

POLYHEDRAL REFORMULATION OF A SCHEDULING PROBLEM AND RELATED THEORETICAL RESULTS

JEAN DAMAY¹, ALAIN QUILLIOT¹ AND ERIC SANLAVILLE¹

Abstract. We deal here with a scheduling problem **GPPCSP** (*Generalized Parallelism and Preemption Constrained Scheduling Problem*) which is an extension of both the well-known *Resource Constrained Scheduling Problem* and the *Scheduling Problem with Disjunctive Constraints*. We first propose a reformulation of **GPPCSP**: according to it, solving **GPPCSP** means finding a vertex of the *Feasible Vertex Subset* of an *Antichain Polyhedron*. Next, we state several theoretical results related to this reformulation process and to structural properties of this specific *Feasible Vertex Subset* (connectivity, ...). We end by focusing on the preemptive case of **GPPCSP** and by identifying specific instances of **GPPCSP** which are such that any vertex of the related *Antichain Polyhedron* may be projected on its related *Feasible Vertex Subset* without any deterioration of the makespan. For such an instance, the **GPPCSP** problem may be solved in a simple way through linear programming.

Keywords. Partially ordered sets, hypergraphs, linear programming, polyhedra, multiprocessor scheduling, resource constrained project scheduling problem.

Mathematics Subject Classification. 90B35.

INTRODUCTION

We deal here with a scheduling problem **GPPCSP** (*Generalized Parallelism and Preemption Constrained Problem*), which extends the well-known **RCPSP** (*Resource Constrained Project Scheduling Problem*) and which includes *generalized*

Received December 1st, 2005. Accepted November 11, 2007.

¹ LIMOS, UMR CNRS 6158, Bât ISIMA, Université Blaise Pascal, Campus des Cézeaux, BP 125, 63173 Aubière, France; {jean.damay, alain.quilliot, eric.sanlaville}@isima.fr

parallelism constraints as well as specific *preemption constraints*. An instance of such a problem is defined by a task set X , a *generalized parallelism relation* E , a precedence constraint ρ , a duration function d and a *preemption function* P . We talk here about *generalized parallelism constraints*, since the compatibility relation E will enable us to describe any kind of subset made with tasks which are allowed to be simultaneously run (while **RCPSP** disjunctive constraints are usually related to limited amounts of renewable resources). The triple $H = (X, E, \rho)$ defines what we call an *ordered hypergraph*. Our purpose is to show how using the hypergraph and the partially ordered set formalism may yield a polyhedral reformulation of this problem, and then open new paths for algorithm design.

Partially ordered (Poset) sets and hypergraphs are among the combinatorial tools which most often appear inside Combinatorial Optimization models. Partially ordered sets have a large range of applications, many of them related to planning and scheduling (see [3,4,7,13,20,26,34,37]). Hypergraphs are essentially involved in the combinatorial analysis of global constraints (see [44]), which may appear in information storage and retrieval models (see [8,24,30,32]), of timetabling and scheduling problems (see [1,14]), of resource allocation problems or even of some genetics or archeology problems (see [7,28]).

We first associate, with any instance $(H = (X, E, \rho), d, P)$ of the **GPPCSP** Problem, a specific *Antichain Polyhedron* $\Lambda_{H,d}$. Such a polyhedron was first introduced by Sauer and Stone (see [40,41]), Papadimitriou and Yannanakis (see [38]), in the context of non preemptive multiprocessor scheduling problems [37], and has next been used (see for instance [6,11,12,16,26,35]) in order to get theoretical bounds for the *Resource Constraint Project Scheduling problem*.

Next, by involving specific combinatorial properties like the *Extended Interval Representability Property* (see [10]), we show how it is possible to reformulate a **GPPCSP** instance as a search problem defined on some specific feasible subset $V - PR_{H,d}$ (preemptive), $V - NPR_{H,d}$ (non preemptive) or $V - PPR_{H,d,P}$ (partially preemptive) of the vertex set $V(\Lambda_{H,d})$ of the polyhedron $\Lambda_{H,d}$. We study the connectivity (in the sense of the natural adjacency relation which is defined on the vertex set of any polyhedron), of those subsets, and we briefly discuss the way some local search procedures might be derived from these theoretical results.

Finally we focus on the **preemptive** version of the **GPPCSP** Problem. Preemptive problems have been less studied than non preemptive ones. Still, they are related to many practical applications (multiprocessor scheduling, grid computing, truck fleet planning). They also may be viewed as relaxations of non preemptive problems. Thus, capturing some of their combinatorial properties may help in designing solutions for non preemptive problems and for mixed problems. What we mainly do here is to extend a previous result of Moukrim and Quilliot (see [33]) related to the **Multiprocessor Scheduling Problem**: we identify cases when any vertex of the polyhedron $\Lambda_{H,d}$ may be projected, without any deterioration of the makespan, on the feasible preemptive subset $V - PR_{H,d}$. In such a case, solving the related makespan minimization preemptive scheduling problem can be done in a simple way by solving a linear program on the polyhedron $\Lambda_{H,d}$.

1. THE GPPCSP PROBLEM AND THE RELATED ANTICHAIN POLYHEDRON

1.1. TEMPORAL PHASES, SCHEDULES, PREEMPTIVE NUMBERS

If $I = [\alpha, \beta]$ is a closed interval of the real line $[0, +\infty[$, we say that α is the *start-point* of I and that β is the *end-point* of I . If $J = \{[\alpha_0, \beta_0], [\alpha_1, \beta_1], \dots, [\alpha_n, \beta_n]\}$ is a finite family of disjoint closed intervals of the real line $[0, +\infty[$, such that $\alpha_0 < \beta_0 < \alpha_1 < \beta_1 < \dots < \alpha_n < \beta_n$, then we say that J is a *temporal phasis*, with *preemption number* $NP(J)$ equal to n , with *start-point* $START(J)$ equal to α_0 and with *end-point* $END(J)$ equal to β_n . The *length* of J is the sum $\sum_{i=0..n} (\beta_i - \alpha_i)$.

If X is a task set, then a *schedule* of X is a function Φ which associates, with any task x in X , a temporal phasis $\Phi(x)$. The real line $[0, +\infty[$ is also called the *Time Space*.

If Φ is such a schedule, and if t is a value (an instant) in the Time Space $[0, +\infty[$, we denote by $e_\Phi(t)$ the task subset $e_\Phi(t) = \{x \in X \text{ such that } t \text{ belongs to the interior } \Phi(x)^0 \text{ of } \Phi(x)\}$ which is defined by the tasks which are simultaneously run at instant t . We call *Makespan* of Φ the quantity $\text{Max}_{x \in X} \text{END}(\Phi(x))$.

We denote by $\text{SCH}(X)$ the set of all the schedules of X .

1.2. THE GPPCSP SCHEDULING PROBLEM

A **GPPCSP problem** instance (*Generalized Parallelism and Preemption Constrained Scheduling Problem*) is defined by:

- a task set X ;
- E is a subset family (edge set) of X , called *generalized parallelism relation*;
- a binary precedence relation ρ , defined on X ;
- a duration function d , defined from X to the set Q of the positive rational numbers;
- a preemption threshold function P , defined from X to the set \mathbb{N} of the integral numbers;
- a cost real valued function $Cost$ which is defined on the Schedule Set $\text{SCH}(X)$.

Remark 1.1. The subset family E provides us with a description, which is eventually implicit, of the tasks which are allowed to be simultaneously run, and thus defines what we shall call the *Generalized Parallelism Constraint*. The preemption function P tells us how many times a given task may be interrupted during its execution, and thus defines what we shall call the *Partial Preemption Constraint*. Practically, partial preemption constraints may be found when human resources are involved (timetabling problems).

Solving **GPPCSP** on the above instance means computing a schedule Φ of X , in such a way that:

- for any x in X , the length of $\Phi(x)$ is equal to the duration $d(x)$ of x ; (*Duration Constraint C1*)
- for every task x in X , the preemption number $\text{NP}(\Phi(x))$ of the temporal phasis $\Phi(x)$ does not exceed the preemption threshold $P(x)$; (*Partial Preemption Constraint C2*)
- if $x, y \in X$ are such that $x \rho y$, then the start-point $\text{START}(\Phi(y))$ of $\Phi(y)$ is at least equal to the end-point $\text{END}(\Phi(x))$ of $\Phi(x)$; (*Precedence Constraint C3*)
- for any positive real number t , the subset $e_\Phi(t) = \{x \in X \text{ such that } t \text{ belongs to the interior } \Phi(x)^0 \text{ of } \Phi(x)\}$, belongs to the edge set E ; (*Generalized Parallelism Constraint C4*)
- Φ minimizes some quantity $\text{Cost}(\Phi)$.

A schedule Φ which satisfies C1, C2, C3 and C4 is a *feasible schedule* for the related instance of **GPPCSP**.

If the preemption function P is constant and equal to 0, then we say that our instance of **GPPCSP** is non *preemptive*. If P is constant and equal to $+\infty$, then we say that our instance of **GPPCSP** is *fully preemptive*.

If $P = 0$ (resp. $+\infty$), if running any task x in X requires the use of some machine, and if the relation E is defined by: $e \in E$ iff $\text{Card}(e) \leq k$, where k is the number of existing machines, then our **GPPCSP** instance is an instance of the non preemptive (resp. preemptive) **Multiprocessor Scheduling Problem**, (see [19,20,36,37,41,42]).

If $P = 0$ (resp. $+\infty$), if the execution of every task x in X requires the use of some resource amount vector $r(x)$ and if the relation E is defined by: $e \in E$ iff $\sum_{x \in e} r(x) \leq R$, where R is the existing resource amount vector, then we see that our instance of **GPPCSP** is an instance of the non preemptive (resp. preemptive) **Resource Constrained Project Scheduling Problem** (RCPSP), (see [5–7,12,16,26,27,29,33]).

Let us see now an example of a **GPPCSP** instance.

Example 1.2. Let us set $X = \{A, B, C, D, E\}$.

We suppose that the duration function d and the preemption threshold P are given by the following table:

Task x	A	B	C	D	E
$d(x)$	2	2	2	1.5	1
$P(x)$	0	0	1	0	0

We suppose that the precedence relation ρ is defined by: $A \rho D$.

We suppose that the generalized parallelism relation E is defined as follows:

- D and C are not allowed to be simultaneously performed, *i.e.* the pair $\{C, D\}$ does not belong to E ;

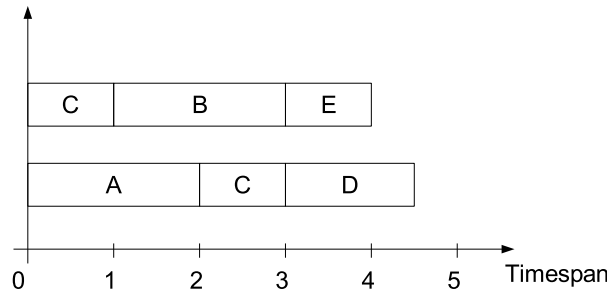
- Every task x in X requires, at any time during its execution, some amount $r(x)$ of a given resource, according to the following table, while the available amount of existing resource is equal to 4.

Task x	A	B	C	D	E
$r(x)$	2	2	2	1	3

It comes that a subset e of X belongs to E iff: it does not contain the pair $\{C, D\}$ and $\sum_{x \in e} r(x) \leq 4$.

We notice that the relation E defined this way is defined in an **implicit way**. If we explicit it, we find the edges $e_0 = \{Nil\}, e_1 = \{A\}, e_2 = \{B\}, e_3 = \{C\}, e_4 = \{D\}, e_5 = \{E\}, e_6 = \{A, B\}, e_7 = \{A, C\}, e_8 = \{B, C\}, e_9 = \{A, D\}, e_{10} = \{B, D\}, e_{11} = \{C, D\}, e_{12} = \{D, E\}$.

If the function $Cost$ is the Makespan function, then an optimal solution of **GPPCSP** may be described as below:



Remark 1.3. The **GPPCSP** is an extension of the **RCPSP** (*Resource Constrained Project Scheduling Problem*) Problem in two senses:

- generalized disjunctive constraints are allowed, whose expression comes through the edge set E , and which do not involve renewable resources;
- preemption and non preemption characteristics are mixed into a unique formalism, which allows us to simultaneously deal with preemptive and non preemptive tasks, while imposing conditions on the way a preemptive task may be divided into subtasks. As a matter of fact, the preemptive version of **RCPSP** Problem has been fairly less studied than the non preemptive one: algorithmic studies involving preemption have mainly been limited to the case of multiprocessor scheduling (see Djellab [19], Josefowska *et al.* [26]).

Definition 1.4 (left (right) translation of a schedule and dominant schedules). During the next sections, we shall try to capture some structural properties of the optimal solutions of the **GPPCSP** Problem. In order to do it, we shall need some tools which will allow us to modify feasible solutions in an elementary way. This will provide the motivation for the following definitions related to translations and domination. Let Φ and Φ' two feasible schedules related to a given instance (X, E, ρ, d, P) of the **GPPCSP** Problem:

- we say that Φ' comes from Φ through *left (right) translation* if for every task x in X , there exists $\delta > 0$ such that the temporal phasis $\Phi'(x)$ comes from the temporal phasis $\Phi(x)$ through a left (right) translation with parameter δ ;
- we say that Φ' *dominates* Φ if for every task x in X , it is possible to write:

$$\Phi(x) = \bigcup_{i=1..n} [\alpha_i, \beta_i] \text{ and } \Phi'(x) = \bigcup_{i=1..n} [\alpha'_i, \beta'_i]$$

in such a way that:

- for every $i = 1..n$, $[\alpha_i, \beta_i]$ and $[\alpha'_i, \beta'_i]$ have equal lengths;
- for every i, j in $1..n$, $[\alpha_i, \beta_i]$ and $[\alpha_j, \beta_j]$ ($[\alpha'_i, \beta'_i]$ and $[\alpha'_j, \beta'_j]$) do not intersect by more than one point;
- for every $i = 1..n$, $\alpha_i \leq \alpha'_i$.

For most *Cost* functions (and not only if *Cost* is the makespan function), it will happen that if Φ' dominates Φ then we shall have $\mathbf{Cost}(\Phi') \leq \mathbf{Cost}(\Phi)$.

Φ is said to be *dominant* if no feasible schedule Φ' dominates Φ but itself.

1.3. ORDERED HYPERGRAPHS

A triple $H = (X, E, \rho)$ as in Section 1.2, that means such that X is some finite (we call it *vertex set* of H) set, E is some (we call it the *edge set* of H) subset family of X , and ρ (we call it the *partial ordering* of H) is some binary relation defined on X , is called an *Ordered Hypergraph*.

We say that H is *monotonic*, if the two following statements are true:

- if $e \in E$ is some edge of H , then any subset of e is also in E ;
- for any $x \in X$, the singleton subset $\{x\}$ is in E .

A sequence $\{x_0..x_n = x_0\}$ of vertices of H such that $x_0 \rho x_1.. \rho x_n = x_0$, will be called a *circuit* for the relation ρ . We say that H is *no-circuit* if there does not exist any circuit in X for the relation ρ .

An instance of **GPPCSP** is then defined by a **monotonic no-circuit ordered hypergraph** $H = (X, E, \rho)$, by a duration function d , and by a preemption threshold function P . Given such an instance, one may notice that the information which is provided by the precedence relation ρ may restrict the parallelism opportunities allowed by the description of E : for instance, an edge e of E may contain 2 tasks x and y such that $x \rho y$. This leads us to define ρ -antichains and valid antichains as follows:

- $H = (X, E, \rho)$ being a monotonic no-circuit ordered hypergraph, we denote by ρ_T the transitive closure of ρ , and we call ρ -antichain of H any subset B of X which satisfies: there does not exist x, y in B such that $x \rho_T y$;
- we say that such a ρ -antichain B is a *valid antichain* of the ordered hypergraph H if it belongs to the edge set E . We denote by E_ρ the set of

all valid antichains of H , and we say that E_ρ is the *Valid Antichain Edge Set* associated with H .

1.3.1. Dual hypergraph

Let us suppose that $H = (X, E, \rho)$ is a monotonic no-circuit ordered hypergraph. Then, we may define on the Valid Antichain Edge Set E_ρ , considered as a vertex set, an ordered hypergraph structure $K_H = (E_\rho, F, \tau)$ as follows:

- if e and $e' \in E_\rho$, then we define the binary relation τ by setting: $e \tau e'$ iff there exists $x \in e$ and $y \in e'$ such that $x \rho y$;
- if $x \in X$, then we set $F_x = \{e \in E_\rho \text{ such that } x \in e\}$. We define the *Edge Family* F of K_H by setting $F = \{F_x, x \in X\}$.

The ordered hypergraph $K_H = (E_\rho, F, \tau)$ which is defined this way, is called the *Dual Ordered Hypergraph* of H . If B is some subset of E_ρ , then we denote by $K_H(B)$ the ordered subhypergraph which is induced from K_H by B .

Example 1.5. Let us keep on with example 1.2 in Section 1.2. Then:

- $E_\rho = \{e_0 = \{Nil\}, e_1 = \{A\}, e_2 = \{B\}, e_3 = \{C\}, e_4 = \{D\}, e_5 = \{E\}, e_6 = \{A, B\}, e_7 = \{A, C\}, e_8 = \{B, C\}, e_{10} = \{B, D\}, e_{11} = \{C, D\}, e_{12} = \{D, E\}\}$;
- we have: $e_i \tau e_j$ every time e_i contains A and e_j contains D ;
- $F_A = \{e_1, e_6, e_7\}$, $F_B = \{e_2, e_6, e_8, e_{10}\}$, $F_C = \{e_3, e_7, e_8, e_{11}\}$, $F_D = \{e_4, e_{10}, e_{11}, e_{12}\}$, $F_E = \{e_5, e_{12}\}$.

We notice that K_H , defined according to this example, is not monotonic. Still, it is no-circuit. But, if we augment the relation ρ by setting $E \rho B$, then K_H contains the circuit (e_6, e_{12}, e_6) .

1.3.2. Interval representations of an ordered hypergraph

As we shall see further, solving **GPPCSP** will mean identifying valid antichains made with tasks which are effectively going to be simultaneously run, and sequencing them in such a way that they define a schedule compatible with both precedence constraints (C3) and partial preemption constraints (C2). Formalizing the induced sequencing problem requires us to define the *L-Interval Representability* of an ordered hypergraph as follows:

If π is a linear ordering of a set Z , then we say that $x_0, \dots, x_k \in Z$ are *consecutive according to π* if $x_0 \pi x_1 \pi \dots \pi x_{k-1} \pi x_k$ and if there does not exist $z \in Z$ and $i \in \{0..k-1\}$ such that $x_i \pi z \pi x_{i+1}$.

Given an ordered hypergraph $K = (Z, F, \sigma)$, together with some function L , which, to any e in F , makes correspond $L(e) \in \{0, 1, \dots, n, \dots + \infty\}$. We say that such a function L is an *edge-preemption threshold function* for K . K is said to be *L-interval representable* if there exists a linear ordering σ^* of Z (*L-Interval Representation*) such that:

- σ^* is an extension of σ , which means that any pair $x, y \in Z$ such that $x \sigma y$ is also such that $x \sigma^* y$;

- any edge e in F may be written $e = e_0 \cup \dots \cup e_{l(e)}$, where the $e_i, i = 0..l(e)$ are disjoint subsets of Z , in such a way that $l(e) \leq L(e)$ and that the elements of any $e_i, i = 0..l(e)$, are consecutive according to the linear ordering σ^* : we say that σ^* makes the elements of e be *consecutive modulo* $L(e)$.

In case $L = 0$, we only say that K is *interval representable*.

Interval structures (graphs, hypergraphs, partially ordered sets) have already been studied in several ways (see [7,8,10,21,30,40]). For instance, Booth and Lueker, (see [10]), got an efficient algorithmic polynomial characterization of *interval hypergraphs* (the case when σ is empty and $L = 0$), based upon the use of PQ-trees. Also Quilliot and Xiao proposed in [40], a characterization through forbidden patterns of interval representable ordered hypergraphs. Still, characterizing the L -interval representability property in the general case remains an open problem.

Example 1.6. Let us suppose that $Z = \{A, B, C\}$, that $F = \{e \in Z \text{ such that } \text{Card}(e) = 2\}$, and that σ is the empty relation. Then we see that $K = (Z, F, \sigma)$ is not interval representable. Still, it becomes L -interval representable if L is defined by: $L(\{A, B\}) = L(\{A, C\}) = 0$ and $L(\{B, C\}) = 1$. In order to check this last point, we may consider that B, A, C are consecutive according to σ^* . Then we get that the elements of $\{A, B\}$ and $\{B, C\}$ are consecutive according to σ^* , while $\{B, C\}$ may be written $\{B, C\} = \{B\} \cup \{C\}$.

Since we want to use the notion of *L-Interval Representability* in order to properly sequence specific valid antichains, we will only apply it here to the Dual Hypergraph K_H defined in Section 1.3.1.

1.4. EXPRESSION OF A GPPCSP SCHEDULE AS A WEIGHTED VALID ANTICHAIN SEQUENCE

Our purpose is now to link the ordered hypergraph formalism to the **GPPCSP** Problem in order to get a reformulation of this problem which might open the way to new algorithmic approaches. In this section, as well as in Sections 1.5 and 1.6, we are going to explain how feasible **GPPCSP** schedules can be related to specific weighted valid antichain sequences and to specific vertices of an *antichain polyhedron*.

So, let us consider a feasible schedule Φ of our **GPPCSP** instance $H = (X, E, \rho, d, P)$. We may associate with Φ some finite sequence U of elements of E_ρ , according to the procedure **ANTICHAIN-SEQ** of algorithm 1.

The *weighted sequence* U is such that:

- several elements of U may correspond to a same antichain e in E_ρ ;
- for any $x \in X$, the sum $\sum_{u \in U/x \in u} \eta(u)$ is equal to $d(x)$; (*Duration Constraint C5*)
- if u and $u' \in U$, $x \in u, y \in u'$ are such that $x \rho y$, then u precedes u' in U ; (*Precedence Constraint C6*);
- for any $x \in X$, the subsequence $U(x)$ of U which is defined by the valid antichains of U which contain x , may be written as an union of pairwise

Algorithm 1 ANTICHAIN-SEQ Procedure

- 1: Build the set Δ of all end-points and start-points of the intervals which are involved in the description of the temporal phases $\Phi(x), x \in X$, as unions of pairwise disjoint closed intervals of the Time Space $[0, +\infty[$; Add the point 0 to Δ ; Δ may be written as a sequence $\Delta = \{\delta_0.. \delta_n\}$, in such a way that $0 = \delta_0 < \delta_1 < \dots < \delta_{n-1} < \delta_n$
 - 2: **for** $i = 0..n - 1$ **do**
 - 3: set $u_i = \{x \in X \text{ such that }]\delta_i, \delta_{i+1}[\subseteq \Phi(x)\}$, and set $\eta(u_i) = \delta_{i+1} - \delta_i$ {for any $i = 0..n - 1, u_i$ is a valid antichain and belongs to E_ρ }
 - 4: **end for**
 - 5: Set $U =$ the sequence $\{u_0, \dots, u_{n-1}\}$: U is weighted by the weight sequence $\eta(u_i), i = 0..n - 1$
-

disjoint subsequences $U_0..U_k$ of U , in such a way that: (*Partial Preemption Constraint C7*)

- for any $j = 1..k$, the elements of U_j are consecutive in U ;
- $k \leq P(x)$.

Clearly, any weighted sequence U which satisfies C5, C6 and C7 defines a feasible schedule for our **GPPCSP** problem, whose makespan is equal to the quantity $\sum_{u \in U} \eta(u)$, which is called the makespan of U . As a matter of fact, we shall indistinctly talk about a feasible schedule in the case of such a weighted sequence U and in the case of a schedule Φ which satisfies the constraints C1, C2, C3 and C4 of Section 1.2.

Example 1.7. Let us come back to our **GPPCSP** instance of example 1.2 in Section 1.2. Applying the **ANTICHAIN-SEQ** procedure to the feasible schedule which was then proposed yields the sequence $\Delta = (0, 1, 2, 3, 4, 4.5)$ and the weighted sequence $U = ((e_7, \eta(e_7) = 1), (e_6, \eta(e_6) = 1), (e_8, \eta(e_8) = 1), (e_{12}, \eta(e_{12}) = 1), (e_4, \eta(e_4) = 1/2))$.

1.5. THE ANTICHAIN POLYHEDRON $\Lambda_{H,d}$

Let us keep on with our instance $(H = (X, E, \rho), d, P)$ of the scheduling problem **GPPCSP**. We call *Antichain Polyhedron* defined by the ordered hypergraph $H = (X, E, \rho)$ and by the duration function d , the set of all feasible solutions $z = (z_e, e \in E) \geq 0$ of the following linear constraint system $\Lambda_{H,d}$:

$$\Lambda_{H,d} : \left\{ z = (z_e, e \in E_\rho) \geq 0, \text{ such that : for any } x \in X, \sum_{e/x \in e} z_e = d(x) \right\}.$$

As a matter of fact, we denote by $\Lambda_{H,d}$ both the polyhedron and its related linear constraint system.

Let Φ be some feasible schedule of the instance of the **GPPCSP** problem which is defined by the hypergraph H , the duration function d and the preemption

threshold function P , and let $U = \{u_i, \text{ with weight } \eta(u_i), i = 0..n - 1\}$, the weighted valid antichain sequence which derives from Φ through application of the **ANTICHAIN-SEQ** procedure of algorithm 1. Then, for every valid antichain $e \in E_\rho$, we may set $z_e = \sum_{i=0..n-1/e=u_i} \eta(u_i)$. In case any valid antichain $e \in E_\rho$ occurs at most one time in the sequence U , then this formula may be rewritten in a simpler way: $z_e = \eta(u_i)$ if there exists $i \in 0..n - 1$ such that $u_i = e$ and $z_e = 0$ otherwise. We define by this way a vector $z = (z_e, e \in E_\rho) \geq 0$, **which belongs to the polyhedron $\Lambda_{H,d}$** . We say that z derives from the feasible schedule Φ . We say that the valid antichain e is *active for the schedule Φ* if $z_e \neq 0$.

Example 1.8. If we refer to the specific **GPPCSP** instance of Example 1.2 in Section 1.2, we see that the vector z defined by $z_{e_0} = 0, z_{e_1} = 0, z_{e_2} = 0, z_{e_3} = 0, z_{e_4} = 1/2, z_{e_5} = 0, z_{e_6} = 1, z_{e_7} = 1, z_{e_8} = 1, z_{e_{10}} = 0, z_{e_{11}} = 0, z_{e_{12}} = 1$, derives from the previously proposed optimal schedule Φ . The related active antichains are e_4, e_6, e_7, e_8 and e_{12} .

The polyhedron $\Lambda_{H,d}$ is a convex closed subset of the space $\mathbb{R}^{\text{Card}(E_\rho)}$. In case E is defined in an implicit way, $\text{Card}(E_\rho)$ may be very large. Its intersection with the hyperspace defined by $z_{Nil} = 0$, where Nil is the empty edge, is compact. Also, all *vertices* of $\Lambda_{H,d}$ are located in this hyperspace $z_{Nil} = 0$. If $z \in \Lambda_{H,d}$, then we say that an edge $e \in E_\rho$ is *active* for z if $z_e \neq 0$. We denote by $ACT(z)$ the set of all edges $e \in E_\rho$ which are active for z .

Let us denote by M_H the constraint $\{0, 1\}$ -matrix associated with $\Lambda_{H,d}$. The rows and the columns of M_H may respectively be identified with the vertices and with the valid antichains of H , or, equivalently, with the edges and the vertices of the dual ordered hypergraph $K_H = (E_\rho, F, \tau)$ of H . Then $\Lambda_{H,d}$ may also be rewritten:

$$\Lambda_{H,d} = \{z = (z_e, e \in E_\rho) \geq 0, \text{ such that } M_H.z = d\}.$$

1.5.1. The vertex set $V(\Lambda_{H,d})$ of $\Lambda_{H,d}$

We denote by $V(\Lambda_{H,d})$ the set of the vertices of the polyhedron $\Lambda_{H,d}$. If $B \subset E_\rho$ and $Y \subset X$, then we denote by $M_H(B, Y)$ the submatrix of the matrix M_H which is defined by B and Y . The following lemma will help us in reformulating the **GPPCSP** problem as a search problem involving specific basis subsets of E_ρ .

Lemma 1.9 (see for instance [43], Chap. 8). *A vertex of $\Lambda_{H,d}$ is a vector $z \geq 0$ such that there exists a subset B of E_ρ , such that:*

$$\text{Card}(B) = \text{Rank}(M_H) = \text{Card}(X),$$

and which satisfies the following relations: *(Basis Subset Equations)*

- the restriction $z^{E_\rho - B}$ of z to the edges which are not in the column set $B \subset E_\rho$, is null;
- the submatrix $M_H(B, X)^{-1}$ is invertible;
- the restriction z^B of z to B is equal to $M_H(B, X)^{-1}.d$.

If z is such a vertex of $\Lambda_{H,d}$, and if B is as in Lemma 1.9, then we say that the column subset $B \subset E_\rho$ is a *basis subset associated* with the vertex z , and that z *derives* from this *basis subset* B . Lemma 1.9 will help us in reformulating the **GPPCSP** problem as a search problem involving specific basis subsets of E_ρ .

Example 1.10. If we refer to our Example 1.2 in Section 1.2, we see that the vector z defined by $z_{e_0} = 0, z_{e_1} = 0, z_{e_2} = 0, z_{e_3} = 0, z_{e_4} = 1/2, z_{e_5} = 0, z_{e_6} = 1, z_{e_7} = 1, z_{e_8} = 1, z_{e_{10}} = 0, z_{e_{11}} = 0, z_{e_{12}} = 1$, which derives from the optimal schedule Φ which was then proposed, is a vertex of the related polyhedron $\Lambda_{H,d}$, with associated basis $B = \{e_4, e_6, e_7, e_8, e_{12}\}$.

1.5.2. Adjacency relation on the vertex set $V(\Lambda_{H,d})$

There exists a natural adjacency relation which is defined on the vertex set of any polyhedron. We know that:

Lemma 1.11 (see for instance [43], Chap. 8). *Two distinct vertices z and z' in $V(\Lambda_{H,d})$ are **adjacent** in the sense of this relation if and only if they admit two associated basis subsets B and B' , such that the cardinality of the symmetric difference $(B \Delta B')$ is equal to 2.*

This lemma will help us in studying the connectivity of the specific vertex subsets of $V(\Lambda_{H,d})$ which will be involved in the reformulation of the **GPPCSP** problem.

1.5.3. Using the $\Lambda_{H,d}$ polyhedron in order to get lower bounds for the linear GP-PCSP problem

In case the performance criterion of our **GPPCSP** instance involves a specific cost vector $\mathbf{C} = (C_e, e \in E_\rho)$, in such a way that $C_e \cdot \delta$ expresses the cost of simultaneously running the tasks of the edge e in E_ρ during some time δ , then $\mathbf{Cost}(\Phi)$ may be written $\mathbf{Cost}(\Phi) = \mathbf{C} \cdot z$, and we say that the related **GPPCSP** instance is *linear*. Of course, the case when the performance criterion is the makespan ($\mathbf{C} = 1$) corresponds to such a linear instance.

In such a case, solving the following linear program $L_{H,d}^C$:

$$L_{H,d}^C : \{ \text{Compute } z \text{ in } \Lambda_{H,d} \text{ which minimizes the quantity } \mathbf{Cost}(z) = \mathbf{C} \cdot z \}$$

through column generation techniques provides a lower bound of the related **GP-PCSP** problem.

This idea of using the $\Lambda_{H,d}$ polyhedron in order to get lower bounds for a scheduling problem was first introduced in [40,41] for the Multiprocessor Scheduling Problem, and was next used in [5,6,11,12,27] for the Resource Constrained Project Scheduling Problem. More precisely, Sauer and Stone [40,41], Papadimitriou, Yannanakis [38], and next Moukrim, Quilliot [36], used it in order to state conditions which make the optimal value of $L_{H,d}^1$ be equal to the optimal value of a related multiprocessor scheduling problem. Djellab [19], and next Josefowska *et al.* [26], used clever adaptations of the linear program $L_{H,d}^1$ in order to get lower and

upper bounds for preemptive multiprocessor scheduling problems. Those adaptations, which yield very good results, involve MAC NAUGHTON's Algorithm (see [4]) for preemptive multiprocessor scheduling without precedence constraint. Also, Baptiste and Demassey [6], used constraint propagation techniques in order to provide the linear program $L_{H,d}^1$ with additional constraints (cuts) and this allowed them to get tight bounds for the non preemptive **RCPSP**, giving rise by this way to very efficient branch/bound algorithms.

Example 1.12. If we refer to the specific **GPPCSP** instance of Example 1.2 in Section 1.2, then we see that the vector z defined by $z_{e_0} = 0, z_{e_1} = 0, z_{e_2} = 0, z_{e_3} = 0, z_{e_4} = 1/2, z_{e_5} = 0, z_{e_6} = 1, z_{e_7} = 1, z_{e_8} = 1, z_{e_{10}} = 0, z_{e_{11}} = 0, z_{e_{12}} = 1$, is an optimal solution of the linear program $L_{H,d}^1$, and provides us with a lower bound equal to 4.5.

1.6. SPECIFIC VERTEX SUBSETS OF THE VERTEX SET $V(\Lambda_{H,d})$

Of course, not every vector z in the polyhedron $\Lambda_{H,d}$ derives from some feasible schedule of the **GPPCSP** instance which is defined by the ordered hypergraph H , by the duration function d and by the preemption threshold function P . We can see it by considering Example 1.2 in Section 1.2, and by considering the vector z which is defined by

$$\begin{aligned} \text{for } i = 1..5, z_{e_i} = 1/2, z_{e_6} = 3/4, z_{e_7} = 3/4, z_{e_8} \\ = 1/2, z_{e_{10}} = 1/4, z_{e_{11}} = 1/4, z_{e_{12}} = 1/2. \end{aligned}$$

As a matter of fact, we notice that:

Proposition 1.13 (derivability characterization). *Given an ordered hypergraph $H = (X, E, \rho)$, a duration function d , a preemption threshold function P , and a vector z in the polyhedron $\Lambda_{H,d}$.*

*The vector z corresponds to a feasible schedule of the **GPPCSP** instance defined by H, d and P , if and only if the ordered subhypergraph $K_H(\text{ACT}(z))$ of the dual hypergraph K_H , which is induced by the subset $\text{ACT}(z)$ of E_ρ made with the edges e which are active for z , is P -interval representable.*

Proof. It is only a matter of applying the definitions. □

Since we are motivated here by the idea of finding tools which would allow us to deal with elements of $\Lambda_{H,d}$ which derive from "extremal" feasible schedules, this result leads us to define the following subsets:

- $V - PPR_{H,d,P}$ = P -Feasible Vertex Subset of the polyhedron $\Lambda_{H,d}$ = {vertices z of the polyhedron $\Lambda_{H,d}$ which may be associated with some basis subset B such that the ordered subhypergraph $K_H(B)$ of the dual ordered hypergraph K_H , which is induced by B , is P -interval representable}.
- $V - PR_{H,d}$ = Fully Preemptive Vertex Subset of the polyhedron $\Lambda_{H,d}$ = the set $V - PPR_{H,d,+\infty}$ which corresponds to the case when the threshold preemption function P is equal to $+\infty$ (fully preemptive case).

- $V - NPR_{H,d}$ = *Non Preemptive Vertex Subset* of the polyhedron $\Lambda_{H,d}$ = the set $V - PPR_{H,d,0}$ which corresponds to the case when the threshold preemption function P is equal to 0 (non preemptive case).

The next section will then be devoted to describe the way the **GPPCSP** problem may be reformulated as a search problem on the set $V - PPR_{H,d,P}$, and to study the structure of this specific vertex set.

Example 1.14. If we refer to the specific **GPPCSP** instance of example 1.2 in Section 1.2, we see that the vector $z = (z_{e_0} = 0, z_{e_1} = 0, z_{e_2} = 0, z_{e_3} = 0, z_{e_4} = 1/2, z_{e_5} = 0, z_{e_6} = 1, z_{e_7} = 1, z_{e_8} = 1, z_{e_{10}} = 0, z_{e_{11}} = 0, z_{e_{12}} = 1)$ belongs to the set $V - PPR_{H,d,P}$, but does not belong to the set $V - NPR_{H,d}$.

2. STRUCTURAL PROPERTIES OF THE ANTICHAIN POLYHEDRON $\Lambda_{H,d}$

This part of our work is going to focus on the link which exists between the specific vertex subsets $V - PPR_{H,d,P}$ and $V - NPR_{H,d}$, of the polyhedron $\Lambda_{H,d}$ and efficient feasible solutions of the **GPPCSP** problem. We shall study the connectivity (in the sense of the general structure of polyhedra: see Sect. 1.5.2) of those subsets, and the results which we are going to obtain will eventually open the way to the design of new kinds of local search algorithms for the **GPPCSP** problem.

2.1. SOME THEORETICAL RESULTS

We are first going to state a Lemma which is going to better identify the link which exists between Proposition 1.13 and the definition of the subset $V - PPR_{H,d,P}$. Proposition 1.13 involves the active antichain subset $ACT(z) \subset E_\rho$ which may be associated with a vector z in the polyhedron $\Lambda_{H,d}$. It only requires the P -interval property to be satisfied by the subhypergraph $K_H(ACT(z))$ of the dual hypergraph K_H which is induced by $ACT(z) = \{e \in E_\rho \text{ such that } z_e \neq 0\}$. But the definition of the subset $V - PPR_{H,d,P}$ involves vertices of $\Lambda_{H,d}$, and requires the whole basis subset $B \subset E_\rho$ related to a vertex of $\Lambda_{H,d}$ to be such that the induced subhypergraph $K_H(ACT(z))$ be P -interval representable. We link both by stating:

Lemma 2.1 (extension lemma). *Let z be a vector in $\Lambda_{H,d}$ such that $ACT(z)$ defines a family of independant columns of the matrix M_H and such that the subhypergraph $K_H(ACT(z))$ is P -interval representable. We suppose that the duration function d is strictly positive. Then z is in the subset $V - PPR_{H,d,P}$.*

Proof. We only need to prove that z is a vertex of $\Lambda_{H,d}$ and that there exists a subset B of E_ρ which is a basis subset for z and which is such that $ACT(z) \subset B$ and that the ordered subhypergraph $K_H(B)$ is P -interval representable.

Let A be a subset of E_ρ such that:

- the ordered subhypergraph $K_H(A)$ of the dual ordered hypergraph K_H which is induced by A is P -interval representable;
- the rank of the restriction of M_H to the A indexed columns is equal to $\text{Card}(A)$.

Then we have to prove that there exists $B \subset E_\rho$, such that the ordered subhypergraph $K_H(A \cup B)$ is P -interval representable, and such that the rank of the submatrix of M_H which is induced by $A \cup B$ is equal to $\text{Card}(X)$. We set $q = \text{Card}(A)$ and we proceed by induction on $\text{Card}(X) - q$. Also, for any pair (U, Y) , where $U \subset E_\rho$ and $Y \subset X$, we denote by $M_H(U, Y)$ the submatrix of M_H which is induced by U (columns) and Y (rows).

If $q = \text{Card}(X)$, then we are done. If $q < \text{Card}(X)$, then we may consider $X' \subset X$, such that the determinant $\text{Det}(M_H(A, X'))$ which is defined by X' and A is different from 0, and $x_0 \in X - X'$. Then, in order to get our result, we only need to prove (because of the induction) that there exists an antichain b in E_ρ such that:

- $\text{Det}(M_H(A + \{b\}, X' + \{x_0\})) \neq 0$;
- the ordered subhypergraph $K_H(A \cup \{b\})$ is P -interval representable.

Let us suppose that A may be written $A = \{e_1, e_2, \dots, e_q\}$, in such a way that the linear ordering defined this way is a P -interval representation of $K_H(A)$. We may remark that, since $d(x_0) > 0$, x_0 must belong to $\bigcup_{i=1..q} e_i$. Let us also suppose (refutation), that there does not exist b in E_ρ such that $\text{Det}(M_H(A + \{b\}, X' + \{x_0\})) \neq 0$ and such that $K_H(A \cup b)$ is P -interval representable. (E1)

Then (E1) yields, that, for any x in $(X' + \{x_0\}) \cap e_1$, the ordered hypergraph $K_H(A \cup \{x\})$ is P -interval representable, and that $\text{Det}(M_H(A, X' + \{x_0\} - \{x\})) = 0$. Keeping on with the same reasoning, we define i_0 as the largest integer i such that for any x in $(X' + \{x_0\}) \cap e_i$, we have $\text{Det}(M_H(A, X' + \{x_0\} - \{x\})) = 0$. If $i_0 < q$, then, for any x in $(X' + \{x_0\}) \cap (e_{i_0+1} - \bigcup_{j=1..i_0} e_j)$, we may define the subset $b = b(x)$ of X as follows:

$$b = \{x\} \cup \{y \in X, \text{ such that } y \in e_{i_0} \cap e_{i_0+1}\}.$$

Clearly, b is in E_ρ , and the ordered hypergraph $K_H(A \cup \{b\})$ is P -interval representable. It follows that we have $\text{Det}(M_H(A, X' + \{x_0\} - \{x\})) = 0$, which allows us to deduce a contradiction on i_0 , except if $i_0 = q$. But if $i_0 = q$, we know that x_0 must be in some edge $e_i, i = 1..q$, and we get that $\text{Det}(M_H(A, X' + \{x_0\} - \{x_0\})) = 0$, which means that the rank of the submatrix $M_H(A, X')$ cannot be equal to q , and a contradiction. \square

We can now state our first *Antichain Polyhedron Theorem*, which will allow us to consider the non preemptive **GPPCSP** problem related to the ordered hypergraph H and to the duration function d , as a search problem defined on the *Non Preemptive Feasible Vertex Subset* $\underline{V} - NPR_{H,d}$.

Theorem 2.2 (non preemptive antichain polyhedron theorem). *Let Φ be some dominant feasible schedule for the non preemptive **GPPCSP** problem which is*

defined by the ordered hypergraph $H = (X, E, \rho)$ and by a strictly positive duration function d , and let z be the E_ρ indexed vector which derives from Φ . Then z is in $\mathbf{V} - \mathbf{NPR}_{H,d}$.

Proof. The feasible non preemptive schedule Φ gives rise, through application of the **ANTICHAIN-SEQ** procedure of algorithm 1, to some weighted antichain sequence $U = \{u_i, i = 0..n - 1, \text{weight of } u_i = \eta(u_i)\}$ and to some positive number sequence $\Delta = \{0 = \delta_0 < \delta_1 < \dots < \delta_{n-1} < \delta_n\}$, in such a way that:

- $\Delta = \{0\} \cup \{\text{the set of all end-points } END(\Phi(x)) \text{ and start-points } START(\Phi(x)) \text{ of the intervals } \Phi(x), x \in X, \text{ which belong to the real line } [0, +\infty[\};$
- for any $i = 0..n - 1, u_i = \{x \in X \text{ such that } \Phi(x) \text{ contains the interval }]\delta_i, \delta_{i+1}[\};$

To any edge u_i corresponds some column vector V_i of the incidence matrix M_H . V_i is a X indexed vector defined by: $V_{i,x} = 1$ if u_i contains the vertex $x \in X$, and $V_{i,x} = 0$ otherwise. Because of the Extension Lemma, we only need to prove that the **vectors $V_i, i \in 0..n - 1$, are linearly independent.** (E2)

In order to prove it, we denote by V the $\{0, 1\}$ -matrix whose columns are the vectors $V_i, i \in 0..n - 1$, and we proceed by induction on n . Clearly, u_0 cannot be the empty antichain: if we had $u_0 = Nil$, then it would be possible to left translate (with a parameter value equal to $-\delta_0$) the whole schedule Φ in such a way that it would contradict the fact that Φ is dominant. Let $x_0 \in X$ be an element of u_0 with minimal duration $d(x_0)$. We claim that $\Phi(x_0) = [0, \delta_0]$. (E3)

If it were not the case, then there would exist x_1 in u_1 such that $0 < START(\Phi(x_1)) < END(\Phi(x_0))$, and such that $START(\Phi(x_1))$ is the smallest possible with this property. But we could contradict the fact that Φ is dominant by left translating (with a parameter value equal to $START(\Phi(x_1))$) the task x_1 .

It follows from (E3) that all vectors V_i but vector V_0 are such that $V_{i,x_0} = 0$, which also means that the row of matrix V which corresponds to vertex x_0 is equal to $(1, 0, \dots, 0)$. Let us consider the ordered subhypergraph of H which is induced by $X - x_0$, and let us define, on the task set $X - x_0$, the following duration function d' :

- $d'(x) = d(x) - \delta_0$ if $x \in e_0$;
- $d'(x) = d(x)$ otherwise.

Then it also follows from (E3) that the schedule Φ' which is obtained by setting, for every x in $X - x_0$:

$$START(\Phi'(x)) = START(\Phi(x)) - \delta_0,$$

is a non preemptive dominant feasible schedule for the non preemptive instance of **GPPCSP** which is defined by $X - x_0, E, \rho$ and d' . It gives rise, through application of the **ANTICHAIN-SEQ** procedure of algorithm 1, to the weighted subsequence $U' = \{u_i, i = 1..n - 1, \text{weight of } u_i = \eta(u_i)\}$, and to the submatrix V' of V which is induced by the removal of the V_0 column and the x_0 row. We conclude by induction on n and we get (E2). \square

Our next result deals with the general case (partially preemptive), with the hypothesis that the cost function $Cost$ is linear. Under this hypothesis, it makes appear any **GPPCSP** instance related to an ordered hypergraph H , to a duration function d , and to a Preemption Threshold function P , as a search problem defined on the P -Feasible Vertex Subset $\mathbf{V} - \mathbf{PPR}_{H,d,P}$. It is non trivial and its usefulness comes from the fact that it will allow us to deal with **GPPCSP** while focusing on specific basis subsets, which are the main objects at the core of Linear Programming.

Theorem 2.3 (general linear case antichain polyhedron theorem). *Given an ordered hypergraph $H = (X, E, \rho)$, a strictly positive duration function d , a preemption threshold function P , and a E_ρ indexed vector $C \geq 0$. Then there exists an optimal solution Φ of the related linear **GPPCSP** instance such that the vector z which derives from Φ belongs to the P -Feasible Vertex Subset $\mathbf{V} - \mathbf{PPR}_{H,d,P}$.*

Proof. Let Φ be some feasible schedule associated with the **GPPCSP** instance defined by H, d, P and C , let z be some vector of $\Lambda_{H,d}$ which derives from Φ , and let $ACT(z)$ be the active edge subset of z . We know that the ordered subhypergraph $K_H(ACT(z))$ of the dual ordered hypergraph K_H , which is induced by $ACT(z)$, is P -interval representable. Solving the following linear program:

$L_{H,d}^C(ACT(z)) : \left\{ \text{Find a } ACT(z) \text{ indexed vector } t \geq 0, \text{ such that,} \right.$

$$\text{for every } x \in X, \quad \sum_{e \in ACT(z)/x \in e} t_e = d(x),$$

and which minimizes the quantity $\left. \sum_{e \in ACT(z)/x \in e} C_e \cdot t_e = d(x) \right\}$,

makes appear a subset A of $ACT(z)$ and an optimal solution t^* of $L_{H,d}^C(ACT(z))$ such that:

- the restriction of t^* to $ACT(z) - A$ is null;
- the rank of the restriction of the matrix M_H to the A indexed columns is equal to $Card(A)$.
- the ordered subhypergraph $K_H(A)$ of the dual ordered hypergraph K_H which is induced by A is P -interval representable.

We easily conclude by applying the Extension Lemma. □

The last result (*Connectivity Theorem*) of this section aims at telling us more about the connectivity (in the sense of the adjacency relation which is defined on the vertex set of any polyhedron: see Sect. 1.5.2), of the subset $\mathbf{V} - \mathbf{PPR}_{H,d,P}$. It should open the way to the resolution of the **GPPCSP** problem through the design of local transformation algorithms whose execution would involve some walk on the set $\mathbf{V} - \mathbf{PPR}_{H,d,P}$.

Theorem 2.4 (antichain polyhedron connectivity theorem). *Given an ordered hypergraph $H = (X, E, \rho)$, a strictly positive duration function d , and a preemption threshold function P . The subset $V - \mathbf{PPR}_{H,d,P}$ is connected in the sense of the usual adjacency relation which exists between the vertices of the polyhedron $\Lambda_{H,d}$.*

Proof. So let us consider our polyhedron $\Lambda_{H,d}^\epsilon$ as well as some threshold preemption function P and some vertex z in $V - \mathbf{PPR}_{H,d,P}$. We first have to introduce some conventions. Let us remark that we can associate with z some basis subset $B = \{e_1, e_2, \dots, e_{N=Card(X)}\}$, such that: (E4)

- $e_1, \dots, e_N \in E_\rho$ define N linearly independent indexed column vectors V_1, V_2, \dots, V_N of the matrix M_H ;
- there exists some feasible schedule Φ (for the **GPPCSP** instance related to H, d and P), such that z derives from Φ ;
- any edge e which is active for Φ belongs to B ;
- the ordered subhypergraph $K_H(B)$ of the dual ordered hypergraph K_H , is P -interval representable.

Since the ordered subhypergraph $K_H(B)$, induced from K_H by B , is P -interval representable, we may suppose that the edges $e_i, i = 1..n$, have been indexed in such a way that: (E5)

- if $x \in e_i$ and $y \in e_j$ are such that $x \rho y$ then $i < j$;
- if $x \in X$, then the indices i such that $x \in e_i$, are consecutive modulo $P(x)$ in the set $\{1..n\}$.

Let α be the largest i in $1..n$, such that for any $j < i, e_j$ is a singleton, ($Card(e_j) = 1$). For any $j < \alpha$, we denote by x_j the unique vertex (task) of e_j .

Let us denote by \mathbf{B}_0 the trivial basis whose all components are singletons: $\mathbf{B}_0 = \{\{x\}, x \in X\}$. We only need, in order to get our result, to prove the following statement: (E6)

- there exists a sequence $\mathbf{B}_0 = B_0, B_1..B_s = B$, of basis subsets, which all satisfy (E4), and which are such that, for any $k = 0..s - 1, B_k$ and B_{k+1} differ by at most one element.

Proceeding by refutation and by induction on α and on the sum $\sum_{i=1..n} Card(e_i)$, we may suppose that (E6) is false and that z and B have been chosen in such a way that: (E7)

- α is maximal;
- the sum $\sum_{i=1..n} Card(e_i)$ is the smallest possible, α being fixed.

A first case arises when $\alpha \geq 2$ and when there exists $i \geq \alpha$ and $\gamma < \alpha$ such that e_i contains the vertex x_γ such that $e_\gamma = \{x_\gamma\}$. In such a case, we denote by β the largest value i such that $x_\gamma \in e_i$, and we replace e_β by $e = e_\beta - \{x_\gamma\}$. Then we obtain a new edge family B' which keeps on satisfying (E4), and we get a contradiction on the minimality of the quantity $\sum_{i=1..n} Card(e_i)$. So we suppose (non trivial case) that either $\alpha = 1$ or that no edge $e_i, i \geq \alpha$, contains a vertex x_γ such that $\gamma < \alpha$ and $e_\gamma = \{x_\gamma\}$. (E8)

Then, the sketch of the remaining part of the proof is going to come as follows: (E9)

- we are first going (first step) to turn the duration function d into another one d^ϵ and the related vector z into another one z^ϵ , in such a way that z^ϵ remains a basis vector associated with B in the related polyhedron $\Lambda_{H,d}^\epsilon$. We shall do it in such a way that any antichain of B is going to be active for z ;
- this last property will next enable us (second step) to prove the existence of a path Γ^ϵ , which connects in $\Lambda_{H,d}^\epsilon$ the vertex z^ϵ to some vertex $z^{*\epsilon}$ and which only involves vertices of $\mathbf{V} - \mathbf{PPR}_{H,d,P}$, in such a way that the induction hypothesis (E7) may be applied to any basis $B^{*\epsilon}$ associated with $z^{*\epsilon}$;
- we conclude (last step) by making ϵ converge to 0 and by deducing our result through a convergence argument based upon the finiteness of:
 $\mathbf{V} - \mathbf{PPR}_{H,d,P}$.

In order to get the first step of the above sketch (E9), we consider some small number $\epsilon > 0$, and we define a vector z^ϵ by setting, for any edge e in B , $z_e^\epsilon = z_e + \epsilon$. This vector z^ϵ is a vertex of the polyhedron $\Lambda_{H,d}^\epsilon$ which is defined by: $\Lambda_{H,d}^\epsilon = \{z = (z_e, e \in E_\rho) \geq 0 \forall x \in X, \sum_{e/x \in e} z_e = d(x) + \epsilon(x)\}$, where $\epsilon(x)$ is given by: $\epsilon(x) = \epsilon \cdot (\text{number of edges } e_i, i = 1..n, \text{ such that } x \in e_i)$.

As a matter of fact, $\Lambda_{H,d}^\epsilon$ is the antichain polyhedron related to the ordered hypergraph H and to the duration function d^ϵ which, to any x in X , makes correspond $d^\epsilon(x) = d(x) + \epsilon(x)$. Introducing this small number ϵ allows us to deal with a vertex z^ϵ of $\Lambda_{H,d}^\epsilon$ which makes every element of B be active. Then the feasible schedule Φ may be turned into a schedule Φ^ϵ , which is feasible for the **GPPCSP** instance defined by H, d^ϵ and P . We also may do in such a way that Φ^ϵ agrees with the (E5) hypothesis. That means that if we set, for every $i = 1..n$:

$$T_i^\epsilon = \{t \in [0, +\infty[\text{ such that } e_i = \{x \in X \text{ such that } t \text{ is in the interior of the interval } \Phi^\epsilon(x)\}\},$$

then the T_i^ϵ are consecutive open intervals with length $z_{e_i}^\epsilon$: if $t \in T_i^\epsilon$, if $t' \in T_j^\epsilon$, and if $i < j$, then $t < t'$.

We deal with the second step of the above sketch (E9) by noticing that, because of (E8), the square matrix defined by $e_1, \dots, e_{\alpha-1}$ (columns) and by $(\bigcup_{i=1.. \alpha-1} \{e_i\})$ (rows) is the identical square matrix, and that, consequently, the square matrix Π defined by e_α, \dots, e_n (columns) and by $X - (\bigcup_{i=1.. \alpha-1} \{e_i\})$ (rows) is invertible. Also, the submatrix defined by the rows $X - (\bigcup_{i=1.. \alpha-1} \{e_i\})$ and the columns e_α, \dots, e_n is null. Let us consider some vertex y_0 which is in e_α , and let us set: $J(y_0) = \{j = \alpha..n, \text{ such that } y_0 \in e_j\}$. Let us denote by β the largest index value in $J(y_0)$, and let us define two edges f and f' of E_ρ as follows:

- $f = \{y_0\}$;
- $f' = e_\beta - \{y_0\}$.

Then we see that the edge family B^* obtained after adding f and f' to B is such that the subhypergraph $K_H(B^*)$ is P -interval representable. Let us then define

the following linear program LP^ϵ :

$$LP^\epsilon : \left\{ \begin{array}{l} \text{Find a } B^* \text{ indexed vector } z \geq 0, \text{ such that, for every } x \in X, \\ \\ \sum_{e \in B^*/x \in e} z_e = d(x) + \epsilon(x), \text{ and which maximizes the quantity } z_f \end{array} \right\}.$$

Clearly, the restriction to B^* of the vector z^ϵ is a feasible solution of LP^ϵ and defines a vertex of the polyhedron of LP^ϵ . Also, this vector z^ϵ is not an optimal solution of LP^ϵ , since one may turn z^ϵ into a better feasible z' solution of LP^ϵ , by choosing a sufficiently small number δ and by setting:

- $z'_f = \delta; z'_{f'} = \delta;$
- $z'_{e_\beta} = z_{e_\beta} - \delta; z'_e = z_e$ for any e in $B^* - \{f, f', e_\beta\}$.

Therefore, the Simplex Algorithm (see [43]) allows us to reach an optimal solution $z^{*\epsilon}$ of LP^ϵ by following some path Γ^ϵ , which starts from vertex z^ϵ and which ends in vertex $z^{*\epsilon}$, in the constraint polyhedron of LP^ϵ .

In order to conclude (last step of the above sketch (E9)), we see that $e_1, \dots, e_{\alpha-1}$ must be in any basis subset associated with any vertex of Γ^ϵ . Also, the antichain f must be in any basis subset $B^{*\epsilon}$ associated with vertex $z^{*\epsilon}$. Since the number of possible basis subsets is finite, there exists a sequence $\epsilon_k, k = 0 \dots \infty$, which converges to 0, and which is such that the related paths $\Gamma^{\epsilon_k}, k = 0 \dots \infty$, all induce a same sequence $B^* = D_0, D_1 \dots D_r = B$ of basis subsets, such that:

- for any $j = 0 \dots r$ and for any $k = 0 \dots \infty, D_j$ is associated with some vertex in Γ^{ϵ_k} ;
- for any $j = 0 \dots r - 1$, the basis subsets D_j and D_{j+1} differ by exactly one element.

But then, it comes that every basis subset $D_j, j = 0 \dots r$, may also be associated with some vertex of $\Lambda_{H,d}$, and is such that the subhypergraph $K_H(D_j)$, which is induced by D_j from the dual ordered hypergraph K_H , is P -interval representable. Thus any basis subset $D_j, j = 0 \dots r$, satisfies (E4). Also, edges $e_1, \dots, e_{\alpha-1}$ and f must be in B^* , and B^* may be written according to (E5) under the form: $B^* = \{f_1 \dots f_n\}$, with: (E10)

- if $x \in f_i$ and $y \in f_j$ are such that $x \rho y$ then $i < j$;
- if $x \in X$, then the indices i such that $x \in e_i$ are consecutive modulo $P(x)$ in the set $\{1 \dots n\}$;
- if $i < \alpha$, then $f_i = e_i$; $f_\alpha = f$.

Then we conclude: because of (E10) and of the (E7) hypothesis about the maximality of α , there exists a sequence of basis subsets $\mathbf{B}_0 = B_0, B_1 \dots B_q = B^*$, which all satisfy (E4) and which are such that, for any $k = 0 \dots q - 1, B_k$ and B_{k+1} differ by at most one element. By concatenating both sequences $\mathbf{B}_0 = B_0, \dots, B_q = B^*$ and $B^* = D_0, \dots, D_r = B$, we get a contradiction on (E7) and the result. \square

2.2. A SHORT DISCUSSION ABOUT ALGORITHMS

We must first insist on the fact that, since this paper is essentially dedicated to the theoretical study of the combinatorial structure of the **GPPCSP** problem, we are not able to do much more here about algorithms than proposing a kind of prospective discussion. Programming heuristics involving the management of the $L_{H,d}^C$ linear program requires a lot of work, and we plan doing it in the context of a specific contribution. Our feeling is that, since Theorem 2.3 allows us to rewrite the linear **GPPCSP** problem as follows

{Given:

- an ordered hypergraph $H = (X, E, \rho)$, and a duration function d ;
- a preemption threshold integer valued function P , defined on X ;
- a cost vector C , with indexation on the valid antichain E_ρ .

Find a vertex z in $\mathbf{V} - \mathbf{PPR}_{H,d,P}$, some associated basis subset B , and some P -interval representation τ^* of the ordered subhypergraph $K_H(B)$, such that $C.z$ is the smallest possible },

then Theorem 2.4 should probably enable us to design local search procedure whose execution would induce a well-driven walk on some connected subset $\mathbf{V} - \mathbf{PPR}_{H,d,P}$, along the edges of $V(\Lambda_{H,d})$. This intuition is based upon the fact that even if elementary Simplex moves are not very powerful by themselves, they can be controlled in many ways (primal-dual simplex method, Lagrangean decomposition) which might eventually help the programmer in defining very large neighbourhood, and thus in designing efficient algorithms. Still, we must admit that we did not work enough about this part of the problem to fully confirm this intuition.

More precisely, one may notice that the moves related to the adjacency relation which is defined on $V(\Lambda_{H,d})$ are most often likely not to cause any change in the value of the related cost value. As a matter of fact, the linear program $L_{H,d}^C$ is likely to be most often strongly degenerated. Because of this, elementary moves along the edges of $V(\Lambda_{H,d})$, (basic iterations of the Simplex Algorithm) should only be used as part of well-driven pathes (sequences of elementary moves) inside the vertex set $V(\Lambda_{H,d})$ of the $\Lambda_{H,d}$ polyhedron: according to this, a local search heuristics for the **GPPCSP** problem will make the current vertex z move inside $V(\Lambda_{H,d})$ or inside the subsets $V - PPR_{H,d,P}$ while jumping, at any iteration of the main search loop, from the origin of such a path until its extremity. Because of this, we shall talk about *micro-moves* when talking about those elementary moves, and we shall talk about *operational moves* when talking those well-driven sequences of such micro-moves.

Several algorithmic schemes may be designed according to those guidelines, and evaluating their efficiency will require a long time dedicated to software development and test. One may think for instance in defining *operational moves* as *micro-move* sequences which would aim at optimizing auxiliary cost functions while avoiding forbidden valid antichains. As an example, let us describe the following algorithmic scheme of algorithm 2, called **GPPCSP-SCHEDULE** which we implemented in [17], in the case when $C = \mathbf{1}$ and when $P = 0$ and $P = +\infty$:

Algorithm 2 GPPCSP-SCHEDULE

```

1: FORBID ← Nil; Not Stop; Cur – Sol ← Undefined; Cur – Value ←
   Undefined;
2: while Not Stop do
3:   Compute an optimal solution  $z$ , together with an associated basis subset
      $B$  of  $E_\rho$ , of the linear program  $PL(\text{FORBID})$ :
     {Find  $z$  in  $\Lambda_{H,d}$ , such that  $z_e = 0$  for any  $e$  in  $\text{FORBID}$ , and which mini-
     mizes  $z.C$ };
4:   Turn  $B$  into a weighted antichain sequence  $B^*$  which satisfies (C5), (C6)
     and (C7) of the section 1.4 and which contains at least one basis subset;
      $B^*$  defines a feasible schedule of the related GPPCSP instance; Let  $\tau^*$  a
      $P$ -interval representation of  $K_H(B^*)$ ;
5:   Compute an optimal feasible basis  $B1$  for the linear program  $PL(E_\rho - B^*)$ :
     {Find  $z$  in  $\Lambda_{H,d}$ , such that  $z_e = 0$  for any  $e$  which is not in  $B^*$ , and which
     minimizes  $z.C$ };
6:   Not Stop1;
7:   while Not Stop1 do
8:     Search  $e \in B1$  and  $f \in E_\rho - B1$  such that:
       –  $K_H(B1 - \{e\} + \{f\})$  is  $P$ -interval representable and admits a
          $P$ -interval representation which is compatible with  $\tau^*$ ;
       – Replacing  $B1$  by  $B1 - \{e\} + \{f\}$  improves  $B1$  in the sense of the
         Simplex Algorithm (taking degeneracy into account);
9:     if  $e$  and  $f$  exist then
10:      replace  $B1$  by  $B1 - \{e\} + \{f\}$ 
11:     else
12:       Stop1;
13:     end if
14:   end while
15:   Update Cur – Sol and Cur – Value;
16:   Remove some elements from FORBID;
17:   Pick up a subset  $A$  of  $B - B1$  and insert it into FORBID;
18:   Update Stop;
19: end while

```

Algorithm 2 is not truly a local search algorithm, but rather a hybrid one. The internal loop (lines 7-14) performs a local search on the neighbourhood structure which is directly induced by the Connectivity Theorem (the adjacency structure which associated with the vertex set of the polyhedron $\Lambda_{H,d}$). But, globally, **GPPCSP-SCHEDULE** performs some kind of random walk (main loop) while handling non feasible solutions. At any iteration of the main loop, **GPPCSP-SCHEDULE** computes a solution z of a restricted version of the linear program $L_{H,d}^C$, next turns it into a vertex of $\mathbf{V} - \mathbf{PPR}_{H,d,P}$ (a feasible schedule), and finally performs the local search associated with the internal loop.

TABLE 1. Non preemptive case, $P = 0$.

PSPLIB30 Instances	SOL/OPT	Case of Equality $SOL = OPT$	CPU (sec.)
All (480 instances)	1.90 %	291	67.5
Hard1 (341 instances)	2.67 %	152	95.0
Hard2 (240 instances)	3.37 %	77	93.6

TABLE 2. Non preemptive case, $P = \infty$

PSPLIB30 Instances	SOL/OPT	Case of Equality $SOL = OPT$	CPU (sec.)
All (480 instances)	0.15 %	442	9.9
Hard1 (341 instances)	0.21 %	303	13.9
Hard2 (240 instances)	0.30 %	202	17.4

The key instructions in **GPPCSP-SCHEDULE** are the instructions of lines 4, 16 and 17. We implemented the above scheme in [17] while using a very simple heuristic for (I1). We will briefly discuss in the Section 4 the way some “exactness” result may give rise to more sophisticated way of performing instruction of line 16. As for the instructions of lines 16 and 17, we chose not to remove any element from *FORBID* and to select the half part of $B - B1$ made with the edges e with smallest value z_e .

We then performed tests on the **RCPSP** instances of the PSPLIB30 library (see [29]), while using a Duron, 1 GHz, through a C++ code compiled by g++. We addressed the cases $P = 0$ (non preemptive case) and the case $P = +\infty$ (preemptive case). We got the following results, where SOL/OPT denotes the gap (in percentage) between the optimal solution OPT and our own solution SOL .

The results of Table 2.2 are acceptable, but they are far from being optimal, and we feel that it should be possible to significantly improve those results by refining instructions of lines 4 and 17. As we just told it, this paper does not claim algorithmic efficiency, and turning the ideas which it contains into efficient algorithms remains a challenge. Better experimental results can be obtained through a combination of branch/bound and constraint propagation techniques (see Baptiste and Demassez [6] on PSPLIB60) and also through specific insertion techniques (see Artigues *et al.* [2,3]).

The results of Table 2.2 are rather good. Testing the preemptive case is difficult (for instance in the case of the PSPLIB60), since in most cases, we do not really know the value OPT of the optimal solution, and since few experimental results exist. As a matter of fact, most experiments related to the preemptive case (Djellab [19], Josefowska *et al.* [26]) have been performed on the *Multiprocessor Scheduling Problem*, which is a very specific case of the **RCPSP** Problem, and they involved specifically designed algorithm. Since our algorithm is very generic, we did not try to adapt it to the multiprocessor scheduling case.

3. COMMUTATIVE ORDERED HYPERGRAPHS: A CASE WHEN OPTIMAL PREEMPTIVE SCHEDULING CAN BE DONE THROUGH LINEAR PROGRAMMING

We focus now on preemptive scheduling, and on the way solving the linear program:

$$L_{H,d}^1 : \{\text{Compute } z \text{ in } \Lambda_{H,d} \text{ which minimizes the quantity } 1.z\}$$

may provide us with an optimal (or at least a good) solution of the preemptive version of the Makespan Minimization **GPPCSP** problem (the case when $P = +\infty$). Since this paper is mainly dedicated to the theoretical analysis of the combinatorial structure of the **GPPCSP** problem, we shall not deal with this problem taken as a whole, but we shall identify cases (Sect. 3.2) when the ordered hypergraph $H = (X, E, \rho)$ is such that, for any duration function d , the optimal value of the program $L_{H,d}^1$ is equal to the optimal value of the **GPPCSP** instance defined by H, d and $P = +\infty$. Doing it will lead us to introduce a specific *commutativity* property, and to allow us to extend a former result of Moukrim and Quilliot (see [36]), related to the *k Processor Scheduling Problem*, to more general problems (**RCPS**, *Scheduling with Disjunctive Constraints Problem* or **GPPCSP**). Though we do not deal here a lot with algorithms, one should be aware that the main application of this theoretical study should be related to the design of reconstruction heuristics, which will turn a feasible solution of the program $L_{H,d}^1$ into a preemptive **GPPCSP** schedule. This point will be briefly discussed in Section 3.3.

3.1. COMMUTATIVE ORDERED HYPERGRAPHS

For any pair (e, e') in the antichain edge set E_ρ , we set:

- $\text{MAX}(e, e') = \{x \in e \cup e' \text{ such that there exists } y \text{ in } e \cup e' \text{ which satisfies } x \rho_T y\}$;
- $\text{MIN}(e, e') = \{y \in e \cup e' \text{ such that there exists } x \text{ in } e \cup e' \text{ which satisfies } x \rho_T y\}$;
- $\text{EQ}(e, e') = (e \cup e') - (\text{MAX}(e, e') \cup \text{MIN}(e, e'))$.

Remark 3.1. Commutativity will aim at helping us turning any solution z of the linear program $L_{H,d}^1$ into a feasible preemptive schedule. If e and e' are two active antichains associated with z , then $\text{MAX}(e, e') \cap e$ and $\text{MIN}(e, e') \cap e'$ identify the tasks which need to commute in order to avoid the existence of a circuit (e, e', e) in the sense of the relation τ , while $\text{EQ}(e, e')$ identifies those vertices which may move from e to e' (or conversely) without creating any new circuit.

We say that the ordered hypergraph $H = (X, E, \rho)$ is *weakly commutative* if for any pair (e, e') in E_ρ , it is possible to *commute* e and e' , that means if it is possible to find $f_1..f_q$ and $f'_1..f'_p$ in E_ρ , such that:

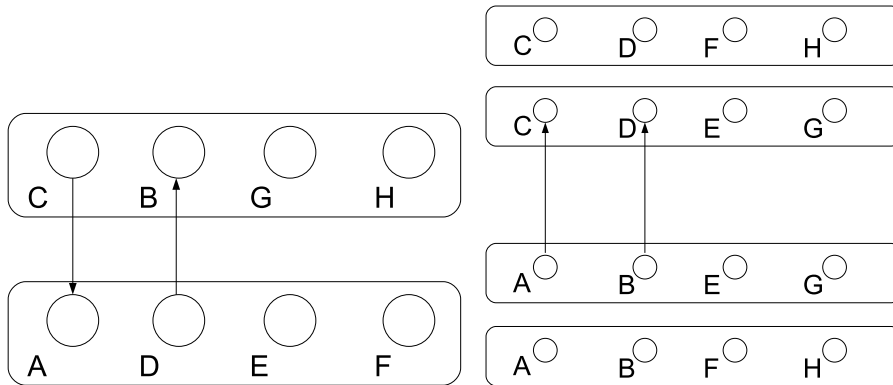


FIGURE 1. Before and after commutation.

– for any $x \in EQ(e, e')$, we have: (C8)

$$1/q \cdot \text{Card}(\{i = 1..q \text{ such that } x \in f_i\}) + 1/p \cdot \text{Card}(\{j = 1..p \text{ such that } x \in f'_j\}) = 1;$$

– for any $i = 1..q, f_i \subset EQ(e, e')$ and for any $j = 1..p, f'_j \subset EQ(e, e')$; (C9)

– for any $i = 1..q, \text{MIN}(e, e') \cup f_i \in E_\rho$; for any $j = 1..p, \text{MAX}(e, e') \cup f'_j \in E_\rho$. (C10)

In such a case, we notice that the antichain set $\{\text{MIN}(e, e') \cup f_i, i = 1..q\} \cup \{\text{MAX}(e, e') \cup f'_j, j = 1..p\}$ does not contain any circuit in the sense of the dual ordered hypergraph $K_H = (E_\rho, F, \tau)$.

Example 3.2 (a weakly commutative hypergraph). Let us consider the ordered hypergraph H defined by:

- $X = \{A, B, C, D, E, F, G, H\}$;
- $A \rho C; B \rho D; C \rho E$;
- $E =$ the monotonic edge family whose maximal elements are $e_1 = \{A, D, E, F\}$, $e_2 = \{B, C, G, H\}$, $e_3 = \{C, D, F, H\}$, $e_4 = \{C, D, E, G\}$, $e_5 = \{A, B, F, H\}$ and $e_6 = \{A, B, E, G\}$.

The two antichains e_1 and e_2 define a circuit of K_H . We may commute them by setting $q = 2 = p$ and: $\text{MIN}(e_1, e_2) = \{A, B\}$; $f_1 = \{F, H\}$; $f_2 = \{E, G\}$; $f'_1 = \{F, E\}$; $f'_2 = \{H, G\}$.

The drawing of Figure 1 represents $e_1 = \{A, D, E, F\}$ and $e_2 = \{B, C, G, H\}$ before and after commutation.

The main meaning of weak commutativity is that if e and e' are two active antichains associated with some feasible solution z of the linear program $L_{H,d}^1$, then the tasks inside e and e' may be rearranged in such a way that the resulting antichains do not define any circuit (in the sense of $K_H = (E_\rho, F, \tau)$), and that the makespan of z is not increased. Commutativity and strong commutativity impose restrictions to the way this operation can be performed, which will enable us to get a theoretical *Exactness Theorem*, as well as reconstruction algorithms.

If, for any pair (e, e') as above, $f_1..f_q$, and $f'_1..f'_p$ may be chosen in such a way that one of the two following assertions (C11) or (C12) is true:

(C11): there exists j in $\{1..p\}$ such that $f'_j \subset EQ(e, e') \cap e'$;

(C12): there exists i in $\{1..q\}$ such that $f_i \subset EQ(e, e') \cap e$;

then we say that H is *commutative*. In case (C11) is true, then we say that e is the *receiver of the commutation of e and e'* , else we say that e' is the *receiver of this commutation*.

If it is possible, for any pair (e, e') as above, to do in such a way that $p = q = 1$, then we say that H is *strongly commutative*. Clearly, the *strong commutativity* implies the *commutativity*, which in turn implies the *weak commutativity*.

Example 3.3 (a commutative hypergraph). Let us consider the ordered hypergraph H defined by:

- $X = \{A, B, C, D, E, F, G, H\}$;
- $F \rho B; F \rho A; C \rho E$;
- $E =$ the monotonic edge family whose maximal elements are $e_1 = \{A, B, C, D\}$, $e_2 = \{E, F, G, H\}$, $e_3 = \{E, A, C\}$, $e_4 = \{D, F, C, G\}$, $e_5 = \{D, F, C, H\}$ and $e_6 = \{D, F, G, H\}$.

Let us suppose that $e = e_1$, and that $e' = e_2$. They define a circuit in the subhypergraph K_H , and they commute by considering that $q = 3$, that $f_1 = \{C, G\}$, $f_2 = \{C, H\}$, $f_3 = \{G, H\}$ and that $p = 1$, with $f'_1 =$ the empty subset Nil . In this case, $e = e_1$ is the receiver of the commutation (since $f'_1 = Nil$). We may check that the ordered hypergraph which is defined by X, E and ρ is not strongly commutative.

In case there exists a number k such that E is the set of the subsets of X with no more than k elements (case of the k Processor Scheduling Problem), then we see that H is commutative if for every pair (e, e') in E_ρ , neither $MAX(e, e')$ nor $MIN(e, e')$ has more than k elements. In such a case, H is also strongly commutative. One checks that, in any case, if the partial ordering ρ defines an interval order (see [38]), then the ordered hypergraph $H = (X, E, \rho)$ is strongly commutative.

3.2. A THEORETICAL RESULT

Then we may state the following result, which tells us that if the weighted ordered hypergraph $H = (X, E, \rho)$ is commutative, then the optimal value of the related preemptive version of the **GPPCSP** problem, may be computed by solving the linear program $L_{H,d}^1$. This result extends a former result of Moukrim and Quilliot related to the specific case of the k -Processor Scheduling Problem (see [36]). It also raises the problem of finding a recognition algorithm for the Commutativity property. We did not address this problem. As a matter of fact, we considered that the most natural way to apply this result was to use it in order to derive reconstruction procedures, which turns a solution of $L_{H,d}^1$ into a preemptive feasible **GPPCSP** solution.

Theorem 3.4 (exactness theorem). *Let $H = (X, E, \rho)$ be some ordered hypergraph. Then two following statements (D1) and (D2) are true:*

- (D1): *If, for any integral duration function $d \geq 0$, there exists a feasible schedule for the preemptive **GPPCSP** instance defined by H and d , with its makespan equal to the optimal value of $L_{H,d}^1$, then H is weakly commutative;*
- (D2): *If H is commutative, then, for any integral duration function $d \geq 0$, there exists a feasible schedule for the preemptive **GPPCSP** instance defined by H and d , with makespan equal to the optimal value of the program $L_{H,d}^1$.*

Definition 3.5 (preliminary definitions). Those definitions will help us in moving from a problem involving fractional numbers to a purely combinatorial problem only involving valid antichain sequences. For any subset U of X , we set:

$$\text{INF}(U) = \{x \in U \text{ such that no } y \text{ exists in } U \text{ which satisfies } y \rho_T x,$$

where ρ_T is the transitive closure of $\rho\}$.

We may, for any subset U of X , define $SUP(U)$ by the same way, and proceed to the same remark. We define a *co-subset* W of E_ρ as being a subset inside which any element may eventually appear several times. If W is such a co-subset, and if $e \in E_\rho$, then we denote by $O(W, e)$ the number of times e appears inside W , and we call this number the occurrence number of e in W . The cardinality of such a co-subset becomes equal to the sum of all the occurrence numbers $O(W, e)$. The union of two co-subsets W' and W'' is a co-subset W such that, for any e in E , $O(W, e) = O(W', e) + O(W'', e)$.

Proof of (D1). Let us consider e and e' in E_ρ , together with some duration function d such that:

- $d(x) = 1$ for every $x \in e \cup e'$;
- $d(x) = 0$, for any $x \in X$ which is not in $e \cup e'$;

This provides us with a situation such that:

- the optimal value of the linear program $L_{H,d}^1$ is equal to 2;
- any **GPPCSP** feasible schedule Φ related to H, d and $P = +\infty$, must be such that: for any $x \in MIN(e, e'), y \in MAX(e, e'), x$ precedes y according to Φ ;

We know (because of the General Linear Case Antichain Polyhedron Th. 2.3) that an optimal schedule Φ for the preemptive version of **GPPCSP** related to $H, d, P = +\infty$, and the makespan minimization criterion, may be chosen in such a way that the vector z which derives from Φ through the **ANTICHAIN-SEQ** construction of algorithm 1, is rational. Let us suppose that the makespan of such a schedule Φ is equal to 2, and that q is an integer such that z may be written $z = z.(1.q)$, where Z is an integral vector. Then we may define two subsets F and F' of E_ρ by setting:

- f is in F , if there exists u in E_ρ , such that u is active for Φ and may be written $u = f \cup \text{MIN}(e, e')$. We count f , inside F , as many times as the integral quantity Z_u ;
- f' is in F' , if there exists u' in E_ρ , such that u' is active for Φ and may be written $u' = f' \cup \text{MAX}(e, e')$. We count f' , inside F' , as many times as the integral quantity $Z_{u'}$.

This construction makes Φ derive from a weak commutation of e and e' through F and F' . We conclude.

Proof of (D2). Let us suppose that H is commutative and that d is a given integral duration function. We first notice that:

(E11: For any pair $a, b \in E_\rho$ such that $b \subset \text{INF}(X - a)$, we have:

(E11'): For any $x \in b$, the subset $\{y \in X \text{ such that } y \rho_T x\}$ of X is included into $a \cap \text{INF}(X)$;

(E11''): $b \subset \text{INF}(X) \cup \text{INF}(X - \text{INF}(X))$.

We must prove that if z is a vertex of $\Lambda_{H,d}$, then it is possible to turn z into an element of $\mathbf{V} - \mathbf{PR}_{H,d}$ without deteriorating its makespan. In order to do it, we are going to proceed in several steps.

First step. We turn this problem, which involves rational numbers, into a purely combinatorial problem which involves co-subsets. If $\{e_1, \dots, e_p\}$ is a subset of E_ρ such that $p = \text{Card}(X)$, then $\{e_1, \dots, e_p\}$ defines a square submatrix of the constraint matrix M_H of $L_{H,d}^1$, and the determinant of this submatrix is called a subdeterminant of M_H . Let us denote by δ the largest subdeterminant of the constraint matrix of the linear program $L_{H,d}^1$, and let set $\epsilon = \text{Index Value}$ of $L_{H,d}^1 = 1/\delta$. Any vertex z of $\Lambda_{H,d}$ may be written $z = \epsilon.Z$, where Z is an integral vector, and we may denote by Q the integer which is such that optimal value of $L_{H,d}^1 = Q.\epsilon$. Then, we pick up $z = \epsilon.Z$ in $\Lambda_{H,d}$ such that Z is integral and $Z.1 = Q$.

Let z be a vertex of $\Lambda_{H,d}$ and let us set: $W = \{e \in E_\rho \text{ such that } z_e \neq 0\}$. We may consider W as a co-subset of E_ρ , in such a way that for any valid antichain $e \in W$, the occurrence number of e in W is equal to Z_e . W may contain some circuit in the sense of the dual ordered hypergraph $K_H = (E_\rho, F, \tau)$, and this keeps us from converting in a straightforward way z into a feasible preemptive **GPPCSP** schedule. Still W is a co-subset such that:

$$- \text{Card}(W) = Q; \tag{E12}$$

$$- \text{for any } x \in X, \text{ the number of valid antichains } e \text{ in } W \text{ which contain } x \text{ is equal to } D(x) = \delta.d(x). \tag{E13}$$

We need to prove, in order to conclude, the existence of some co-subset W^0 of E_ρ such that (E12) and (E13) are true and such that W^0 does not contain any circuit in the sense of the dual ordered hypergraph $K_H = (E_\rho, F, \tau)$.

Second step. We proceed by induction on the cardinality Q of the co-subset W , and we simplify, through application of the induction hypothesis, the above co-subset problem.

Let us consider some valid antichain e which is in W . If, for any $x \in e$, we set $d_e(x) = d(x) - \epsilon$, and if, for any $y \notin e$, we set $d_e(y) = d(y)$, then we turn d into a *reduced* duration function d_e in such a way that:

- the optimal value of L_{H,d_e}^1 is equal to $\epsilon.(Q - 1)$;
- d_e remains equal to the product of ϵ with an integral function D_e .

It means that the induction hypothesis may be applied to H and d_e , and that if e is included into $INF(X)$ or into $SUP(X)$, then we become able to conclude.

It follows that we only need to prove that: (E14)
there exists some co-subset W_1 of E_ρ such that (E12) and (E13) are true, and such that W_1 contains some antichain e which is such that: either $e \subset INF(X)$ or $e \subset SUP(X)$.

Third step. We keep on with the induction hypothesis, and we check that it allows us to easily conclude except in the case when:

- for every antichain e in W , $e \cap INF(X) \neq \emptyset$ and $e \cap SUP(X) \neq \emptyset$;
- there exists u in W such that $u \subset INF(X) \cup INF(X - INF(X))$.

In order to do it, let us suppose that some antichain $e \in W$ is such that $e \cap INF(X) = \emptyset$. Then we get, by applying our induction hypothesis to the hypergraph H and d_e , that there exists a co-subset W'_1 such that: (E15)

- W'_1 contains some antichain e' which is such that: $e' \subset INF(X - e)$;
- $\text{Card}(W'_1) = Q - 1$;
- for any $x \in X$, the number of valid antichains e in W'_1 which contain x is equal to $D(x) - 1 = \delta.(d(x) - \epsilon)$ if $x \in e$ and is equal to $D(x) = \delta.d(x)$ otherwise.

But then, we deduce from $e \cap INF(X) = \emptyset$ that $e' \subset INF(X)$ and the result. It comes that, for any antichain e in W , we may suppose that $e \cap INF(X)$ is not empty, as well as (symmetry) $e \cap SUP(X)$.

By the same way, we may suppose that W is such that there exists $u \in W$ with: $u \subset INF(X) \cup INF(X - INF(X))$. If it is not true, we may pick up e in W , and apply the induction hypothesis to H and d_e , and get W'_1 such that (E15) above is true. Then it comes that there exists u in W'_1 such that $u \subset INF(X - e)$, which also means such that $u \subset INF(X) \cup INF(X - INF(X))$. We only need to replace W by $\{e\} \cup W'_1$ in order to get the result.

Fourth (last) step. We are going to choose W in such a way that it may be viewed as extremal in relation to the rearrangement problem defined by (E14). Then we shall commute well chosen antichains of W , and, while using a linear programming trick, we shall contradict this extremality assumption. This will allow us to conclude.

More precisely, by considering W and u as above, and by applying the induction to H and to d_u , we get W'_1 such that (E15) above is true. By replacing W by $\{u\} \cup W'_1$, we may do in such way that W and $u \in W$ satisfy: (E16)

$$* \quad u \subset INF(X) \cup INF(X - INF(X)).$$

- * W may be written $W = \{u\} \cup W'_1$ in such a way that:
 - W'_1 does not contain any circuit in the sense of the dual ordered hypergraph $K_H = (E_\rho, F, \tau)$;
 - $\text{Card}(W'_1) = Q - 1$;
 - For any $x \in X$, the number of valid antichain u in W'_1 which contain x is equal to: $D(x) - 1 = \delta \cdot (d(x) - \epsilon)$ if $x \in u$ and to $D(x) = \delta \cdot d(x)$ otherwise;
- * $u \cap \text{INF}(X - \text{INF}(X))$ is minimal for the inclusion relation.

Since W'_1 does not contain any circuit in the sense of the relation τ , we may label the elements of W'_1 by setting $W'_1 = v_1..v_{Q-1}$ in such a way that for any $i, j = 1..Q - 1, v_i \tau v_j$ implies $i < j$.

In case no i exists such that $v_i \tau u$, then we are done, else we may pick up i_0 , which is such that $v_{i_0} \tau u$, and which is the largest possible with this property. If $i_0 < Q - 1$, then we may conclude by applying the induction hypothesis on X, E and ρ and on the duration function d^* defined, for any $x \in X$, by:

$$d^*(x) = d(x) - (\text{Card}(N(x)) \cdot \epsilon,$$

where $N(x)$ is the set $N(x) = \{i = i_0 + 1, \dots, Q - 1, \text{ such that } x \in v_i\}$. So we suppose now (non trivial case), that $i_0 = Q - 1$, and we try to commute u and $v = v_{Q-1}$. This means that we apply the commutativity hypothesis to u and v : this makes appear $f_1..f_k$, and $g_1..g_p$ in E_ρ , such that (constraints C8, C9, C10, C11, C12):

- for any $x \in EQ(u, v), 1/k \cdot \text{Card}(\{i = 1..k/x \in f_i\}) + 1/p \cdot \text{Card}(\{j = 1..p/x \in g_j\}) = 1$;
- for any $i = 1..k, f_i \subset EQ(u, v)$ and for any $j = 1..p, g_j \subset EQ(u, v)$;
- for any $i = 1..k, \text{MIN}(u, v) \cup f_i$ belongs to E_ρ ; for any $j = 1..p, \text{MAX}(u, v) \cup g_j$ belongs to E_ρ ;
- one of the two following assertions (E17.1) or (E17.2) is true:
 - (E17.1): there exists j in $\{1..p\}$ such that $g_j \subset EQ(u, v) \cap v$;
 - (E17.2): there exists i in $\{1..k\}$ such that $f_i \subset EQ(u, v) \cap u$.

Two cases must be considered:

- First case. (E17.1) above is true (u is the receiver of the commutation of u and v). We may suppose that $g_1 \subset EQ(u, v) \cap v$, and we set $h_1 = f_1 \cup \text{MAX}(u, v)$. In such a case, we deduce from (E16) that any element in h_1 is also in $\text{SUP}(X)$, since any element of $\text{MAX}(u, v)$ is in $\text{SUP}(X)$, as well as any element of $EQ(u, v) \cap v$. Also, we may deduce some optimal solution z^* of $L_{H,d}^1$ by setting, for any valid antichain e in E_ρ :
 - $A_e = \text{Card}(\{s = 1..Q - 2 \text{ such that } e = vs\})$;
 - $B_e = \text{Card}(\{i = 1..k \text{ such that } e = f_i \cup \text{MAX}(u, v)\})$;
 - $C_e = \text{Card}(\{j = 1..p \text{ such that } e = g_j \cup \text{MAX}(u, v)\})$;
 - $z_e^* = \epsilon \cdot [A_e + 1/k \cdot B_e + 1/p \cdot C_e]$.

This solution is such that $z_{h_1}^* \neq 0$. The vector z^* may not be a vertex of the polyhedron $\Lambda_{H,d}$, but it can be expressed as a barycentric combination

of vectors which are in $V(\Lambda_H)$ and which are optimal solutions of the linear program $L_{H,d}^1$. That means that there exists a vertex z' of Λ_H , which is an optimal solution of $L_{H,d}^1$, and which is such that $z_{h_1}^* \neq 0$. It becomes sufficient to replace z by z' in order to conclude.

- Second case. (E17.2) above is true (v is the receiver of the commutation of u and v). We may suppose that $f_1 \subset EQ(u, v) \cap u$ and we set $h_1 = f_1 \cup MIN(u, v)$.

In such a case, we deduce from (E16) that $h_1 \subset INF(X - INF(X))$ and that $h_1 \cap INF(X - INF(X))$ is strictly included into $u \cap INF(X - INF(X))$. (E18)

Also, by proceeding by the same way as for the first case, we see that there must exist a vertex z' of $\Lambda_{H,d}$, which is an optimal solution of $L_{H,d}^1$, and which is such that $z_{h_1}^* \neq 0$. We conclude by deducing from (E18) a contradiction on the minimality of $u \cap INF(X - INF(X))$. \square

3.3. A FEW WORDS ABOUT RECONSTRUCTION PROCEDURES

As was done in [35] by Moukrim and Quilliot in the case of the k Processor Scheduling Problem, we may derive from the above results several reconstruction procedures. These procedures take in input a valid antichain weighted sequence U which only satisfies the duration constraint C5 of section 1.2, and turn it, in the case when the ordered hypergraph $H = (X, E, \rho)$ is commutative, into a feasible schedule Φ of the related preemptive makespan minimization **GPPCSP** instance, in such a way that the makespans of U and Φ are equal. In the general case, it will be possible to turn these procedures into heuristic reconstruction procedures, which compute an efficient feasible **GPPCSP** schedule from a solution of the linear program $L_{H,d}^1$.

We are going to describe here in a simplified way such a reconstruction procedure **SCHEDULE-RECONSTRUCT** (algorithm 3), which acts in the case when the ordered hypergraph H is **strongly commutative**. This procedure is normally designed in order to handle weighted valid antichain sequences. Still, in order to simplify its description, we are going to turn those lists into fixed size arrays. In order to do this, we consider an optimal solution $z = \epsilon.Z \in V(\Lambda_{H,d})$ of $L_{H,d}^1$ such that Z is integral. We denote by Q the quantity $Z.1 = Q$, and we do as in the proof of the above theorem: that means that we set $W(z) = \{e \in E_\rho \text{ such that } z_e \neq 0\}$, and that we consider $W(z)$ as a co-subset of E_ρ , in such a way that for any valid antichain $e \in W(z)$, the occurrence number of e in $W(z)$ is equal to Z_e . Then we turn $W(z)$ into an array **AR**, with indexation on the set $1..Q$, and with values in E_ρ , and we act on **AR** in order to make it such that:

if i, j in $1..Q$ are such that $AR[i] \tau AR[j]$, (in the sense of the hypergraph $K_H = (E_\rho, F, \tau)$), then $i < j$; (E19)

we do it through a well-driven sequence of applications of the following operator

COMMUTE:

COMMUTE($i, j, i < j$ in $\{1..Q\}$):

Compute f and g subsets of $EQ(AR[i], AR[j])$ in such a way that:

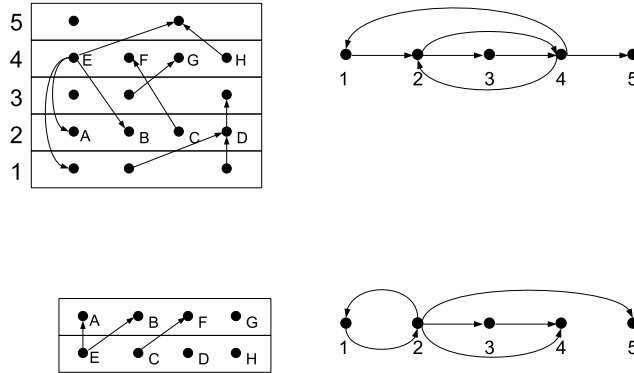


FIGURE 2. Array and induced graph before and after commutation.

- f and g are disjoint and $f \cup g = EQ(AR[i], AR[j])$;
- $MIN(AR[i], AR[j]) \cup f$ and $MAX(AR[i], AR[j]) \cup g$ are in E_ρ ; (E20)
- Either $f \subset (EQ(AR[i], AR[j]) \cap AR[j])$ or $g \subset (EQ(AR[i], AR[j]) \cup AR[j])$;
- If (E18) holds with $f = (EQ(AR[i], AR[j]) \cap AR[j])$ and $g = (EQ(AR[i], AR[j]) \cup AR[j])$ then $f = (EQ(AR[i], AR[j]) \cap AR[j])$ and $g = (EQ(AR[i], AR[j]) \cap AR[j])$;

Replace $AR[i]$ by $MIN(AR[i], AR[j]) \cup f$;
 Replace $AR[j]$ by $MAX(AR[i], AR[j]) \cup g$;

We control the way this operator is applied through the following predicate

GOOD-PATTERN:

GOOD-PATTERN($i, j, k, l, i \leq l < k \leq j$ in $\{1..Q\}$):

GOOD-PATTERN is true if:

- there does not exist s, t such that $i \leq s < t \leq k - 1$ and $AR[t] \tau AR[s]$;
- $AR[k] \tau AR[l]$, and l is maximal in $\{i..k - 1\}$ with this property.

Example 3.6. The array AR represented in Figure 3.3 (with $Q = 5$) makes appear that **GOOD-PATTERN**(1, 5, 4, 2) is true. We commute $AR[2]$ and $AR[4]$, while supposing that E is made with the subsets of X with cardinality no more than 4. The first drawing represents the complete array AR , together with the graph which is induced by $\{AR[1]..AR[5]\}$ and by the relation τ . The second drawing represents $AR[2]$ and $AR[4]$ after they commuted, together with the resulting induced graph.

Then the procedure **SCHEDULE-RECONSTRUCT** comes as in algorithm 3:

We let the reader come back to the proof of the Exactness Theorem 3.4 and check that this procedure turns z into an array AR such that (E19) is true, that means turns z into a preemptive **GPPCSP** feasible schedule whose makespan is equal to $1.z$. Also, one will check that, in case the hypergraph $H = (X, E, \rho)$ is not strongly

Algorithm 3 SCHEDULE-RECONSTRUCT

Require: a vertex z of $\Lambda_{H,d}$ which is an optimal solution of the linear program $L_{H,d}^1$

- 1: Turn $W(z)$ into a valid antichain array AR , with indexation on the set $\{1\dots Q\}$, as described above
- 2: Initialize a Stack $STACK$ with one level, which contains the index pair $(1, Q)$;
- 3: **while** $NotEmpty(STACK)$ **do**
- 4: Let (i, j) be the index pair $Top(STACK)$;
- 5: Look for k and l such that **GOOD-PATTERN** (i, j, k, l) ;
- 6: **if** k and l do not exist **then**
- 7: $Pop(STACK)$
- 8: **else**
- 9: $COMMUTE(l, k)$;
- 10: **end if**
- 11: **if** $AR[l]$ is the receiver of this commutation **then**
- 12: $Push$ the pair $(i, k - 1)$ at the top of $STACK$
- 13: **else**
- 14: $Push$ the pair $(l + 1, k)$ at the top of $STACK$
- 15: **end if**
- 16: **end while**

commutative, an eventual failure of the **SCHEDULE-RECONSTRUCT** process will make appear a valid antichain pair $(AR[i], AR[j])$ which cannot strongly commute.

In the specific case when **GPPCSP** is the k processor scheduling problem, that means when the relation E is defined by: $A \subset X$ is in E iff $Card(A) \leq k$, then this procedure may be simplified as in algorithm 4 (proof left to the reader).

Algorithm 4 K -Proc-SCHEDULE-RECONSTRUCT

Require: a vertex z of $\Lambda_{H,d}$ which is an optimal solution of the linear program $L_{H,d}^1$

- 1: Turn $W(z)$ into a valid antichain array AR , with indexation on the set $\{1\dots Q\}$, as described above
- 2: $Not\ Stop$
- 3: **while** $Not\ Stop$ **do**
- 4: Look for k and l such that **GOOD-PATTERN** $(1, Q, k, l)$
- 5: **if** k and l do not exist **then**
- 6: $STOP$
- 7: **else**
- 8: $COMMUTE(l, k)$;
- 9: **end if**
- 10: **end while**

This last algorithm may be compared with the Moukrim/Quilliot algorithm of [35]. As we told it, it is possible to turn the above procedure **SCHEDULE-RECONSTRUCT** into a heuristic procedure, which computes an efficient feasible **GPPCSP** schedule from a solution of the linear program $L_{H,d}^1$. We do it by extending the **COMMUTE** operator in such a way that it acts on non valid antichains, and by rearranging a weighted not valid antichain sequence which satisfies constraints (C5) and (C6) of Section 1.2, in such a way that it becomes a valid antichain sequence which satisfies (C5) and (C6).

4. CONCLUSION

We just introduced here a combinatorial framework involving hypergraphs and partially ordered sets, in order to make easier dealing with a parallelism constrained scheduling problem. This framework tends to unify preemptive and non preemptive models. We used it in order to get theoretical results related to the structure of the feasible sets which may be derived from those models, and in order to design an algorithmic scheme which works in a heuristic way on both preemptive and non preemptive parallelism constrained scheduling problems.

Still, we almost did not deal here with algorithmic aspects of scheduling. Some questions may be set: for instance, would it be possible to impose, inside the statement of the general Connectivity Theorem, some bound on the makespan allowed for the schedules which derive from the vertices of the set $V - PPR_{H,d,P}$. The existence of such bounds would then provide the basis for the design of efficient **GPPCSP** local search procedures designed in order to produce a well driven walk on the set $V - NPR_{H,d}$. Also, would it be possible to design an algorithm, endowed with some kind of exactness property, which would efficiently turn an element of $V - PR_{H,d}$ into an element of $V - NPR_{H,d}$.

REFERENCES

- [1] J.F. Allen, Towards a general theory of action and time *Art. Intel.* **23** (1984) 123–154.
- [2] C. Artigues and F. Roubellat, A polynomial activity insertion algorithm in a multiresource schedule with cumulative constraints and multiple nodes. *EJOR* **127-2** (2000) 297–316.
- [3] C. Artigues, P. Michelon and S. Reusser, Insertion techniques for static and dynamic resource constrained project scheduling. *EJOR* **149** (2003) 249–267.
- [4] K. R. Baker, *Introduction to Sequencing and Scheduling*. Wiley, NY (1974).
- [5] P. Baptiste, *Resource constraints for preemptive and non preemptive scheduling*. MSC Thesis, University Paris VI (1995).
- [6] P. Baptiste, Demasse, Tight LP bounds for resource constrained project scheduling. *OR Spectrum* **26** (2004) 11.
- [7] S. Benzer, On the topology of the genetic fine structure *Proc. Acad. Sci. USA* **45** (1959) 1607–1620.
- [8] C. Berge, *Graphes et Hypergraphes*. Dunod Ed., Paris (1975).
- [9] J. Blazewicz, K.H. Ecker, G. Schmlidt and J. Weglarz, *Scheduling in computer and manufacturing systems*. 2th edn, Springer-Verlag, Berlin (1993).
- [10] K.S. Booth and J.S. Lueker, Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms. *J. Comp. Sci.* **13** (1976) 335–379.

- [11] P.Brucker and S. Knust, A linear programming and constraint propagation based lower bound for the RCPSP. *EJOR* **127** (2000) 355–362.
- [12] P. Brucker, S. Knust, A. Schoo and O. Thiele, A branch and bound algorithm for the resource constrained project scheduling problem. *EJOR* **107** (1998) 272–288.
- [13] J. Carlier and P. Chretienne, *Problèmes d’ordonnements : modélisation, complexité et algorithmes*. Masson Ed., Paris (1988).
- [14] M. Carter, A survey on practical applications of examination timetabling algorithms. *Oper. Res.* **34** (1986) 193–202.
- [15] M. Chein and M. Habib, The jump number of Dags and posets. *Ann. Discrete Math.* **9** (1980) 189–194.
- [16] E. Demeulemeester and W. Herroelen, New benchmark results for the multiple RCPSP. *Manage. Sci.* **43** (1997) 1485–1492.
- [17] J. Damay, *Techniques de resolution basées sur la programmation linéaire pour l’ordonnement de projet*. Ph.D. Thesis, Université de Clermont-Ferrand, (2005).
- [18] J. Damay, A. Quilliot and E. Sanlaville, Linear programming based algorithms for preemptive and non preemptive RCPSP. *EJOR* **182** (2007) 1012–1022.
- [19] K. Djellab, Scheduling preemptive jobs with precedence constraints on parallel machines. *EJOR* **117** (1999) 355–367.
- [20] D. Dolev and M.K. Warmuth, Scheduling DAGs of bounded heights. *J. Algor.* **5** (1984) 48–59.
- [21] P. Duchet, *Problèmes de représentations et noyaux*. Thèse d’Etat Paris VI (1981).
- [22] B. Dushnik and W. Miller, Partially ordered sets. *Amer. J. Math.* **63** (1941) 600–610.
- [23] D.R. Fulkerson and J.R. Gross, Incidence matrices and interval graphs. *Pac. J. Math.* **15** (1965) 835–855.
- [24] S.P. Ghosh, File organization: the consecutive retrieval property. *Comm. ACM* **9** (1975) 802–808.
- [25] R.L. Grahamson, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnoy-Khan, Optimization and approximation in deterministic scheduling: a survey. *Ann. Discrete Math.* **5** (1979) 287–326.
- [26] J. Josefowska, M. Mika, R. Rozycki, G. Waligora and J. Weglarz, An almost optimal heuristic for preemptive Cmax scheduling of dependant task on parallel identical machines. *Annals Oper. Res.* **129** (2004) 205–216.
- [27] W. Herroelen, E. Demeulemeester and B. de Reyck, A classification scheme for project scheduling, in *Project Scheduling: recent models, algorithms and applications*. Kluwer Acad Publ. (1999) 1–26.
- [28] D.G. Kindall, Incidence matrices, interval graphs and seriation in archaeology, *Pac. J. Math.* **28** (1969) 565–570.
- [29] R. Kolisch, A. Sprecher and A. Drexel, Characterization and generation of a general class of resource constrained project scheduling problems, *Manage. Sci.* **41**, (10), (1995) 1693–1703.
- [30] L.T. Kou, Polynomial complete consecutive information retrieval problems. *SIAM J. Comput.* **6** (1992) 67–75.
- [31] E.L. Lawler, K.J. Lenstra, A.H.G. Rinnoy-Kan and D.B. Schmoys, Sequencing and scheduling: algorithms and complexity, in *Handbook of Operation Research and Management Sciences, Vol 4: Logistics of Production and Inventory*, edited by S.C. Graves, A.H.G. Rinnoy-Kan and P.H. Zipkin, North-Holland, (1993) 445–522.
- [32] F. Luccio and F. Preparata, Storage for consecutive retrieval. *Inform. Processing Lett.* **5** (1976) 68–71.
- [33] A. Mingozzi, V. Maniezzo, S. Ricciardelli and L. Bianco, An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation. *Manage. Sci.* **44** (1998) 714–729.
- [34] R.H. Moring and F.J. Rademacher, Scheduling problems with resource duration interactions. *Methods Oper. Res.* **48** (1984) 423–452.
- [35] A. Moukrim and A. Quilliot, Optimal preemptive scheduling on a fixed number of identical parallel machines. *Oper. Res. Lett.* **33** (2005) 143–151.

- [36] A. Moukrim and A. Quilliot, A relation between multiprocessor scheduling and linear programming. *Order* **14** (1997) 269–278.
- [37] R.R. Muntz and E.G. Coffman, Preemptive scheduling of real time tasks on multiprocessor systems. *J.A.C.M.* **17** (1970) 324–338.
- [38] C.H. Papadimitriou and M. Yannakis, Scheduling interval ordered tasks. *SIAM J. Comput.* **8** (1979) 405–409.
- [39] J.H. Patterson, A comparizon of exact approaches for solving the multiple constrained resource project scheduling problem. *Manage. Sci.* **30** (1984) 854–867.
- [40] A. Quilliot and S. Xiao, Algorithmic characterization of interval ordered hypergraphs and applications. *Discrete Appl. Math.* **51** (1994) 159–173.
- [41] N. Sauer and M.G. Stone, Rational preemptive scheduling. *Order* **4** (1987) 195–206.
- [42] N. Sauer and M.G. Stone, Preemptive scheduling of interval orders is polynomial. *Order* **5** (1989) 345–348.
- [43] A. Schrijver, *Theory of Linear and Integer Programming*. Wiley, NY (1986).
- [44] P. Van Hentenryk, *Constraint Programming*. North Holland (1997).