On a Parallel Machine Scheduling Problem with Equal Processing Times

Svetlana A. Kravchenko [1]
United Institute of Informatics Problems,
Surganova St. 6, 220012 Minsk, Belarus
kravch@newman.bas-net.by

Frank Werner
Otto-von-Guericke-Universität, Fakultät für Mathematik,
39106 Magdeburg, Germany
frank.werner@mathematik.uni-magdeburg.de

August 24, 2007

Abstract

In this paper, we give a polynomial algorithm for problem $P \mid r_j, p_j = p \mid \sum f_j(C_j)$, where f_j is any non-decreasing function such that for any indices i and j, function $f_i - f_j$ is monotonous, and a polynomial algorithm for problem $P \mid r_j, p_j = p, D_j \mid \max \varphi_j(C_j)$, where φ_j is any non-decreasing function for any j.

Keywords: parallel machine scheduling, linear programming

MSC classification: 90B35

1 Introduction

The problem considered can be stated as follows. There are n jobs $J_1, \ldots, J_n$ which have to be processed on m identical parallel machines. For each job J_j, $j = 1, \ldots, n$, a processing time $p_j = p$, which is equal for all the jobs, and a release date r_j are given in advance. In

[1]Supported by the Alexander von Humboldt Foundation

dependence on the problem, each job can be characterized by a due date d_j, a weight w_j, and by a deadline D_j. Each machine can process only one job at a time. Preemption of processing is not allowed, i.e. the processing of any job started at time t on one of the identical machines will be completed at time $t+p$ on the same machine. The aim is to find a schedule such that the criterion function takes its minimal value.

In Section 3, we give a polynomial algorithm for problem $P \mid r_j, p_j = p \mid \sum f_j(C_j)$, where $f_j(C_j)$ is the cost associated with job J_j completed at time C_j. Here we suppose that f_j is any non-decreasing function such that for any indices i and j, function $f_i - f_j$ is monotonous. Thus, we prove that parallel machine scheduling problems with release dates and equal processing times can be polynomially solved even for such complex criteria as $\sum w_j C_j^2$ or $\sum e^{w_j C_j}$.

In Section 5, we give a polynomial algorithm for problem $P \mid r_j, p_j = p, D_j \mid \max \varphi_j(C_j)$, where $\varphi_j(C_j)$ is the cost associated with job J_j completed at time C_j. The classical scheduling criteria belonging to $\max \varphi_j(C_j)$ are the minimization of maximum lateness $L_{max} = \max_j\{C_j - d_j\}$ and maximum tardiness $\max T_j = \max_j\{L_j, 0\}$.

Known polynomially solvable problems related to the subject of this paper are

$P \mid r_j, p_j = p, D_j \mid C_{max}, \sum C_j$	-	Simons [6]
$P \mid r_j, p_j = p \mid \sum w_j C_j$	-	Brucker & Kravchenko [3]
$P \mid r_j, p_j = p \mid \sum T_j$	-	Brucker & Kravchenko [4]
$Pm \mid r_j, p_j = p \mid \sum f_j(C_j)$	-	Baptiste [1]
$P \mid r_j, p_j = p \mid L_{max}$	-	Baptiste & Brucker [2].

Here $C_{\max} = \max\{C_1, \ldots, C_n\}$ denotes the makespan, and $\sum T_j = \sum_j \max\{0, C_j - d_j\}$ denotes the total tardiness.

This paper presents a generalization of the approach given in [1], [3] and [4]. In the next section, we summarize some basic properties of the problem types considered in this paper, and we present two linear programming (LP) relaxations.

2 Basic properties and LP formulation

An optimal schedule for problem $P \mid r_j, p_j = p, D_j \mid F$, where F is any non-decreasing function, can be found in the class of schedules, where each job is processed in one of the following intervals of length p :

$$\Big\{[r_j + kp, r_j + kp + p \ [\ \mid k \in \{0, 1, 2, \ldots\}\Big\}.$$

The proof of this fact is analogous to that in [3]. Suppose that in an optimal schedule, there is a job processed in the interval $[e, e+p[$ that is not from the above set $\{[r_j + kp, r_j + kp + p \ [\ \mid k \in \{0, 1, 2, \ldots\}\}$. Find the longest chain of intervals $[a_1, a_2[, [a_2, a_3[, \ldots, [a_{k-1}, a_k[$ such that $[e, e+p[$ belong to $\{[a_1, a_2[, [a_2, a_3[, \ldots, [a_{k-1}, a_k[\}$. In this case, no point $\{a_1, \ldots, a_k\}$ is from the set $\{r_1, \ldots, r_n\}$. Denote by J_1 a job processed in $[a_1, a_2[$. Since F is assumed to be a non-decreasing function, one can shift J_1 to the left. Therefore, we can find an optimal

schedule in the class of schedules, where each job is processed in an interval from the set $\{[r_j + kp, r_j + kp + p \; [\; | \; k \in \{0, 1, 2, \ldots\}\}$.

On the other hand, if for some r_k and for any index j the inequality $r_j + \sum_{r_j \leq r_i < r_k} p_i \leq r_k$ holds, then the problem can be decomposed into subproblems that can be solved independently from each other. Therefore, without loss of generality, we can suppose that $\min_j\{r_j\} + 2np \geq \max_j\{D_j\}$ holds. Thus, we will restrict the possible positions of the jobs by the set of intervals

$$\Big\{[r_j + kp, r_j + kp + p \; [\; | \; k \in \{\ldots, -1, 0, 1, \ldots\}, r_j + kp \geq \min_i\{r_i\},$$

$$r_j + kp + p \leq \max_i\{D_i\}\Big\}.$$

Take all the different intervals from this set and enumerate them in increasing order of their left endpoints. Denote the obtained set by $\{I_i \mid i \in \{1, \ldots, z\}\}$ and for each I_i, denote by $D(I_i)$ the right endpoint of I_i and by $R(I_i)$ the left endpoint of I_i. One can see that there exists some q such that $I_{i+1} \cap \ldots \cap I_{i+q} \neq \emptyset$ for any $i \in \{0, \ldots, z - q\}$ and therefore, if m jobs are processed in I_{i+k}, then no job can be processed in the other intervals $I_{i+1}, \ldots, I_{i+k-1}, I_{i+k+1}, \ldots, I_{i+q}$. Set $y = \max\{q \mid I_{i+1} \cap \ldots \cap I_{i+q} \neq \emptyset, i \in \{0, \ldots, z - q\}\}$, i.e. $y = \mid \{r_j + kp \mid k \in \mathbb{Z}, r_j + kp \in I_1\} \mid$.

For problem $P \mid r_j, p_j = p \mid \sum f_j(C_j)$ consider the following LP relaxation:

$$\text{minimize} \sum_{i=1}^{z} \sum_{j=1}^{n} f_j(D(I_i)) x_{ji} \tag{2.1}$$

subject to

$$\sum_{i=1}^{z} x_{ji} = p, \quad j = 1, \ldots, n \tag{2.2}$$

$$\sum_{j=1}^{n} x_{j,i+1} + \sum_{j=1}^{n} x_{j,i+2} + \ldots + \sum_{j=1}^{n} x_{j,i+y} \leq mp, \quad i = 0, \ldots, z - y \tag{2.3}$$

$$x_{ji} = 0 \text{ if } R(I_i) < r_j \quad i = 1, \ldots, z, \quad j = 1, \ldots, n \tag{2.4}$$

$$x_{ji} \geq 0, \quad i = 1, \ldots, z, \quad j = 1, \ldots, n \tag{2.5}$$

If we set x_{ji} equal to the amount of job J_j processed in the interval I_i, then any feasible schedule for problem $P \mid r_j, p_j = p \mid \sum f_j(C_j)$ can be described as a feasible solution of problem (2.1) – (2.5). On the other hand, if the obtained solution to problem (2.1) – (2.5) is integer, then it is a solution of problem $P \mid r_j, p_j = p \mid \sum f_j(C_j)$.

Further we will suppose that problem (2.1) – (2.5) has been solved and that the obtained solution $x^* = \mu$ is not integer. Parallel to the notation x_{ji}, we will use $x_{j,i}$ to avoid a confusion between the first index and the second one.

For problem $P \mid r_j, p_j = p, D_j \mid \max f_j(C_j) \leq F$, where $F = f_j(D(I_i))$ for some $i \in \{1, \ldots, z\}$, and $j \in \{1, \ldots, n\}$, consider the following feasibility problem:

$$\sum_{i=1}^{z} x_{ji} = p, \quad j = 1, \ldots, n \tag{2.6}$$

$$\sum_{j=1}^{n} x_{j,i+1} + \sum_{j=1}^{n} x_{j,i+2} + \ldots + \sum_{j=1}^{n} x_{j,i+y} \leq mp, \quad i = 0, \ldots, z - y \tag{2.7}$$

$$x_{ji} = 0 \text{ if } R(I_i) < r_j \quad i = 1, \ldots, z, \quad j = 1, \ldots, n \tag{2.8}$$

$$x_{ji} = 0 \text{ if } f_j(D(I_i)) > F \tag{2.9}$$

$$x_{ji} \geq 0, \quad i = 1, \ldots, z, \quad j = 1, \ldots, n \tag{2.10}$$

If we set x_{ji} equal to the amount of job J_j processed in the interval I_i, then any feasible schedule for problem $P \mid r_j, p_j = p, D_j \mid \max f_j(C_j) \leq F$ can be described as a feasible solution of problem (2.6) – (2.10). On the other hand, if the obtained solution of problem (2.6) – (2.10) is integer, then it is a solution of problem $P \mid r_j, p_j = p, D_j \mid \max f_j(C_j) \leq F$. Further we will suppose that problem (2.6) – (2.10) has been solved and the obtained solution $x^* = \chi$ is not integer.

Moreover, suppose that neither the solution μ nor solution χ contain mixed jobs, since in any case one can avoid mixed jobs by an apropriate transformation. By mixed jobs we mean two jobs, say J_y and J_z, such that there exists a set of intervals I_a, I_b, I_c, I_d with $a < b < c < d$ or with $a = b < c \leq d$ or with $a < b < c = d$ (the last case is considered only for problem $P \mid r_j, p_j = p, D_j \mid \max f_j(C_j) \leq F$) and $x^*_{ya} \neq 0$, $x^*_{yc} \neq 0$, $x^*_{zb} \neq 0$, $x^*_{zd} \neq 0$. Without loss of generality, we suppose that $x^*_{ya} = x^*_{yc} = x^*_{zb} = x^*_{zd} = \delta$.

- If $a < b < c < d$, then by an appropriate transformation we can avoid mixed jobs in the considered solutions. A solution χ can be transformed in the following way: we set $\chi_{yc} = 0$, $\chi_{yd} = \delta$, $\chi_{zd} = 0$, $\chi_{zc} = \delta$ and the other values of χ are not changed.

  For solution μ, if $f_z - f_y$ is non-decreasing, we set $\mu_{yc} = 0$, $\mu_{yd} = \delta$, $\mu_{zd} = 0$, $\mu_{zc} = \delta$ and all other values of μ are not changed. In this case, the objective function value will decrease by $\delta\Big(f_z(D(I_c)) - f_y(D(I_c))\Big) - \delta\Big(f_z(D(I_d)) - f_y(D(I_d))\Big)$. However, if $f_z - f_y$ is non-increasing, see Figure 1(a), we set $\mu_{yc} = 0$, $\mu_{yb} = \delta$, $\mu_{zb} = 0$, $\mu_{zc} = \delta$. In this case, the objective function will decrease by $\delta\Big(f_z(D(I_c)) - f_y(D(I_c))\Big) - \delta\Big(f_z(D(I_b)) - f_y(D(I_b))\Big)$.

- If $a = b < c < d$, we set $\chi_{zb} = 0$, $\chi_{zc} = \delta$, $\chi_{yc} = 0$, $\chi_{yb} = \delta$ and the other values of χ are not changed.

  For solution μ, if $f_z - f_y$ is non-decreasing, we set $\mu_{ya} = 0$, $\mu_{yd} = \delta$, $\mu_{zd} = 0$, $\mu_{za} = \delta$ and all other values of μ are not changed. In this case, the objective function value will decrease by $\delta\Big(f_z(D(I_a)) - f_y(D(I_a))\Big) - \delta\Big(f_z(D(I_d)) - f_y(D(I_d))\Big)$. However, if $f_z - f_y$ is non-increasing, we set $\mu_{yc} = 0$, $\mu_{yb} = \delta$, $\mu_{zb} = 0$, $\mu_{zc} = \delta$. In this case the objective function will decrease by $\delta\Big(f_z(D(I_c)) - f_y(D(I_c))\Big) - \delta\Big(f_z(D(I_b)) - f_y(D(I_b))\Big)$.

- If $a < b < c = d$, we set $\chi_{zb} = 0$, $\chi_{zc} = \delta$, $\chi_{yc} = 0$, $\chi_{yb} = \delta$ and the other values of χ are not changed.

In all cases, we obtain a feasible solution after the transformation. Thus, we can suppose that the considered solutions x^* do not contain mixed jobs.

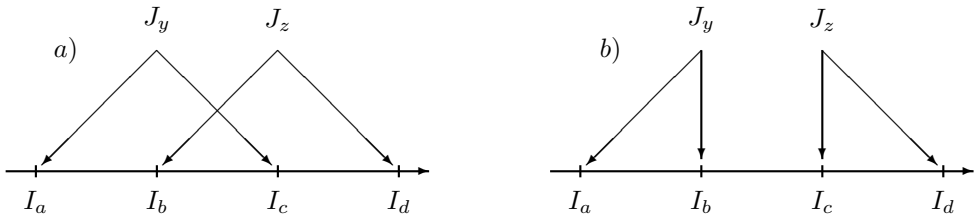

Figure 1: Transformation for the case when $f_z - f_y$ is non-increasing

3 Algorithm

In this section, we present a polynomial algorithm for the two types of scheduling problems under consideration. Using the optimal solution x^* of problem (2.1) – (2.5) or a feasible schedule of problem (2.6) – (2.10), we determine all intervals where jobs are processed in an optimal (feasible) schedule. With this purpose, we calculate the value $v(I_k)$ for each interval I_k as follows:

$$v(I_k) = \sum_{i=1}^{k} \sum_{j=1}^{n} x^*_{ji}, \qquad k = 1, \ldots, z.$$

Note that $v(I_z) = \sum_{i=1}^{z} \sum_{j=1}^{n} x^*_{ji} = np$ holds if the solution is feasible.

Since any interval can be occupied by m jobs, we define the values:

$$v_1 = v(I_1), \ldots, v_m = v(I_1),$$

$$v_{m+1} = v(I_2), \ldots, v_{2m} = v(I_2),$$

$$\ldots$$

$$v_{(z-1)m+1} = v(I_z), \ldots, v_{zm} = v(I_z),$$

i.e. we take m copies of each interval enumerated in nondecreasing order of their left endpoints.

Now, using $v_1, \ldots, v_{zm}$, we mark all intervals where some jobs are processed in an optimal (feasible) schedule. The marking procedure is as follows:

Step 1: To select the first marked interval, one moves from v_1 to v_{zm} and takes the first v_j such that $v_j > 0 \cdot p$ holds. Denote the corresponding interval by I_1^*, i.e. $v_j = v(I_1^*)$.

Step 2: To select the second marked interval, one moves from $v_j = v(I_1^*)$ to v_{zm} and takes the first v_g such that $v_g > 1 \cdot p$ holds. Denote the corresponding interval by I_2^*, i.e. $v_g = v(I_2^*)$.

...

Step n: To select the n-th marked interval, one moves from $v(I_{n-1}^*)$ to v_{zm} and takes the first v_h such that $v_h > (n-1) \cdot p$ holds. Denote the corresponding interval by I_n^*, i.e. $v_h = v(I_n^*)$.

The marked intervals determine the places where the jobs are processed in an optimal (feasible) schedule.

Consider the bipartite graph $G = (V, E)$, where $V = \{J_1, \ldots, J_n\} \cup \{I_1^*, \ldots, I_n^*\}$, $E = \{(J_j, I_i^*) \mid x_{ji}^* \neq 0\}$. Any perfect matching of G corresponds to an integer solution of problem (2.1) – (2.5) and problem (2.6) – (2.10), respectively, and therefore to an optimal (feasible) schedule for problem $P \mid r_j, p_j = p \mid \sum f_j(C_j)$ and $P \mid r_j, p_j = p, D_j \mid \sum \max f_j(C_j) \leq F$, respectively.

4 Proof of optimality/feasibility

In the section, we prove that the algorithm presented in Section 3 generates an optimal (feasible) schedule for problems $P \mid r_j, p_j = p \mid \sum f_j(C_j)$ and $P \mid r_j, p_j = p, D_j \mid \sum \max f_j(C_j) \leq F$.

Lemma 1 *Graph G contains a perfect matching.*

Proof: To prove this lemma, we have to separate all x_{ji}^*, i.e. if $x_{ai}^* \neq 0$, $x_{bi}^* \neq 0$, then we will consider in the interval I_i a subinterval I_{ai} with the length x_{ai}^* and a subinterval I_{bi} with the length x_{bi}^*.

Using x^*, we define for each job J_j its starting time which is equal to $\min\{R(I_i) \mid x_{ji}^* \neq 0\}$. Since there are no mixed jobs, the maximal number of jobs with the same starting time is equal to the number of machines. Now, for any interval I_i, we consider all $x_{ji}^* \neq 0$ and associate the subinterval I_{ji} of the length x_{ji}^* with each job J_j. All subintervals I_{ji} are contained in the interval I_i in increasing order of the starting time of the parts of the corresponding job J_j. Note that the marking procedure can be made for the subintervals since each of them appears with some positive length, i.e. if the interval I_i contains the subintervals $I_{1,i}$, $I_{2,i}$, ..., $I_{k,i}$ in the given order, then we will count $v(I_i) = \sum_{g=1}^{i} \sum_{j=1}^{n} x_{jg}^* = v(I_{1,i}) + v(I_{2,i}) + \ldots + v(I_{k,i})$, where $v(I_{1,i}) = v(I_{i-1}) + x_{1,i}^*$, ..., $v(I_{k,i}) = v(I_{i-1}) + x_{1,i}^* + \ldots + x_{k,i}^*$. If we apply the marking procedure to the set of subintervals, then each marked interval will contain one marked subinterval. Thus, for each marked interval I_i^* we denote the marked subinterval by $I_{\gamma(i),i}$.

Now, since each marked subinterval $I_{\gamma(i),i}$ corresponds to some job $J_{\gamma(i)}$ such that $x_{\gamma(i),i}^* \neq 0$, then to prove the lemma, it suffices to show that each job corresponds to one marked subinterval only. Suppose that J_a corresponds to two marked subintervals $I_{\gamma(i),i} = I_{a,i}$ and $I_{\gamma(j),j} = I_{a,j}$. Let $I_{a,i}$ be $(k+1)$-th marked subinterval and suppose that $x_{a,i}^* = (\epsilon+\delta)p$, $x_{a,j}^* = \beta p$, $v(I_{a,i}) = kp+\epsilon p$, where $\epsilon + \delta < 1$. Then $v(I_{a,i}) - x_{a,i}^* = kp - \delta p$, where $\epsilon + \delta < 1$. Assume that $z-1$ jobs be processed between the subintervals $I_{a,i}$ and $I_{a,j}$. Then $v(I_{a,j}) - x_{a,j}^* = kp + (z-1)p + \epsilon p$, and

therefore, $I_{a,j}$ is the $(k+z+1)$-th marked interval and $v(I_{a,j}) = kp+(z-1)p+\epsilon p+\beta p > (k+z)p$ holds. However, since $x_{a,i} + x_{a,j} = (\epsilon + \delta + \beta)p < p$, we have $\epsilon + \beta < 1$, and therefore job J_a cannot correspond to two marked subintervals. □

Note that a perfect matching corresponds to an integer solution. Denote this solution by $\tilde{x}$. For problem $P \mid r_j, p_j = p, D_j \mid \max f_j(C_j) \leq F$, $\tilde{x}$ is a feasible solution. Since $F = f_j(D(I_i))$ holds for some $i \in \{1, \ldots, z\}$ and $j \in \{1, \ldots, n\}$, and the number of different values $F_j(D(I_i))$ is polynomially bounded, we will obtain an optimal solution for problem $P \mid r_j, p_j = p, D_j \mid \max f_j(C_j)$ in a polynomial number of steps.

Lemma 2 *$\tilde{x}$ is an optimal solution for problem (2.1) – (2.5).*

Proof: To prove that $\tilde{x}$ is an optimal solution, we have to prove that

$$\sum_{i=1}^{z}\sum_{j=1}^{n} f_j(D(I_i))x^*_{ji} = \sum_{i=1}^{z}\sum_{j=1}^{n} f_j(D(I_i))\tilde{x}_{ji}.$$

Suppose that

$$\sum_{i=1}^{z}\sum_{j=1}^{n} f_j(D(I_i))\tilde{x}_{ji} > \sum_{i=1}^{z}\sum_{j=1}^{n} f_j(D(I_i))x^*_{ji}.$$

Consider the vector $(1+\epsilon)x^*-\epsilon\tilde{x}$. Now we show that one can find an $\epsilon > 0$ such that $(1+\epsilon)x^*-\epsilon\tilde{x}$ is a feasible solution of problem (2.2) – (2.5).

Since $\tilde{x}_{ji} \neq 0$ implies $x^*_{ji} \neq 0$, we take $\epsilon_1 = \min\{x^*_{ji} \mid \tilde{x}_{ji} \neq 0\}$. Then $(1 + \epsilon_1)x^* - \epsilon_1\tilde{x}$ satisfies (2.2), (2.4), and (2.5). Now consider (2.3).

If

$$\sum_{j=1}^{n} x^*_{j,i+1} + \sum_{j=1}^{n} x^*_{j,i+2} + \ldots + \sum_{j=1}^{n} x^*_{j,i+y} \leq \sum_{j=1}^{n} \tilde{x}_{j,i+1} + \sum_{j=1}^{n} \tilde{x}_{j,i+2} + \ldots + \sum_{j=1}^{n} \tilde{x}_{j,i+y},$$

then

$$\sum_{j=1}^{n}((1+\epsilon)x^*_{j,i+1} - \epsilon\tilde{x}_{j,i+1}) + \sum_{j=1}^{n}((1+\epsilon)x^*_{j,i+2} - \epsilon\tilde{x}_{j,i+2}) + \ldots + \sum_{j=1}^{n}((1+\epsilon)x^*_{j,i+y} - \epsilon\tilde{x}_{j,i+y}) \leq mp$$

holds for any ϵ.

If

$$mp > \sum_{j=1}^{n} x^*_{j,i+1} + \sum_{j=1}^{n} x^*_{j,i+2} + \ldots + \sum_{j=1}^{n} x^*_{j,i+y} > \sum_{j=1}^{n} \tilde{x}_{j,i+1} + \sum_{j=1}^{n} \tilde{x}_{j,i+2} + \ldots + \sum_{j=1}^{n} \tilde{x}_{j,i+y},$$

then we take ϵ_2 such that

$$\sum_{j=1}^{n}(1+\epsilon_2)x^*_{j,i+1} + \sum_{j=1}^{n}(1+\epsilon_2)x^*_{j,i+2} + \ldots + \sum_{j=1}^{n}(1+\epsilon_2)x^*_{j,i+y} = mp.$$

In this case, we get

$$(1+\epsilon_2)\Big(\sum_{j=1}^{n} x^*_{j,i+1}+\sum_{j=1}^{n} x^*_{j,i+2}+\ldots+\sum_{j=1}^{n} x^*_{j,i+y}\Big)-\epsilon_2\Big(\sum_{j=1}^{n} \tilde{x}_{j,i+1}+\sum_{j=1}^{n} \tilde{x}_{j,i+2}+\ldots+\sum_{j=1}^{n} \tilde{x}_{j,i+y}\Big) < mp.$$

Finally, if

$$mp = \sum_{j=1}^{n} x^*_{j,i+1} + \sum_{j=1}^{n} x^*_{j,i+2} + \ldots + \sum_{j=1}^{n} x^*_{j,i+y},$$

then

$$mp = \sum_{j=1}^{n} \tilde{x}_{j,i+1} + \sum_{j=1}^{n} \tilde{x}_{j,i+2} + \ldots + \sum_{j=1}^{n} \tilde{x}_{j,i+y}$$

holds since in case of $v(I_{i+y}) = v(I_i) + mp$, there are m marked intervals in the set I_{i+1}, ..., I_{i+y}.

Thus, inequality (2.3) can also be satisfied by an appropriate selection of ϵ. So we obtain that for some ϵ, the vector $(1+\epsilon)x^* - \epsilon\tilde{x}$ is a feasible solution of problem (2.2) – (2.5). However, in this case

$$\sum_{i=1}^{z}\sum_{j=1}^{n} f_j(D(I_i))\Big((1+\epsilon)x^*_{ji} - \epsilon\tilde{x}_{ji}\Big) =$$

$$\sum_{i=1}^{z}\sum_{j=1}^{n} f_j(D(I_i))x^*_{ji} - \epsilon\Big(\sum_{i=1}^{z}\sum_{j=1}^{n} \tilde{f}_j(D(I_i))x_{ji} - f_j(D(I_i))x^*_{ji}\Big) <$$

$$\sum_{i=1}^{z}\sum_{j=1}^{n} f_j(D(I_i))x^*_{ji}$$

i.e. x^* is not an optimal solution of the LP problem, which is a contradiction to the assumption.

□

5 Example

Consider the following instance of problem $P \mid r_j, p_j = 2 \mid \sum T_j$. There are two machines and four jobs with $r_1 = r_2 = 0$, $r_3 = 1$, $r_4 = 2$, $d_1 = 2$, $d_2 = 4$, $d_3 = 3$, $d_4 = 4$. The possible places of the jobs are described by the following four intervals $I_1 = [0, 2[$, $I_2 = [1, 3[$, $I_3 = [2, 4[$, $I_4 = [3, 5[$.

Then the corresponding LP formulation is as follows:

$$\text{minimize } x_{12} + 2x_{13} + 3x_{14} + x_{24} + x_{33} + 2x_{34} + x_{44}$$

subject to

$$
\begin{array}{rcrcrcrcrcrcrcr}
x_{11} & + & x_{12} & + & x_{13} & + & x_{14} & = & 2 & & & & & & \\
x_{21} & + & x_{22} & + & x_{23} & + & x_{24} & = & 2 & & & & & & \\
 & & x_{32} & + & x_{33} & + & x_{34} & = & 2 & & & & & & \\
 & & & & x_{43} & + & x_{44} & = & 2 & & & & & & \\
x_{11} & + & x_{12} & + & x_{21} & + & x_{22} & + & x_{32} & \leq & 4 & & & & \\
x_{12} & + & x_{22} & + & x_{32} & + & x_{13} & + & x_{23} & + & x_{33} & + & x_{43} & \leq & 4 \\
x_{13} & + & x_{23} & + & x_{33} & + & x_{43} & + & x_{14} & + & x_{24} & + & x_{34} & + & x_{44} & \leq & 4
\end{array}
$$

$$x_{31} = x_{41} = x_{42} = 0$$

$$x_{ij} \geq 0 \text{ for } i, j = 1, \ldots, 4$$

The optimal objective function value for this problem is 2. One of the possible optimal solutions is $x_{11} = 2$, $x_{21} = 1$, $x_{24} = 1$, $x_{32} = 1$, $x_{33} = 1$, $x_{43} = 2$. In this case, we get $v(I_1) = 3$, $v(I_2) = 4$, $v(I_3) = 7$, $v(I_4) = 8$, and both copies of the intervals I_1 and I_3 are marked. Then a perfect matching and an optimal solution is $x_{11} = 2$, $x_{21} = 2$, $x_{33} = 2$, and $x_{43} = 2$.

ACKNOWLEDGEMENTS

The results described in this paper were attained during a visit of S.A. Kravchenko at the Otto-von-Guericke-Universität of Magdeburg which was supported by a fellowship of the Alexander von Humboldt Foundation.

References

[1] Ph. Baptiste, Scheduling equal-length jobs on identical parallel machines, Discrete Applied Mathematics 103(2000) 21-32.

[2] Ph. Baptiste, and P. Brucker, Scheduling equal processing time jobs: a survey. In: Leung Y.T., editor. Handbook of scheduling: algorithms, models, and performance analysis. CRC Press, Boca Raton, 2004, pp. 14-1 – 14-37.

[3] P. Brucker, and S.A. Kravchenko, Scheduling jobs with equal processing times and time windows on identical parallel machines, OSM Reihe P, Heft 257, Universität Osnabrück, Fachbereich Mathematik/Informatik, 2004.

[4] P. Brucker, and S.A. Kravchenko, Scheduling jobs with release times on parallel machines to minimize total tardiness, OSM Reihe P, Heft 258, Universität Osnabrück, Fachbereich Mathematik/Informatik, 2005.

[5] S.A. Kravchenko, On the complexity of minimizing the number of late jobs in unit time open shop, Discrete Applied Mathematics 100(2000) 127-132.

[6] B. Simons, Multiprocessor scheduling of unit-time jobs with arbitrary release times and deadlines, SIAM J. Comput. 12(1983) 294-299.