1,\, 2, .. \, n & (22.b) \\
 & 0 \leq x_{i,k} - x_{i,k-1}, & i = 1,\, 2, .. \, n; \; k = 2,\, 3, .. \, D & (22.c) \\
 & x_{i,k} - kr_i \leq B, & i = 1,\, 2, .. \, n; \; k = 1,\, 2, .. \, D & (22.d) \\
 & -x_{i,k} + kr_i \leq B, & i = 1,\, 2, .. \, n; \; k = 1,\, 2, .. \, D & (22.e) \\
 & x_{i,k} \in \mathbb{N}, & i = 1,\, 2, .. \, n; \; k = 1,\, 2, .. \, D & (22.f)
\end{array}
\tag{22}
$$

Inequality (22.d) and (22.e) state that B is greater than any deviation and as the objective is to minimise B, both problems are equivalent.

This formulation is not satisfactory since the polyhedron defined by the constraints (22.a) - (22.c) is not integer. CPLEX can solve it very efficiently most of the time. However, for some instances with many demands equal to 1, it may take several hours to obtain the solution using our computer. Therefore, we do not try and solve the max-abs problem directly, but use dichotomy and solve several max-abs decision problems as described in [11].

In fact, a max-abs decision problem can be described as a total deviation assignment problem. It consists in computing for the given B the intervals $[E(i,j) \mathinner{..} L(i,j)]$ for all $i = 1,\, 2,\, ..\, n$, $j = 1,\, 2,\, ..\, d_i$ and to consider the costs :

$$C_{i,j,k} = \begin{cases} 0 \text{ if } k \in [E(i,j) \mathinner{..} L(i,j)] \\ 1 \text{ otherwise} \end{cases} \tag{23}$$

The max-abs decision problem has a solution if and only if the min-sum problem with those costs has an optimal objective value equal to 0.

Using dichotomy, we find the minimum value of B such that the max-abs decision problem has a solution in few steps and gain a lot of time as each resolution takes less than 0.2 seconds.

9.3 Comparison of B-bounded and unbounded total deviation problems

The method described in Section 9.1 does not work for the B-bounded problems as the polytope (24) is not integer.

$$\begin{array}{rll} P_B = & \{y; & \\ & \sum_{i=1}^{n} \sum_{j=1}^{d_i} y_{i,j,k} \;=\; 1, & \forall k = 1,\, 2, ..\, D \\ & \sum_{k=1}^{D} y_{i,j,k} \;=\; 1, & \forall i = 1,\, 2, ..\, n,\, \forall j = 1,\, 2, ..\, d_i \\ & \sum_{p=1}^{k} \sum_{j=1}^{d_i} y_{i,j,p} - kr_i \leq B, & \forall i = 1,\, 2, ..\, n \\ & kr_i - \sum_{p=1}^{k} \sum_{j=1}^{d_i} y_{i,j,p} \leq B, & \forall i = 1,\, 2, ..\, n\} \end{array} \tag{24}$$

In this case, one has to impose y integer. The resolution is nonetheless very fast. This linear program allows to obtain the optimal objective value for B-bounded min-sum problems. When comparing objective values of unbounded and B-bounded problems, we do not need in a first time the effective optimal objective value of the B-bounded problem, but only to know if the optimal of the unbounded problem can be attained over P_B. In this case, it is possible to formulate the problem with y real. We first compute the optimal objective value of the unbounded problem as in Section 9.1. Denote S the optimal value of the min-sum objective function. To the polytope P of (20) we add the constraint $\sum_{(i,j,k)} C^s_{i,j,k} y_{i,j,k} = S$ with C^s the cost matrix for the min-sum problem. Denote P_S the polytope obtained. As an objective function we take a total deviation objective function with costs defined by (23). As previously, if the minimal objective value is 0, then there is a solution in P_S with maximum deviation lower than B. In this case, both problems have the same optimal solution. Otherwise, no element of P_B is optimal for the unbounded total deviation problem.

9.4 Generation of instances

In order to find counter-examples for the conjectures we wanted to tackle, we applied exhaustive search, i.e. we generated all instances for given values of D and tested if they verified the chosen statement. The recursive procedure *perm* generates all vectors of demands such that for all $i, j \in [1 \mathinner{..} n]$

$$i \leq j \Longrightarrow d_i \leq d_j \tag{25}$$

In addition, it produces the vectors of demands in the inverse lexicographic order. For example, for $D = 16$, it will produce the vectors $d = (16)$, $d = (8, 8)$, $d = (7, 9)$, $d = (6, 10)$, $d = (5, 11)$, $d = (5, 5, 6)$, etc. Since it saves memory not to give all vectors α and d the maximum size possible, we use vectors with dynamic memory allocation, i.e. vectors with a size null at their initialisation and that have a function *pushback(v,k)* which adds the element k at the end of the vector v and makes the memory allocation. Note that

- all vectors are integer.
- the p^{th} element of a vector v is denoted $v(p)$.
- the result of the division of integer a by the integer b is the quotient of the Euclidean division of a by b.

In the *perm* procedure, the vector d is the vector of demands of the instance that will be tested. To obtain it, we use the auxiliary vector α that contains a different formulation of vector d : for all $i \in [1 \mathinner{..} sizeof(\alpha)]$, $\alpha(i)$ represents the number of part types of demand equal to i. At the first call of the procedure, the vector α is empty and D is equal to the sum of all demands.

```
perm(int D, vector α) {
  If D = 0 Then
    vector d ;
    For i = 1, 2, .. sizeof(α) Do
      For j = 1, 2, .. α(i) Do
        pushback(d, i) ;
      End
    End
// One can use the instance generated by adding code here
    Return ;
  End

  If D < sizeof(α) + 1 Then
    Return ;
  End

  pushback(α, 0) ;
  For i = 1, 2, .. D/(sizeof(α) + 1) Do
    α(sizeof(α)) = i ;
    perm(D − i ∗ sizeof(α), α) ;
```

End
}

Example
Consider we want to obtain all vectors with total demand $D = 4$. The tree of calls is represented on Figure 12. The vertices are named after the parameters D and α and the edges represent the calls made for those parameters. The sign $\varnothing$ appears under the name of a vertex D, α when no instance is produced by $perm(D, \alpha)$. If an instance is produced, the demands are represented together with D and α. □

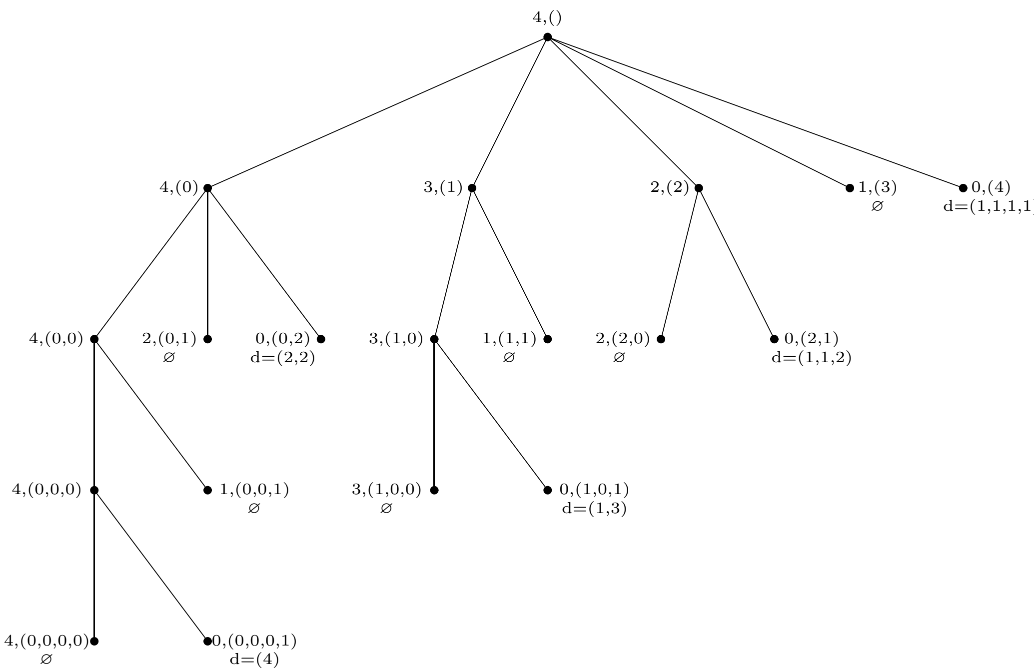

FIG. 12 – Parameters for the calls of *perm* for $D = 4$

10 Conclusion and extensions

Three objective functions for the JIT scheduling problem have been examined. Existing tools to prove the optimality of total deviation solutions have also been completed. Extensive testing has allowed us to conclude regarding several questions about total deviation objective functions and maximum deviation bounded problems as well as to refute several recent conjectures. We have proved that the 1-bounded sum-abs and the 1-bounded sum-sqr

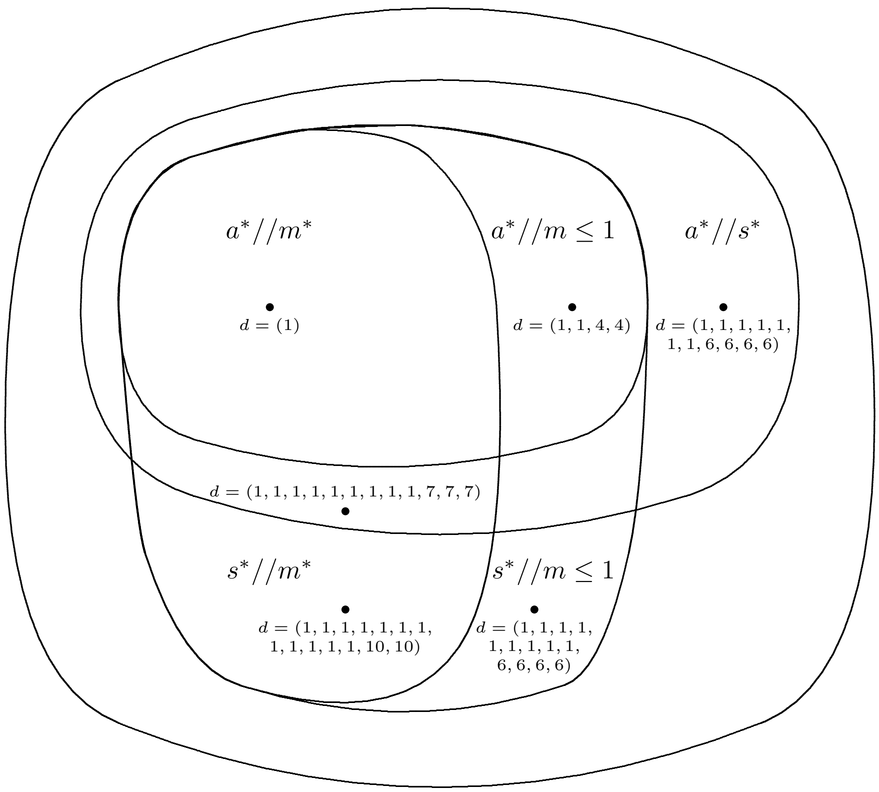

FIG 13: Instances optimising simultaneously different criteria

problems were equivalent but could have no common optimal solutions with the sum-abs and the sum-sqr problems.

Figure 13 is a summary of the results obtained for sum-abs, sum-sqr and max-abs objective functions. The notations a^*, s^* and m^* indicate that the sum-abs, sum-sqr and max-abs objective functions respectively are optimised by the instances of the set where it appears. The symbol // means that the criteria of each side are optimised simultaneously. When $m \leq 1$ appears in the right-hand-side, sequences can be found with maximum deviation lower than 1 that optimise the left-hand-side objective function.

Still, our testing has not allowed us to conclude regarding the following question :

Conjecture 3 *For* $n = 3$ *there are instances such that the sum-abs problem has no optimal solution such that the maximum deviation is lower than* 1.

This conjecture has been tested for all instances with $D \leq 100$ and no example has been found.

Further researches can be undertaken regarding the set of all solutions of the different problems. We proved that sum-abs and sum-sqr problems are not always equivalent when they have a common optimal solution but we have no result concerning 1-bounded sum-abs problems. It is likely that if an optimal solution of the sum-abs problem has maximum deviation lower than 1, it will not imply that all sum-abs optimal solution have maximum deviation lower or equal to 1, but this question has not been studied here. Since sum-abs and sum-sqr problems can have no 1-bounded solution, computational improvement proposed by Steiner and Yeomans in [12] cannot be applied to the general case, but studying 1-bounded solutions may reveal that they can approximate the solutions of the unbounded problems with a low ration.

Acknowlegments

The authors thank Van Dat Cung for his help regarding the tests described in this document.

Références

[1] A. Asratian, T.M.J. Denley, and R. Hắggkvist. *Bipartite graphs and their applications*. University Press, Cambridge, 1998.

[2] N. Brauner and Y. Crama. The maximum deviation just-in-time scheduling problem. *Discrete Applied Mathematics*, 134 :25–50, 2004.

[3] R.R. Inman and R.L. Bulfin. Sequencing JIT mixed-model assembly lines. *Management Science*, 37 :901–904, 1991.

[4] V. Jost. *Deux problèmes d'approximation diophantienne : Le partage proportinnel en nombre entiers et Les pavages équilibrés de* $\mathbb{Z}$. Mémoire de DEA en Recherche Operationnelle, INPG, Grenoble, France, 2002.

[5] M. Y. Kovalyov, W. Kubiak, and J. S. Yeomans. A computational analysis of balanced JIT optimization algorithms. *Journal of Information Systems and Operational Research - INFOR*, 39 :299–315, 2001.

[6] W. Kubiak. Minimizing variation of production rates in just-in-time systems : a survey. *European Journal of Operational Research*, 66 :259–271, 1993.

[7] W. Kubiak and S. Sethi. Optimal just-in-time schedules for flexible transfer lines. *The International Journal of Flexible Manufacturing Systems*, 6 :137–154, 1994.

[8] L. Lovasz and M.D. Plummer. *Matching theory*. Akademiai Kiado, Budapest, 1986.

[9] J. Miltenburg. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, 35 :192–207, 1989.

[10] Y. Monden. *Toyota production system*. Institute of Industrial Engineers Press, 1983.

[11] G. Steiner and J. S. Yeomans. Level schedules for mixed-model, just-in-time processes. *Management Science*, 39 :728–735, 1993.

[12] G. Steiner and J. S. Yeomans. A bicriterion objective for levelling the schedule of a mixed-model, just-in-time assembly processes. *Mathematical and Computer Modelling*, 20 :123–134, 1994.

Les Cahiers du laboratoire Leibniz

Le **Laboratoire Leibniz** est fortement pluridisciplinaire. Son activité scientifique couvre un large domaine qui comprend des thèmes fondamentaux aussi bien en informatique qu'en mathématiques, avec une ouverture sur l'apprentissage machine, la modélisation de systèmes complexes adaptatifs, et les applications aux environnements informatiques pour l'apprentissage humain.

Les **Cahiers du laboratoire Leibniz** ont pour vocation la diffusion de rapports de recherche, de supports de cours, de textes de séminaires ou de projets de publications réalisés par des membres du laboratoire. Ils peuvent accueillir aussi des textes de chercheurs n'appartenant pas au laboratoire Leibniz mais qui travaillent sur des thèmes proches et ne disposent pas de tels supports de publication. Ces chercheurs sont priés de contacter un des membres du comité éditorial ; le comité décidera de l'acceptation du texte proposé.

Le contenu des textes publiés dans les **Cahiers du laboratoire Leibniz** relève de la seule responsabilité de leurs auteurs.

The research at **Laboratoire Leibniz** is multidisciplinary. It covers a large domain of fundamental and applied subjects in informatics an mathematics, with openings to machine learning, the modelisation of adaptive complex systems, and applications to teaching software.

The **Cahiers du laboratoire Leibniz** aim at diffusing research reports, lectures, and texts of conferences or pre-prints of the members of the laboratory. Moreover, the **Cahiers** welcome manuscripts of researchers belonging to other laboratories, working on subjects close to ours, but not disposing of such a medium. These researchers should contact one of the members of the editorial board; the latter will decide whether to accept the proposal.

The responsibility of the contents of the **Cahiers** lies exclusively with the authors.

Directeur de la publication

Nicolas Balacheff

Réalisation : Jacky Coutin

ISSN : 1298-020X - © laboratoire Leibniz