Discrete Applied Mathematics 161 (2013) 1586–1597

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Homogeneously non-idling schedules of unit-time jobs on identical parallel machines

Alain Quilliot [a], Philippe Chrétienne [b,*]

[a] Université Blaise Pascal, CNRS LIMOS, France
[b] Université Paris 6, CNRS LIP6, France

ARTICLE INFO

Article history:
Received 18 November 2011
Received in revised form 18 July 2012
Accepted 21 January 2013
Available online 19 February 2013

Keywords:
Scheduling unit-time tasks on m parallel machines
Non-idling machine
Matching
Feasibility criteria

ABSTRACT

In this paper, we study the basic homogeneous m-machine scheduling problem where weakly dependent unit-time jobs have to be scheduled within the time windows between their release dates and due dates so that, for any subset of machines, the set of the time units at which at least one machine is busy, is in interval. We first introduce the notions of pyramidal structure, k-hole, m-matching, preschedule, k-schedule and schedule for this problem. Then we provide a feasibility criteria for a preschedule. The key result of the paper is then to provide a structural necessary and sufficient condition for an instance of the problem to be feasible. We conclude by giving the directions of ongoing works and by bringing open questions related to different variants of the basic non-idling m-machine scheduling problem.

1. Introduction

Most scheduling problems assume that no cost is incurred when a machine waits between the completion of a job and the start of the next job. Moreover, it is well-known that such waiting delays are often necessary to get optimality. This is the key feature why list algorithms, that do not allow a machine to wait for a more urgent job, do not generally provide optimal schedules. However, in some applications such as those described in [8], the cost of making a running machine stop and restart later is so high that a non-idling constraint is put on the machine so that only schedules without any intermediate delays are required. For example, if the machine is an oven that must heat different and not compatible pieces of work at a given high temperature, keeping the required temperature of the oven while it is empty may clearly be too costly. Problems concerning power management policies may also yield similar scheduling problems [5] where for example each idling period has a cost and the total cost has to be minimized [1]. Note that the non-idling constraint will not necessarily ensure full machine utilization but will remove the cost of machine re-starts, maybe at the price of processing the jobs later.

Contrary to the well-known no-wait constraint in shop scheduling where no idle time is allowed between the successive operations of a same job, the non-idling machine constraint has just begun to receive research attention in the literature. To the best of our knowledge, the first work on such problems concerns the earliness–tardiness single-machine scheduling problem with no unforced idle time, where a Branch and Bound approach has been developed [11]. More recently, some aspects of the impact of the non-idling constraint on the complexity of single-machine scheduling problems as well as the important role played by the earliest starting time of a non-idling schedule has been studied in [3]. Moreover, in [6,2], exact methods have been designed to solve the basic one-machine non-idling problem and in [7], approximation algorithms have been developed for the non-idling single-machine scheduling problem with release and delivery times.

* Corresponding author. Tel.: +33 1 44 27 72 09; fax: +33 1 44 27 70 00.
E-mail addresses: alain.quilliot@isima.fr (A. Quilliot), Philippe.Chretienne@lip6.fr (P. Chrétienne).

0166-218X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2013.01.019

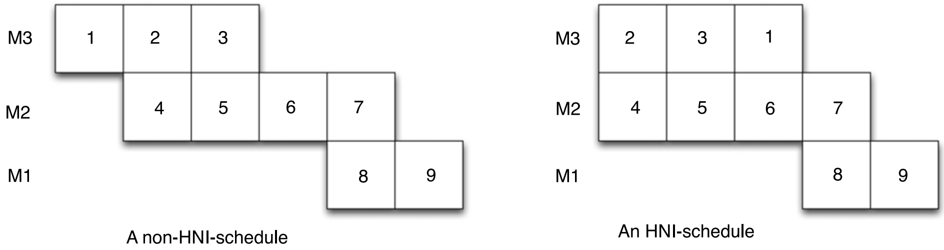

Fig. 1. A HNI-schedule and a non-HNI schedule.

In this paper, we study the basic homogeneous m-machine non-idling problem where weakly dependent unit-time jobs have to be scheduled within their time windows so that the non-idling constraint must be satisfied not only for each machine but for every subset of machines. As suggested by a referee, it is interesting to note that, when the jobs are independent, an approach based on transversals [10] leads to an algorithm, whose complexity is exponential in the number of jobs, solving the problem as the search of a path from a source to a sink in a graph whose nodes correspond to time-consecutive suitable triplets of transversals and where there is an arc linking two triplets if the two last transversals of the first triplet correspond to the two first transversals of the second triplet. In Section 2, the problem and the key notions of pyramidal structure, k-hole, m-matching, preschedule, k-schedule and schedule are defined. Section 3 provides feasibility criteria for the existence of an m-matching and a preschedule. Section 4 provides a necessary and sufficient condition for an instance to have at least a schedule. A conclusion finally resumes the main results of the paper and relates about future works and open questions about that basic m-machine non idling scheduling problem.

2. Problem definition

2.1. Preliminary definitions

We consider the discrete time space $\mathbb{N}$ each element of which is called a time-unit. A subset Θ of $\mathbb{N}$ is an *interval* if it is made of a finite number of consecutive time-units. The smallest (respectively largest) time-unit of an interval Θ is denoted by $a(\Theta)$ (respectively $b(\Theta)$). If p and q are two distinct natural numbers, the interval whose bounds are p and q is denoted by $I(p, q)$ (note that we may have $p < q$ or $q < p$). Interval Θ_2 is said to dominate interval Θ_1 (what is denoted by $\Theta_1 \propto \Theta_2$) if $b(\Theta_1) + 1 < a(\Theta_2)$. If $\Theta_1 \propto \Theta_2$, we denote by $Mid(\Theta_1, \Theta_2)$ the (non-empty) interval $[b(\Theta_1) + 1, a(\Theta_2) - 1]$. By convention, two intervals Θ_1 and Θ_2 are said to be *connected* if $\Theta_1 \cup \Theta_2$ is an interval.

2.2. The homogeneous non-idling scheduling problem

We are given a set $J = \{J_1, \ldots, J_n\}$ of n unit-time jobs that are to be processed on a set $M = \{M_1, \ldots, M_m\}$ of m identical machines. Job J_i must be executed within a given time-window $F(i) = \{r_i, \ldots, d_i\}$ which is an interval. It will be convenient to denote by r_{min} (respectively d_{max}) the smallest r_i (respectively largest d_i) and by $\mathcal{H}$ the interval $[r_{min}, d_{max}]$. The jobs are constrained by a *weak* precedence relation denoted by $\preceq$ where $J_i \preceq J_j$ means that J_j must not be performed before J_i. A schedule must also satisfy the so-called *homogeneous non-idling* (HNI in short) constraint: for any subset $M' \subseteq M$, the set of the time-units at which *at least one machine* in M' does not idle away is an interval. In Fig. 1, the schedule on the left does not satisfy the HNI constraint since, for the subset $\{M_1, M_3\}$, the set of the times units when M_1 or M_3 is busy is not an interval. On the contrary, the schedule on the right is an HNI-schedule.

A schedule (T, μ) assigns a time-unit $T(i)$ and a machine $\mu(i)$ to each job J_i so that :

1. $\forall J_i \in J, T(i) \in F(i)$;
2. $\forall t \in \mathcal{H}, |\{i|T(i) = t\}| \leq m$;
3. $J_i \preceq J_j \Rightarrow T(i) \leq T(j)$;
4. For any $M' \subseteq M$, the time units t at which there is at least a job J_i processed at t (i.e: $T(i) = t$) by a machine of M' ((i.e: $\mu(i) \in M'$) make a single interval.

The problem Π_0 is to decide whether a given instance $(J, F, \preceq, m)$ has at least one schedule. If the answer is yes, the instance is said to be *feasible*. It must be pointed out that the precedence relation $\preceq$ has not the same meaning as the classical one since if $J_i \preceq J_j$, then J_i and J_j may be processed at the same time-unit. Clearly, due to the machine constraint, there is no difference when $m = 1$. It is also of interest to note that when $\preceq$ has the usual meaning, the corresponding problem is NP-complete, since in the reduction of the CLIQUE problem described in [9], the schedule of the instance corresponding to a "yes" instance of CLIQUE satisfies the HNI constraint.

Let $I = (J, F, \preceq, m)$ be an instance of Π_0. A time function T that satisfies the first two conditions will be called a m-*matching* of I. A function T that satisfies the first three conditions will be called a *preschedule* of I.

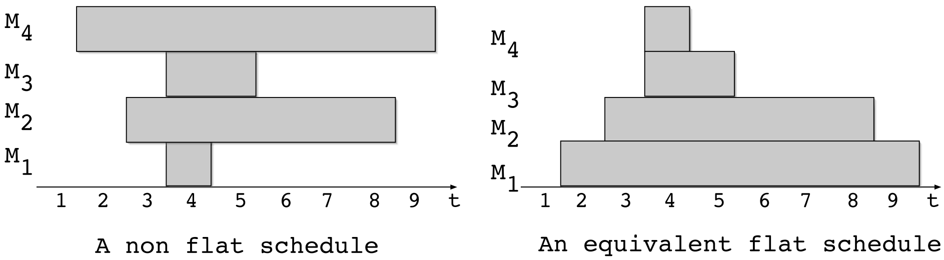

Fig. 2. A non-histogram schedule and an equivalent histogram schedule.

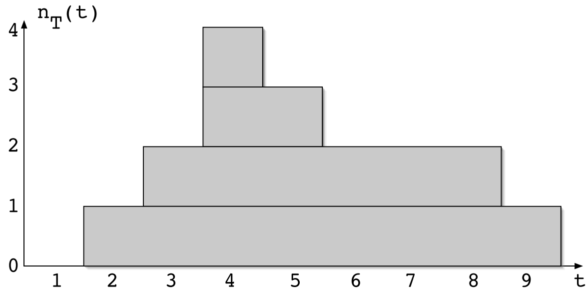

Fig. 3. A pyramidal function $n_T(t)$.

Let T be an m-matching of I. We denote by $s(T)$ (respectively $e(T)$) the smallest (respectively largest) time-unit such that at least one job is processed in T. If $s(T) \le t \le e(T)$, we denote by $n_T(t)$ the number of jobs scheduled at time-unit t.

(T, μ) is said to be an *histogram schedule* of Π_0 if, for any time unit t such that $n_T(t) > 0$, $\{M_1, \ldots, M_{n_T(t)}\}$ is the set of the busy machines at time unit t. Clearly, if (T, μ) is a schedule, we get an equivalent (i.e: same starting times of the tasks) histogram schedule by reassigning the tasks processed at time unit t to the machines $M_1, \ldots, M_{n_T(t)}$. Fig. 2 shows a non-histogram schedule and an equivalent histogram schedule.

So, from now on, we will only consider histogram schedules and we notice that an histogram schedule is completely defined by the function T. The following property characterizes the histogram schedules of an instance of Π_0 in terms of the shape of the function $n_T(t)$. The shape of $n_T(t)$ is said to be "pyramidal" if there do not exist 3 time units t_0, t_1, t_2 such that $t_0 < t_1 < t_2$, $n_T(t_0) > n_T(t_1)$, and $n_T(t_2) > n_T(t_1)$ (see Fig. 3).

Property 1. *Let T be a preschedule of an instance of Π_0. T is an histogram schedule if and only if the function $n_T(t)$, has a pyramidal shape.*

Proof. Assume that T is an histogram schedule and that the function $n_T(t)$ has not a pyramidal shape. Then there exist 3 time units t_0, t_1, t_2 such that $t_0 < t_1 < t_2$, $n_T(t_0) > n_T(t_1)$, and $n_T(t_2) > n_T(t_1)$. Clearly machine $M_{n_T(t_1)+1}$ is idle at time unit t_1 while it is busy at time units t_0 and t_2. So T is not a schedule. Conversely, if the function $n_T(t)$ of a preschedule T has a pyramidal shape, then T clearly satisfies the HNI condition and thus is an histogram schedule. $\square$

The family of time windows $F(i)$, $i \in \{1, \ldots, n\}$ is said to be *consistent* with the precedence relation $\preceq$ if:

$$J_i \preceq J_j \Rightarrow r_i \le r_j \text{ and } d_i \le d_j.$$

Let us now recall a well-known property that allows to restrict our study to instances of Π_0 such that F is consistent with respect to $\preceq$.

Property 2. *Let $I = (J, F, \preceq, m)$ be an instance of Π_0. There is an instance $I' = (J, F', \preceq, m)$ of Π_0 with the same set of schedules as I such that F' is consistent with respect to $\preceq$ and such that for any job J_i, $F'(i) \subseteq F(i)$.*

So, from now on, the consistency of F with respect to $\preceq$ will be assumed for any instance $(J, F, \preceq, m)$ of Π_0.

3. Existence of a preschedule

In this section, we first use the Hall's marriage theorem [4] to get a characterization of the instances for which an m-matching exists. Then we show that the existence of an m-matching implies the existence of a preschedule. If Θ is an interval, it will be useful to denote by $J(\Theta)$ the subset of the jobs J_i such that $F(i) \subseteq \Theta$.

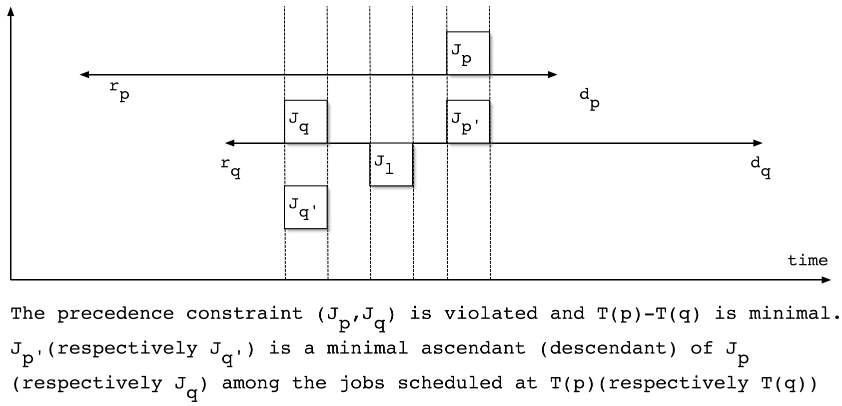

Fig. 4. Getting a preschedule from an m-matching.

3.1. Existence of an m-matching

The following property is a straightforward application of the Hall's marriage theorem:

Property 3. *The instance $(J, F, \preceq, m)$ of Π_0 has at least one m-matching if and only if, for any interval Θ of $\mathcal{H}$, we have $|J(\Theta)| \leq |\Theta| \times m$.*

Proof. From the Hall's marriage theorem, we know that the instance $(J, F, \preceq, m)$ of Π_0 has at least one m-matching if and only if for any subset I of J, we have $|I| \leq m \times |\Gamma(I)|$ where $\Gamma(I) = \cup_{i\in I} F(i)$. If $I = J(\Theta)$, then we have $\Gamma(I) \subseteq \Theta$ so that $|\Gamma(I)| \leq |\Theta|$ and $m \times |\Theta| \geq m \times |\Gamma(I)| \geq |I| = |J(\Theta)|$.

Conversely, let I be an arbitrary subset of J. Clearly $\Gamma(I)$ is the union $\cup_{k=1}^{r} \Theta_k$ of non-empty and pairwise disjoint intervals. With interval Θ_k is associated the subset I_k of the jobs J_i of I such that $F(i) \subseteq \Theta_k$. The subsets $I_1, \ldots, I_r$ make a partition of I such that $\Gamma(I_k) = \Theta_k$. Since $I_k \subseteq J(\Theta_k)$, we have

$$m \times |\Gamma(I)| = m \times \sum_{k=1}^{r} |\Theta_k| \geq \sum_{k=1}^{r} |J(\Theta_k)| \geq \sum_{k=1}^{r} |I_k| \geq |I|. \quad \square$$

3.2. Existence of a preschedule

We now show that any m-matching of the instance I may be transformed into a preschedule.

Property 4. *Let $I = (J, F, \preceq, m)$ be an instance of Π_0. I has at least one preschedule if and only if I has at least one m-matching.*

Proof. Let T be an m-matching of the instance I which is not a preschedule. (J_i, J_j) is called an *inversion* of T if $T(i) > T(j)$ and $J_i \preceq J_j$. Let $INV(T)$ be the set of the inversions of T. If $(i, j) \in INV(T)$, let $B(i, j)$ be the set of the jobs J_l such that $T(j) < T(l) < T(i)$ and assume that (p, q) is an inversion of T such that $T(p) - T(q)$ is minimal. Finally, we denote by $J_{p'}$ a minimal job (with respect to $\preceq$) among the ascendants of J_p that are scheduled at time-unit $T(p)$ and by $J_{q'}$ a maximal job (with respect to $\preceq$) among the descendants of J_q that are scheduled at time-unit $T(q)$ (see Fig. 4). From the definition of p' and q', it is clear that we have $(p', q') \in INV(T)$.

Let us first assume that $B(p, q) = \emptyset$. By exchanging $J_{p'}$ and $J_{q'}$, we get, from the consistency of F and the definition of p' and q', an m-matching T' such that $|INV(T')| = |INV(T)| - 1$.

Assume now that $J_l \in B(p, q)$. From the definition of (p', q'), we cannot have $J_{q'} \preceq J_l$ since otherwise we would have $(p', l) \in INV(T)$ and $T(p') - T(l) < T(p) - T(q)$. Similarly, we cannot have $J_l \preceq J_{p'}$ since otherwise we would have $(l, q') \in INV(T)$ and $T(l) - T(q') < T(p) - T(q)$. So, again, by exchanging $J_{p'}$ and $J_{q'}$, we get, from the consistency of F and the definition of p' and q', an m-matching T' such that $|INV(T')| \leq |INV(T)| - 1$.

Finally, iterating the above transformation as long as the current m-matching has at least one inversion, we stop after at most $O(n^2)$ iterations and get a preschedule. $\square$

4. Existence of a schedule

Let $(J, F, \preceq, m)$ be an instance of Π_0. If the condition provided by Property 3 is satisfied, we know from Property 4 that $(J, F, \preceq, m)$ has at least a preschedule. However, such a preschedule may have k-holes and thus may not fit the "pyramidal" structure required by the schedules of $(J, F, \preceq, m)$. This section will provide an additional condition so that an instance has

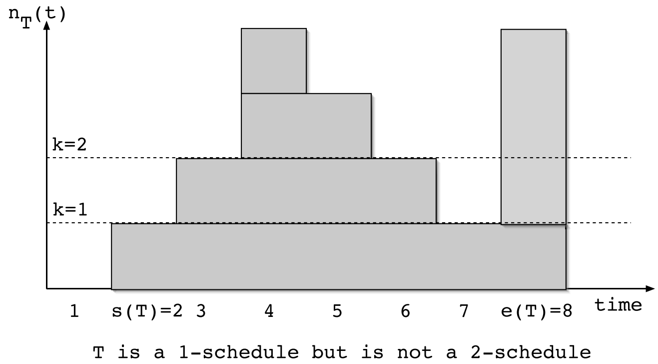

Fig. 5. k-holes and k-schedules.

at least a schedule. We first give two simple lower bounds that must be met by a schedule. In order to get a set of basic moves from a preschedule T in the set of preschedules, we then define the *propagation graph* $G(T)$ of a preschedule T. These basic moves will correspond to some *labelled paths* of $G(T)$ that will be called the *propagation paths* of T.

4.1. Preliminary definitions and properties

A time interval $\Theta \subseteq \{s(T), \dots, e(T)\}$ is a T-*block* if every job J_i scheduled in Θ is such that $F(i) \subseteq \Theta$. Time-unit $t \in \{s(T)+1, \dots, e(T)-1\}$ is a k-*hole* of T if there exist t' and t'' such that $s(T) \le t' < t < t'' \le e(T)$, $n_T(t') > k$, $n_T(t) = k$ and $n_T(t'') > k$ (In Fig. 5, time unit 6 is a 2-hole and time unit 7 is a 1-hole).

Let T be a preschedule of I. From the pyramidal structure of a schedule, it is straightforward to see that T is a schedule if and only if, for any $k \in \{0, \dots, m-1\}$, T has no k-hole. This leads us to define the notion of a k-schedule for $k \in \{1, \dots, m\}$. T is called a k-*schedule* of I if T has no l-hole for any $l \in \{0, \dots, k-1\}$. Equivalently, a preschedule T is a k-schedule if the time function $t \to \min\{n_T(t), k\}$ has a pyramidal shape. The time diagram of Fig. 5 represents a 1-schedule but not a 2-schedule since $t = 7$ is a 1-hole. Note that any schedule is an m-schedule and that the converse is true.

Let Θ be an interval. We denote by $Int(\Theta)$ the set of the non empty intervals contained in Θ and by $\lambda(\theta)$ the *non negative integer* $\max_{\Theta' \in Int(\Theta)} \lceil \frac{|J(\Theta')|}{|\Theta'|} \rceil$. Then we have the following property (whose proof is omitted).

Property 5. *Let Θ be an interval. In any m-matching of $(J, F, \preceq, m)$, there is at least one time-unit of Θ such that at least $\lambda(\Theta)$ machines are busy.*

Notice that if Θ is a T-block, then we have $\lambda(\Theta) \ge \lceil \frac{\sum_{t\in\Theta} n_T(t)}{|\Theta|} \rceil$ since in that case, $J(\Theta)$ is exactly the set of the jobs scheduled in Θ by T. Assume now that Θ_1 and Θ_2 are two intervals of $\mathcal{H}$ such that $\Theta_1 \propto \Theta_2$. If we denote by $\mu(\Theta_1, \Theta_2)$ the number $|Mid(\Theta_1, \Theta_2)| \times \min\{\lambda(\Theta_1), \lambda(\Theta_2)\}$, we have the following property (whose proof is omitted).

Property 6. *In any schedule of $(J, F, \preceq, m)$, at least $\mu(\Theta_1, \Theta_2)$ jobs are scheduled in the interval $Mid(\Theta_1, \Theta_2)$.*

The following two properties, that concern T-blocks, will be useful to derive the main theorem of this paper.

Property 7. *Let T be an m-matching and let Θ_1, Θ_2 be two connected T-blocks. Then $\Theta_1 \cup \Theta_2$ is a T-block and $\lambda(\Theta_1 \cup \Theta_2) \ge \max\{\lambda(\Theta_1), \lambda(\Theta_2)\}$.*

Proof. If job J_i is scheduled in Θ_1, then $F(i) \in Int(\Theta_1) \subseteq Int(\Theta_1 \cup \Theta_2)$ since Θ_1 is a T-block. In the same way, if job J_i is scheduled in Θ_2, then $F(i) \in Int(\Theta_2) \subseteq Int(\Theta_1 \cup \Theta_2)$. So $\Theta_1 \cup \Theta_2$ is a T-block. Assume that $\lambda(\Theta_1) = \frac{|J(\Theta_1')|}{|\Theta_1'|}\rceil$ where $\Theta_1' \in Int(\Theta_1)$. Since $\Theta_1' \in Int(\Theta_1 \cup \Theta_2)$, we get $\lambda(\Theta_1) \le \lambda(\Theta_1 \cup \Theta_2)$. In the same way, we get $\lambda(\Theta_2) \le \lambda(\Theta_1 \cup \Theta_2)$. We thus conclude that $\lambda(\Theta_1 \cup \Theta_2) \ge \max\{\lambda(\Theta_1), \lambda(\Theta_2)\}$. $\square$

Property 8. *Let T be an m-matching, let Θ_1 be a T-block and let Θ_2 be an interval such that $\Theta_1 \cap \Theta_2 = \emptyset$. If for any $(u, t) \in \Theta_1 \times \Theta_2$, $n_T(u) \ge n_T(t)$ (respectively $n_T(u) > n_T(t)$), then $\lambda(\Theta_1) \ge \lambda(\Theta_2)$ (respectively $\lambda(\Theta_1) > \lambda(\Theta_2)$).*

Proof. Let Θ be an arbitrary interval of Θ_2 (i.e: $\Theta \in Int(\Theta_2)$). Every job of $J(\Theta)$ is scheduled in Θ so that we have $|J(\Theta)| \le \sum_{t\in\Theta} n_T(t)$. Since Θ_1 is a T-block, $J(\Theta_1)$ is exactly the set of the jobs scheduled in Θ_1 so that we have:

$$\lambda(\Theta_1) \ge \left\lceil \frac{|J(\Theta_1)|}{|\Theta_1|} \right\rceil = \left\lceil \frac{\sum_{u\in\Theta_1} n_T(u)}{|\Theta_1|} \right\rceil .$$

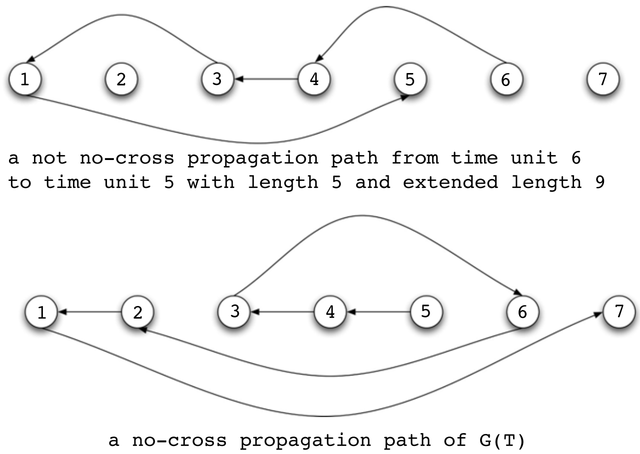

Fig. 6. Propagation paths.

Assume that $M = \max_{t\in\Theta} n_T(t)$. Then we have $|J(\Theta)| \leq M|\Theta|$ and $|J(\Theta_1)| = \sum_{u\in\Theta_1} n_T(u) \geq M|\Theta_1|$. We thus derive that

$$\lambda(\Theta_1) \geq \left\lceil \frac{|J(\Theta_1)|}{|\Theta_1|} \right\rceil \geq M \geq \left\lceil \frac{|J(\Theta)|}{|\Theta|} \right\rceil$$

and we may conclude that $\lambda(\Theta_1) \geq \lambda(\Theta_2)$.

For the special case when $\forall (u, t) \in \Theta_1 \times \Theta_2$, $n_T(u) > n_T(t)$, we have

$$\lambda(\Theta_1) \geq \left\lceil \frac{\sum\limits_{u\in\Theta_1} n_T(u)}{|\Theta_1|} \right\rceil > M$$

and we now may conclude that $\lambda(\Theta_1) > \lambda(\Theta_2)$. $\square$

4.2. Propagation paths of an m-matching

Given an m-matching T, we now define the *propagation paths of* T. The *propagation graph* $G(T) = (\mathcal{H}, E(T))$ is the directed graph such that $(t, t') \in E(T)$ if $t \neq t'$ and there is at least one job J_i scheduled at t such that $t' \in F(i)$. If job J_i is scheduled at t and if $t' \in F(i)$, then J_i is said to be a *label* of the arc (t, t').

A *propagation path* of T is an elementary path $\gamma = (t_0, \ldots, t_k)$ of $G(T)$. The subpath of γ from t_i to t_j will be denoted by $\gamma(t_i, t_j)$. The length $L(\gamma)$ of γ is $k+1$ (i.e: number of vertices of γ) and the *extended length* $\hat{L}(\gamma)$ of γ is the sum $\sum_{s=1}^{k} |t_s - t_{s-1}|$. A propagation path γ is *monotone* if the sequence $(t_0, \ldots, t_k)$ is decreasing. A propagation path γ is *no-cross* if for any $r \in \{1, \ldots, k\}$, we have either $t_r > \max_{i\in\{1,\ldots,r-1\}}\{t_i\}$ or $t_r < \min_{i\in\{1,\ldots,r-1\}}\{t_i\}$ (see Fig. 6).

A propagation path is *labelled* when all its arcs are assigned a label and is said to be *fitted* if it is labelled and satisfies $n_T(t_k) < m$. It is important to note that, from the consistency property, it may be assumed that the label of an arc (t, t') is a job which is *minimal with respect to* $\preceq$ among the jobs scheduled at time t by T. Only such labelled propagation paths will be considered in the rest of the paper.

If $\gamma = (t_0, \ldots, t_k)$ is a propagation path of $G(T)$ fitted with the labelling $\sigma = (J_{[0]}, \ldots, J_{[k-1]})$, we get an other m-matching $T' = \mathcal{T}(T, \gamma, \sigma)$ by putting $T'(J_{[p]}) = t_{p+1}$ for all $p \in \{0, \ldots, k-1\}$. Fig. 7 illustrates this transformation. Let T be a preschedule and let $\gamma = (t_0, \ldots, t_k)$ be a propagation path of $G(T)$. If γ has at least a labelling σ such that $T' = \mathcal{T}(T, \gamma, \sigma)$ is a preschedule, then γ is said to be compatible with $\preceq$ ($\preceq$*-compatible* in short).

The aim of the two following lemmas is to show that no-cross propagation paths allow to move from a preschedule into another one.

Property 9. *Let T be a preschedule and assume that $\gamma = (t_0, \ldots, t_q)$ is a propagation path of $G(T)$ from $t_0 = u$ to $t_q = v$. Then there also exists a no-cross propagation path from u to v in $G(T)$.*

Proof. Assume that γ is not no-cross and let t_r $(2 \leq r \leq q)$ be the first node of γ such that $\min_{i\in\{1,\ldots,r-1\}}\{t_i\} < t_r < \max_{i\in\{1,\ldots,r-1\}}\{t_i\}$. From the definition of t_r, we know there is a smallest index s $(0 \leq s \leq r-2)$ such that t_r belongs to $I(t_s, t_{s+1})$. Thus $(t_s, t_r) \in E(T)$ and $\gamma(t_0, t_s) \cdot (t_s, t_r)$ is no-cross. The above transformation may then be iterated while the current propagation path is not no-cross. $\square$

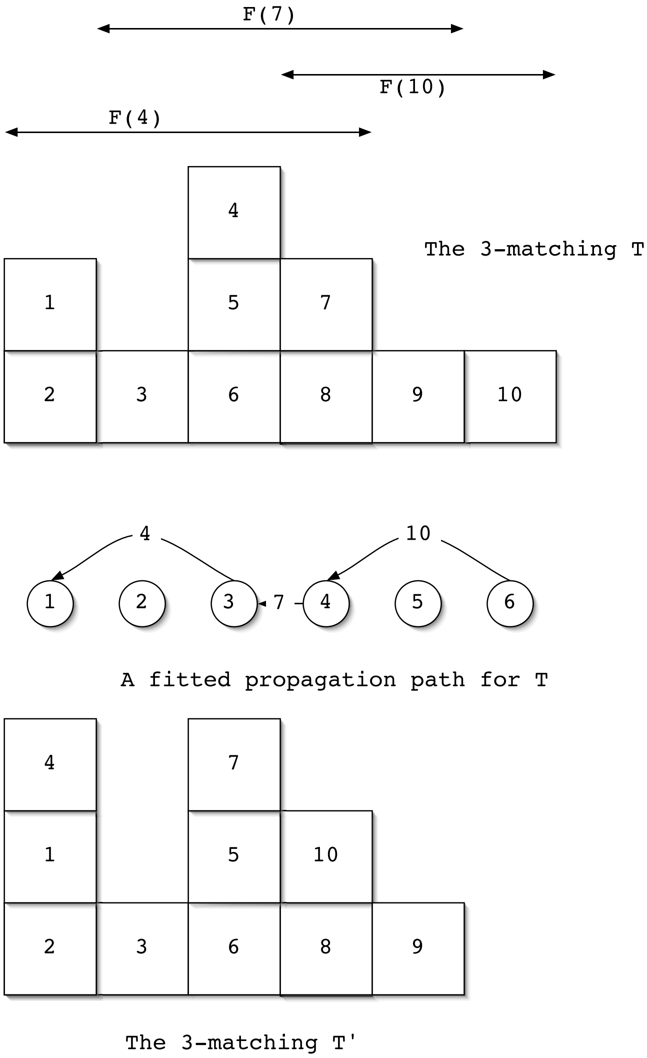

Fig. 7. The new m-matching $T' = \mathcal{T}(T, \gamma, \sigma)$.

The next lemma shows we do not need take care of the $\preceq$-compatibility since if there is a fitted propagation path from time unit t_0 to time unit t_q, there also exists a $\preceq$-compatible fitted propagation path from t_0 to t_q.

Property 10. *Let T be a preschedule and assume that $(t_0, \ldots, t_q)$ is a propagation path of $G(T)$ from $t_0 = u$ to $t_q = v$ (where $n_T(t_q) < m$). Then there exists a no cross and $\preceq$-compatible propagation path from u to v in $G(T)$.*

Proof. Let $\gamma = (t_0, \ldots, t_q)$ be a no cross propagation path from u to v whose extended length $\hat{L}(\gamma)$ is minimum and whose length $L(\gamma)$ is maximum among the propagation paths with a minimum extended length. Assume also that γ is not compatible with $\preceq$ and let $\sigma = (J_{[0]}, \ldots, J_{[q-1]})$ be a labelling of γ. Jobs $J_{[0]}, \ldots, J_{[q-1]}$ are called the *moving jobs* while the other jobs are the *static jobs*. Since T is a preschedule and γ is not $\preceq$-compatible, $\preceq$ is violated in $T' = \mathcal{T}(T, \gamma, \sigma)$ either because an inversion has been created between a static job J' scheduled at t' and a moving job $J_{[s]}$ scheduled at t_{s+1} (case (1) below) or between two moving jobs $J_{[r]}$ and $J_{[s]}$ respectively scheduled at time t_{r+1} and t_{s+1} (case (2) below).

Case (1).

For that case, we either have ($J' \preceq J_{[s]}$ and $t_{s+1} < t'$) or ($J_{[s]} \preceq J'$ and $t_{s+1} > t'$). Since these two subcases are symmetric and the same line of reasoning applies to both, we will only consider the first one.

Let us assume first that t' is not a node of γ. Then consider the smallest index p such that $t' \in I(t_p, t_{p+1})$. Assume first that $p = s$. Since $J' \preceq J_{[s]}$, we have $(t', t_{s+1}) \in E(T)$ from the consistency of F. Since $t_{s+1} < t' < t_s$, we have $(t_s, t') \in E(T)$. Now $\gamma' = \gamma(t_0, t_s) \cdot (t_s, t') \cdot (t', t_{s+1}) \cdot \gamma(t_{s+1}, t_q)$ is a no cross propagation path from $t_0 = u$ to $t_q = v$ such that $\hat{L}(\gamma') = \hat{L}(\gamma)$ and $L(\gamma') > L(\gamma)$. A contradiction.

Consider now the case when $p < s$. From the definition of p, we know that $(t_p, t') \in E(T)$. Moreover, since $J' \preceq J_{[s]}$, we have $(t', t_{s+1}) \in E(T)$ from the consistency of F (see Fig. 8). Since γ has a minimum extended length and from the definition

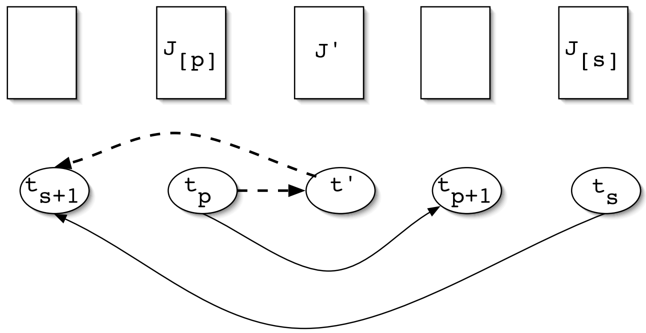

Fig. 8. $J' \preceq J_{[s]}$, $t_{s+1} < t'$ and $t' \notin \gamma$.

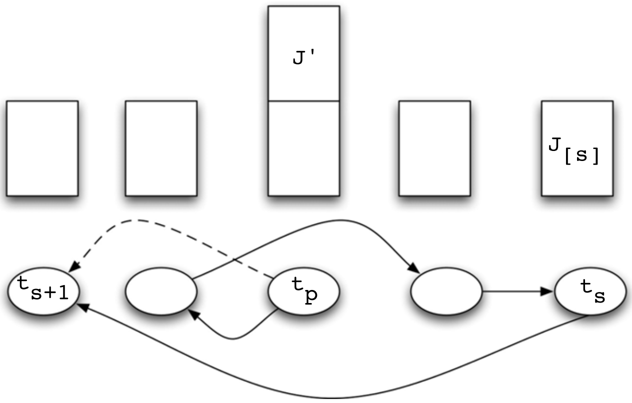

Fig. 9. $J' \preceq J_{[s]}$, $t_{s+1} < t'$ and $t' \in \gamma$.

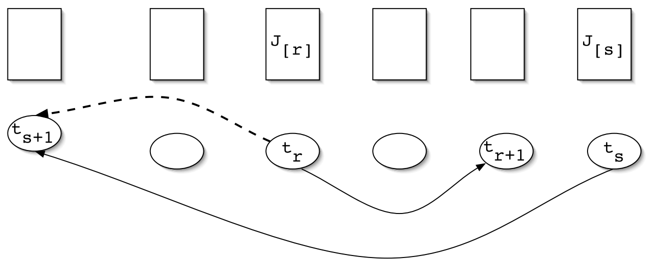

Fig. 10. $J_{[r]} \preceq J_{[s]}$ and $t_{s+1} < t_{r+1}$.

of p, we have that $\gamma(t_0, t_p) \cdot (t_p, t') \cdot (t', t_{s+1}) \cdot \gamma(t_{s+1}, t_q)$ is a no-cross propagation path whose extended length is strictly smaller than $\hat{L}(\gamma)$, what is a contradiction.

Assume now that t' is a node of γ and more precisely that $t' = t_p$ where $0 \leq p \leq q - 1$ (see Fig. 9).

Since γ is no-cross, we have $p \leq s$. Moreover, since $J_{[s]}$ is a minimal job (with respect to $\preceq$) among the jobs scheduled at time unit t_s, we cannot have $J' \preceq J_{[s]}$. So we know that $p < s$. Since $J' \preceq J_{[s]}$, we have $(t', t_{s+1}) \in E(T)$. Now $\gamma(t_0, t_p) \cdot (t_p, t_{s+1}) \cdot \gamma(t_{s+1}, t_q)$ is a no-cross propagation path whose extended length is strictly smaller than $\hat{L}(\gamma)$, what is a contradiction.

Case (2).

As shown in Fig. 10, let us assume that $J_{[r]} \preceq J_{[s]}$ and $t_{s+1} < t_{r+1}$ (the symmetric case may be treated in a same way). Since γ is no cross, we must have $r < s$. Since $J_{[r]} \preceq J_{[s]}$, we have $(t_r, t_{s+1}) \in E(T)$ from the consistency of F. So, $\gamma(t_0, t_r) \cdot (t_r, t_{s+1}) \cdot \gamma(t_{s+1}, t_q)$ is a no-cross propagation path whose extended length is strictly smaller than $\hat{L}(\gamma)$, what is a contradiction.

We thus may conclude that γ is a no-cross and $\preceq$-compatible propagation path from u to v in $G(T)$. $\square$

Let T be a preschedule and let u be a time unit such that $n_T(u) > 0$. The next lemma gives an important property of the set $A_T(u)$ of the time units that may be reached by the propagation paths of $G(T)$.

Property 11. *$A_T(u)$ is a T-block.*

Proof. Assume first that $A_T(u)$ is made of more than one interval. Since $u \in A_T(u)$, we may denote by Θ_0 the interval of $A_T(u)$ that contains u. Let Θ_1 be an interval of $A_T(u)$ that is a neighbour of Θ_0 and assume for example that $\Theta_0 \propto \Theta_1$. Let $t \in Mid(\Theta_0, \Theta_1)$ and $v \in \Theta_1$. There is a propagation path $\gamma = (t_0, \dots, t_k)$ from u to v in $G(T)$. Let (t_q, t_{q+1}) be the first arc of γ such that $t_q \le b(\Theta_0)$ and $t_{q+1} \ge a(\Theta_1)$. Then, for any job J_i that is a label of (t_q, t_{q+1}), we have that $t \in F(i)$, so that we may conclude that $t \in A_T(u)$, a contradiction. Thus $A_T(u)$ is an interval. Assume now that $T(i) = t \in A_T(u)$. If $t' \in F(i) \setminus A_T(u)$, then $(t, t') \in E(T)$ and we would have $t' \in A_T(u)$, a contradiction. $A_T(u)$ is thus a T-block. □

4.3. Feasibility of an instance

We are now able to provide a characterization of the instances $(J, F, \preceq, m)$ which are feasible for the problem Π_0.

Property 12. *The instance $(J, F, \preceq, m)$ of Π_0 is feasible if and only if the following two conditions are satisfied:*

1. *for any interval Θ of $\mathcal{H}$, $|J(\Theta)| \le |\Theta| \times m$;*
2. *for any sequence $(\Theta_1, \dots, \Theta_p)$ of intervals of $\mathcal{H}$ such that $\Theta_1 \propto \dots \propto \Theta_p$, we have*

$$|J \setminus \cup_{s=1}^{p} J(\Theta_s)| \ge \sum_{s=1}^{p-1} \mu(\Theta_s, \Theta_{s+1}).$$

The "only if" part of the proof is a straightforward consequence of Properties 3 and 6. In order to prove the "if" part, we introduce the analogue of Property 12 for k-schedules and prove it by induction on k.

Recall that, for $k \in \{1, \dots, m\}$, a preschedule T is a *k-schedule* if T has no l-hole for any $l \in \{0, \dots, k-1\}$.

We now adapt the definition of $\mu(\Theta_1, \Theta_2)$ (where Θ_1 and Θ_2 are two time intervals such that $\Theta_1 \propto \Theta_2$) to k-schedules by defining $\tilde{\mu}(\Theta_1, \Theta_2, k)$ by the number $|Mid(\Theta_1, \Theta_2)| \times \min\{\lambda(\Theta_1), \lambda(\Theta_2), k\}$. Then the following property, whose proof is omitted, is the analogue of Property 6.

Property 13. *Let Θ_1 and Θ_2 be two time intervals such that $\Theta_1 \propto \Theta_2$. In any k-schedule of $(J, F, \preceq, m)$, at least $\tilde{\mu}(\Theta_1, \Theta_2, k)$ jobs are scheduled in the interval $Mid(\Theta_1, \Theta_2)$.*

We now prove the analogue of Property 12 for k-schedules.

Property 14. *The instance $(J, F, \preceq, m)$ of Π_0 has at least one k-schedule if and only if the following two conditions are satisfied:*

1. *for any Θ of $\mathcal{H}$, $|J(\Theta)| \le |\Theta| \times m$;*
2. *for any sequence $(\Theta_1, \dots, \Theta_p)$ of intervals of $\mathcal{H}$ such that $\Theta_1 \propto \dots \propto \Theta_p$, we have*

$$|J \setminus \cup_{s=1}^{p} J(\Theta_s)| \ge \sum_{s=1}^{p-1} \tilde{\mu}(\Theta_s, \Theta_{s+1}, k).$$

Proof. Let $I = (J, F, \preceq, m)$ be an instance of Π_0. The "only if" part of the proof comes directly from Properties 3 and 13. For the "if" part, we assume that the property is true for k-schedules and we assume that the instance I has no $(k+1)$-schedule. If I has at least a preschedule but has no k-schedule, then we know from the induction that there is a sequence $(\Theta_1, \dots, \Theta_p)$ of time intervals such that $\Theta_1 \propto \dots \propto \Theta_p$ such that

$$|J \setminus \cup_{s=1}^{p} J(\Theta_s)| < \sum_{s=1}^{p-1} \tilde{\mu}(\Theta_s, \Theta_{s+1}, k).$$

Since $\tilde{\mu}(\Theta_s, \Theta_{s+1}, k) \le \tilde{\mu}(\Theta_s, \Theta_{s+1}, k+1)$, we also have

$$|J \setminus \cup_{s=1}^{p} J(\Theta_s)| < \sum_{s=1}^{p-1} \tilde{\mu}(\Theta_s, \Theta_{s+1}, k+1)$$

and we are done.

We now assume that I has at least one k-schedule and that any such schedule has at least a k-hole. Let $\Sigma_k(I)$ be the set of the k-schedules of I. If $T \in \Sigma_k(I)$, we need some notations that are attached to T but for which the reference to T will be omitted to get simpler notations:

- H is the set of k-holes and $h = |H|$;
- S_p, $p \in \{1, \dots, m\}$ is the set of the time units t such that $n_T(t) = p$ and $N_p = |S_p|$;
- $\hat{S} = \cup_{p=k+2}^{m} S_p$;
- u^- (respectively u^+) is the smallest (respectively largest) k-hole;
- for $p \in \{1, \dots, k+1\}$, C_p^- (respectively C_p^+) is the smallest (respectively largest) time unit t such that $n_T(t) \ge p$.

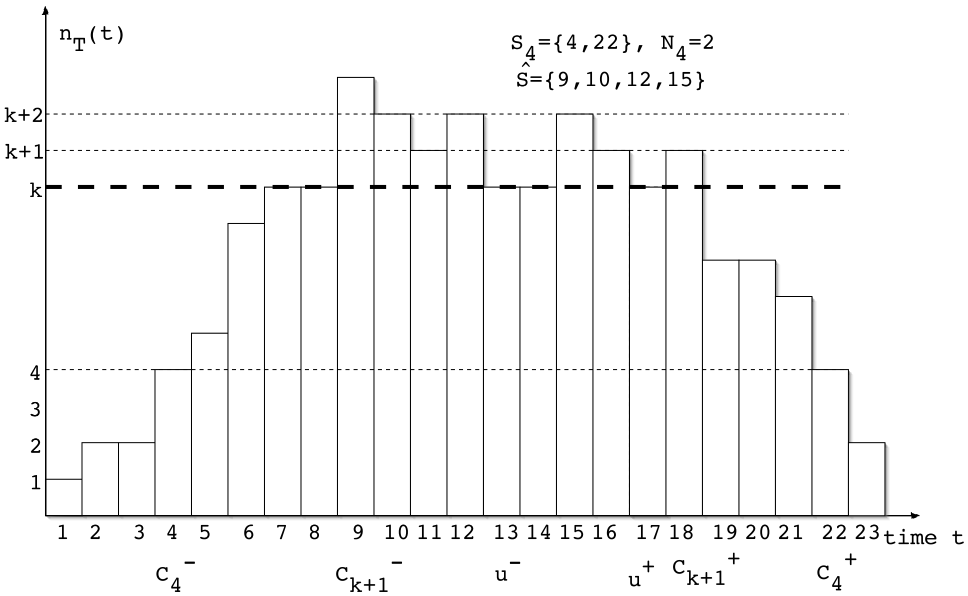

Fig. 11. A schedule T of $\Sigma_k(I)$.

Note that for $p \in \{1, \ldots, k+1\}$, we have $C_p^- < u^-$ and $C_p^+ > u^+$ and that for $p \in \{2, \ldots, k+1\}$, we have $C_{p-1}^- \leq C_p^-$ and $C_{p-1}^+ \geq C_p^+$. Fig. 11 illustrates these notations.

Since $\Sigma_k(I)$ is not empty, we consider one of its schedules, denoted by T, such that:

1. h is minimum;
2. the vector $(N_m, \ldots, N_1)$ is minimal for the lexicographic order among the schedules such that h is minimum.

From Property 10 and Condition 1., we know that there is no propagation path in $G(T)$ from a time unit of $\hat{S}$ to a time unit of H. Similarly, there is no propagation path γ in $G(T)$ from a time unit $t \in \hat{S}$ to a time unit $C_p^+ + 1$ (or $C_p^- - 1$) for $p \in \{1, \ldots, k+1\}$ since otherwise a labelling σ of γ would exist such that $T' = \mathcal{T}(T, \gamma, \sigma)$ would be a k-schedule such that, if $t \in S_l$ $(l \geq k+2)$, we would have $N'_l = N_l - 1$ and $N'_q = N_q$ for $q \in \{k+2, \ldots, m\} \setminus \{l\}$. So T would not satisfy Condition 2.

For $u \in \hat{S}$, it comes from Property 11 that $A_T(u)$ is a T-block satisfying $A_T(u) \subseteq S_{k+1} \cup \hat{S}$ and $\lambda(A_T(u)) \geq k+2$. Let us now consider the connected components of the intersection graph of the family of intervals $A_T(u)$ for $u \in \hat{S}$. We know from Property 7 that the intervals associated with these components are pairwise disjoint T-blocks that we denote by $\mathcal{M}_1, \ldots, \mathcal{M}_s$ and for which we may assume without loss of generality that $\mathcal{M}_1 \propto \cdots \propto \mathcal{M}_s$. From the definition of $\mathcal{M}_1, \ldots, \mathcal{M}_s$, we have:

$$\hat{S} \subseteq \mathcal{M}_1 \cup \cdots \cup \mathcal{M}_s \subseteq S_{k+1} \cup \hat{S}$$

and from Property 7, we also have $\lambda(\mathcal{M}_i) \geq k+2$ for $i \in \{1, \ldots, s\}$. Now let $i \in \{1, \ldots, s-1\}$. Any $t \in Mid(\mathcal{M}_i, \mathcal{M}_{i+1})$ is such that $n_T(t) \leq k+1$ so that we have:

$$\sum_{t \in Mid(\mathcal{M}_i, \mathcal{M}_{i+1})} n_T(t) \leq (k+1)|Mid(\mathcal{M}_i, \mathcal{M}_{i+1})| = \tilde{\mu}(\mathcal{M}_i, \mathcal{M}_{i+1}, k+1).$$

Consider now time unit $t = C_p^+$ for $p \in \{1, \ldots, k+1\}$. Again from Condition 1., there is no propagation path in $G(T)$ from t to a time-unit of H and, from Condition 2., there is no propagation path in $G(T)$ from t to any time unit $C_j^+ + 1$ for $j \in \{1, \ldots, p\}$. So, the intervals $A_T(C_p^+)$ are T-blocks such that $\lambda(A_T(C_p^+)) \geq p$ and are all included in $(S_p \cup \cdots \cup S_{k+1} \cup \hat{S}) \setminus H$. Considering the connected components of the intersection graph of the intervals $A_T(C_p^+)$ for $p \in \{1, \ldots, k+1\}$, we get a family $\mathcal{R}_1, \ldots, \mathcal{R}_r$ of T-blocks such that:

1. $\mathcal{R}_r \propto \cdots \propto \mathcal{R}_1$;
2. $\lambda(\mathcal{R}_r) \geq k+1$ (since $A_T(C_{k+1}^+) \subset \mathcal{R}_r$ and Property 7);
3. $\lambda(\mathcal{R}_r) > \cdots > \lambda(\mathcal{R}_1)$ (Property 8);
4. for any $p \in \{1, \ldots, k+1\}$, C_p^+ belongs to an interval $\mathcal{R}_{i(p)}$ such that $\lambda(\mathcal{R}_{i(p)}) \geq p$;
5. for any $i \in \{2, \ldots, r\}$, we have:
$$\sum_{t \in Mid(\mathcal{R}_i, \mathcal{R}_{i-1})} n_T(t) = \lambda(\mathcal{R}_{i-1})|Mid(\mathcal{R}_i, \mathcal{R}_{i-1})| \leq \tilde{\mu}(\mathcal{R}_i, \mathcal{R}_{i-1}, k+1)$$
6. $u^+ < a(\mathcal{R}_r)$.

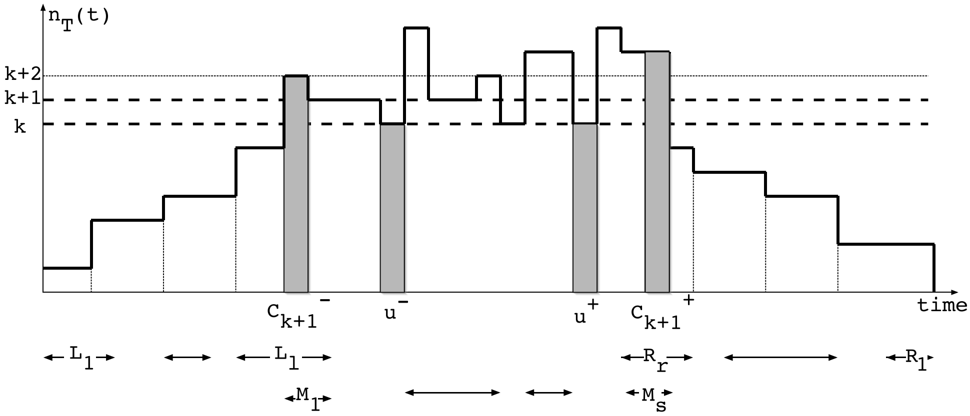

Fig. 12. The T-blocks families $\mathcal{L}$, $\mathcal{R}$,$\mathcal{M}$.

Using symmetrical arguments, we get from the time units C_p^- for $p \in \{1, \dots, k+1\}$ a family of T-blocks $\mathcal{L}_1, \dots, \mathcal{L}_l$ with the following properties:

1. $\mathcal{L}_1 \propto \cdots \propto \mathcal{L}_l$;
2. $\lambda(\mathcal{L}_l) \geq k+1$ (since $A_T(C_{k+1}^-) \subset \mathcal{L}_l$ and Property 7);
3. $\lambda(\mathcal{L}_1) < \cdots < \lambda(\mathcal{L}_l)$ (Property 8);
4. for any $p \in \{1, \dots, k+1\}$, C_p^- belongs to an interval $\mathcal{L}_{j(p)}$ such that $\lambda(\mathcal{L}_{j(p)}) \geq p$;
5. for any $i \in \{1, \dots, l-1\}$, we have:
$$\sum_{t \in Mid(\mathcal{L}_i, \mathcal{L}_{i+1})} n_T(t) = \lambda(\mathcal{L}_i)|Mid(\mathcal{L}_i, \mathcal{L}_{i+1})| \leq \tilde{\mu}(\mathcal{L}_i, \mathcal{L}_{i+1}, k+1)$$
6. $u^- > b(\mathcal{L}_l)$.

It comes from the definition of the T-block families $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{M}$ that:

1. for $i \in \{1, \dots, r-1\}$, and $j \in \{1, \dots, s\}$, we have $\mathcal{R}_i \cap \mathcal{M}_j = \emptyset$;
2. for $i \in \{1, \dots, l-1\}$, and $j \in \{1, \dots, s\}$, we have $\mathcal{L}_i \cap \mathcal{M}_j = \emptyset$.

Fig. 12 shows a possible configuration of these T-block families.

Let us consider the intersection graph of the T-blocks of the 3 families $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{R}$ and let us denote by $I_1, \dots, I_q$ the T-blocks associated with its connected components. These T-blocks may be indexed so that $\mathcal{I}_1 \propto \cdots \propto \mathcal{I}_q$ and it comes from the definition of $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{R}$ that:

1. for any $i \in \{1, \dots, q\}$, $\mathcal{I}_i \cap H = \emptyset$;
2. the time units of $\hat{S}$ as well as the time units C_p^+ and C_p^- (for any $p \in \{1, \dots, k+1\}$) belong to $\cup_{i \in \{1,\dots,q\}} \mathcal{I}_i$;
3. any $\mathcal{I}_i$ such that $u^- < a(\mathcal{I}_i) \leq b(\mathcal{I}_i) < u^+$ is one T-block of the $\mathcal{M}$ family.

Let us assume that $C_{k+1}^- \in \mathcal{I}_{\hat{l}}$ and $C_{k+1}^+ \in \mathcal{I}_{\hat{r}}$ $(1 \leq \hat{l} < \hat{r} \leq q)$. Again from the definition of the families $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{R}$, we know that:

1. for $i \in \{1, \dots, \hat{l}-1\}$, $\mathcal{I}_i$ belongs to the $\mathcal{L}$ family;
2. for $i \in \{\hat{r}+1, \dots, q\}$, $\mathcal{I}_i$ belongs to the $\mathcal{R}$ family;
3. for $i \in \{\hat{l}, \dots, \hat{r}\}$, we have $\lambda(\mathcal{I}_i) \geq k+1$ since
   - $A_T(C_{k+1}^-) \subseteq \mathcal{I}_{\hat{l}}$ and $\lambda(A_T(C_{k+1}^-)) \geq k+1$
   - $A_T(C_{k+1}^+) \subseteq \mathcal{I}_{\hat{r}}$ and $\lambda(A_T(C_{k+1}^+)) \geq k+1$
   - for $i \in \{\hat{l}+1, \dots, \hat{r}-1\}$, $\mathcal{I}_i$ is a T-block of the $\mathcal{M}$ family;
4. $\lambda(\mathcal{I}_{\hat{r}}) > \lambda(\mathcal{I}_{\hat{r}+1}) > \cdots > \lambda(\mathcal{I}_q)$ (from Property 8);
5. $\lambda(\mathcal{I}_1) < \lambda(\mathcal{I}_2) > \cdots > \lambda(\mathcal{I}_{\hat{l}})$ (from Property 8).

Let $i \in \{\hat{r}+1, \dots, q\}$ and let $p^+(i)$ be the largest number $p \in \{1, \dots, k\}$ such that $C_p^+ \in \mathcal{I}_i$. Then we derive from Property 7 that $\lambda(\mathcal{I}_i) = \lambda(A_T(C_{p^+(i)}^+))$ and that for any $t \in Mid(\mathcal{I}_{i-1}, \mathcal{I}_i)$, we have $n_T(t) = \lambda(\mathcal{I}_i)$. Similarly, for $i \in \{1, \dots, \hat{l}-1\}$, if $p^-(i)$ is the largest number $p \in \{1, \dots, k\}$ such that $C_p^- \in \mathcal{I}_i$, we derive that, $\lambda(\mathcal{I}_i) = \lambda(A_T(C_{p^-(i)}^-))$ and that for any $t \in Mid(\mathcal{I}_i, \mathcal{I}_{i+1})$, we have $n_T(t) = \lambda(\mathcal{I}_i)$. Thus the following inequalities are satisfied:

1. for any $i \in \{\hat{r}, \ldots, q-1\}$,
$$\sum_{t \in Mid(\mathcal{I}_i, \mathcal{I}_{i+1})} n_T(t) = |Mid(\mathcal{I}_i, \mathcal{I}_{i+1})| \times \lambda(\mathcal{I}_{i+1}) \leq \tilde{\mu}(\mathcal{I}_i, \mathcal{I}_{i+1}, k+1)$$
2. for any $i \in \{1, \ldots, \hat{l}-1\}$,
$$\sum_{t \in Mid(\mathcal{I}_i, \mathcal{I}_{i+1})} n_T(t) = |Mid(\mathcal{I}_i, \mathcal{I}_{i+1})| \times \lambda(\mathcal{I}_i) \leq \tilde{\mu}(\mathcal{I}_i, \mathcal{I}_{i+1}, k+1).$$

Moreover, we know that for $i \in \{\hat{l}, \ldots, \hat{r}-1\}$,
$$\sum_{t \in Mid(\mathcal{I}_i, \mathcal{I}_{i+1})} n_T(t) \leq (k+1) \times |Mid(\mathcal{I}_i, \mathcal{I}_{i+1})| \leq \tilde{\mu}(\mathcal{I}_i, \mathcal{I}_{i+1}, k+1).$$

Finally, let i^+ be such that $u^+ \in Mid(\mathcal{I}_{i^+}, \mathcal{I}_{i^++1})$. We have $n_T(u^+) = k$ and for any $t \in Mid(\mathcal{I}_{i^+}, \mathcal{I}_{i^++1})$, $n_T(t) \leq k+1$. So we derive that
$$\sum_{t \in Mid(\mathcal{I}_{i^+}, \mathcal{I}_{i^++1})} n_T(t) < (k+1)|Mid(\mathcal{I}_{i^+}, \mathcal{I}_{i^++1})| \leq \tilde{\mu}(\mathcal{I}_{i^+}, \mathcal{I}_{i^++1}, k+1).$$

Since $I_1, \ldots, I_q$ are T-blocks, we have
$$\sum_{i=1}^{q-1} \sum_{t \in Mid(\mathcal{I}_i, \mathcal{I}_{i+1})} n_T(t) = |J \setminus (\cup_{i=1}^{q} J(\mathcal{I}_i))| < \sum_{i=1}^{q-1} \tilde{\mu}(\mathcal{I}_i, \mathcal{I}_{i+1}, k+1).$$

The sequence $\mathcal{I}_1, \ldots, \mathcal{I}_q$ thus matches the required conditions to prove the "if" part of the theorem. □

5. Conclusion

In this paper, we have studied a variant of the basic homogeneous m-machine non-idling problem where weakly dependent unit-time jobs have to be scheduled within their time windows so that the non-idling constraint must be satisfied not only for each machine but for every subset of machines. A structural necessary and sufficient condition for an instance to be feasible has been provided. In the near future, a new paper will present a polynomial algorithm based on that characterization together with new results concerning the extension of the problem where a time dependent cost is associated with the processing of each job. As an important open question, it remains quite important, to our point of view, to get the complexity status of the non-homogeneous variant of the problem (more frequent in applications) when the non-idling constraint has only to be met on each machine or on a given subset of machines.

References

[1] P. Baptiste, Scheduling Unit Tasks to Minimize the Number of Idle Periods: a Polynomial Time Algorithm for Off-line Dynamic Power Management, Research Report, Laboratoire d'Informatique CNRS LIX, 2005.
[2] J. Carlier, A. Moukrim, K. Guédira, Exact resolution of the ne-machine sequencing problem with no machine idle time, Comput. Ind. Eng. 59 (2) (2010) 193–199.
[3] P. Chrétienne, On single-machine scheduling without intermediate delays, Discrete Appl. Math. 13 (156) (2008) 2543–2550.
[4] P. Hall, On representatives of subsets, J. Lond. Math. Soc. 10 (1) (1935) 26–30.
[5] S. Irani, K. Pruhs, Algorithmic Problems in Power Management, vol. 36, ACM Press, New York, USA, 2005, pp. 63–76.
[6] A. Jouglet, Single-machine scheduling with no idle time and release dates to minimize a regular criterion, J. Scheduling 15 (2) (2012) 217–238.
[7] I. Kacem, H. Kellerer, Approximation algorithms for no idle time scheduling on a single machine with release times and delivery times, Discrete Appl. Math. (2011). http://dx.doi.org/10.1016/j.dam.2011.07.005.
[8] K. Landis, Group Technology and Cellular Manufacturing in the Westvaco Los Angeles VH Department, Project Report in IOM 581, School of Business, University of Southern california, 1993.
[9] J.K. Lenstra, A.H.G. Rinnoy Kan, Complexity of scheduling under precedence constraints, Oper. Res. 26 (1) (1978) 22–35.
[10] M. Middendorf, V.G. Timkovsky, Transversal graphs for partially ordered sets: sequencing, merging and scheduling problems, J. Comb. Optim. 3 (4) (1999) 417–435.
[11] J.M.S. Valente, R.A.F.S. Alves, An exact approach to early/tardy scheduling with releases dates, Comput. Oper. Res. 32 (2005) 2905–2917.