

# Compilation de problèmes d’ordonnancement à l’aide de groupes d’opérations permutables

Louis Riviere<sup>1,2,3</sup>, Christian Artigues<sup>1,2</sup> and H el ene Fargier<sup>1,3</sup>

<sup>1</sup> Artificial and Natural Intelligence Toulouse Institute, Universit e de Toulouse, France

<sup>2</sup> LAAS-CNRS, Universit e de Toulouse, CNRS, UPS, Toulouse, France

<sup>3</sup> IRIT, Universit e de Toulouse, CNRS, UPS, Toulouse, France

**Mots-cl es :** *Ordonnancement, compilation de connaissances, op erations permutables*

## 1 Introduction

Les **probl emes d’ordonnancement** sont g en eralement sujets aux **al eas**, ce qui peut d egrader la qualit e d’une solution calcul ee   l’avance sur des donn ees d eterministes. Etant donn ee la complexit e de calcul de ces solutions, il n’est en g en eral pas possible de corriger en **temps r eel** la solution de fa on optimale. Certaines approches pour l’ordonnancement sous incertitude (dites proactives-r eactives) ne se contentent pas de calculer   l’avance un ordonnancement unique, mais proposent une solution construite pour pouvoir s’adapter en temps r eel aux al eas. Il s’agit alors de trouver un compromis entre la qualit e de la solution calcul ee et l’effort n ecessaire   son calcul.

Dans ce papier, nous tentons d’ clairer une de ces approches proactives-r eactives sous la lumi ere du formalisme de la **compilation de connaissances**. La compilation de connaissance est un domaine de recherche dont l’objectif est de compiler des probl emes d’un langage source   un langage cible, pour ensuite permettre le calcul plus rapide d’op erations (ou requ etes) en temps r eel. Nous  tudierons les possibilit es et les int er ets de la compilation du probl eme  $1|prec; r_i; \tilde{d}_i|$  vers le langage des listes de groupe d’op erations permutables [1], en posant les bases d’une **carte de compilation** pour les probl emes d’ordonnancement.

## 2 Le langage du probl eme

Le probl eme  $PB = 1|prec; r_i; \tilde{d}_i|$  est le probl eme consistant en la recherche d’un ordonnancement r ealisable  tant donn e un ensemble  $N$  de t aches sur une seule machine avec  $P \in \mathbb{R}^{+|N|}$  la dur ee des t aches,  $R \in \mathbb{R}^{+|N|}$  leurs dates de disponibilit e,  $D \in \mathbb{R}^{+|N|}$  leurs dates limite d’ex ecution et  $E \subseteq N^2$  l’ensemble des contraintes de pr ec edence liant ces t aches.

Les **variables de d ecisions d’ordonnancement** telles que  $\vec{x}_i = t$  ssi la t ache  $i$  commence au temps  $t$  permettent d’exprimer une solution  $\vec{x}$  de ce probl eme.

Ce probl eme peut  tre vu comme le langage  $L_{PB}$  (d’apr es le formalisme de [2]) :

- Syntaxe :  $\Phi_{PB}$  l’ensemble des formules (instances)  $\varphi = (P, R, D, E)$ .
- S emantique : Associe   chaque  $\varphi$  la fonction  $f_\varphi^{L_{PB}} : D_{\chi_{sched}} \rightarrow \mathbb{B}$  tq  $f_\varphi^{L_{PB}}(\vec{x}) = \top$  ssi

$$\begin{aligned} r_i &\leq \vec{x}_i \leq d_i - p_i && \forall i \in N \\ \vec{x}_i + p_i &\leq \vec{x}_j \vee \vec{x}_j + p_j \leq \vec{x}_i && \forall (i, j) \in N^2, i \neq j \\ \vec{x}_i &< \vec{x}_j && \forall (i, j) \in E \end{aligned}$$

En d’autres termes  $f_\varphi^{L_{PB}}$  d etermine si un ordonnancement est solution du probl eme. Ces fonctions bool eennes (ou ensembles de solutions) sont nos objets d’int er et, que l’on cherche   exprimer dans diff erents langages.

### 3 Requêtes

Les requêtes correspondent aux opérations que nous souhaitons effectuer sur nos fonctions, quelqu'ait le langage dans lesquelles elles sont exprimées, pour ainsi pouvoir comparer la complexité de ces requêtes dans différents langages.

Nous considérons dans ce papier quelques requêtes correspondant à certains cas d'application :

- Consistance(CO) et Extraction (EX) : Extraire, si elle existe, une solution au problème.
- Conditionnement (CD) : Modifier l'ensemble des solutions pour refléter une décision.
- Augmentation d'une date de disponibilité ( $R\uparrow$ ), ajout d'une contrainte de précédence ( $E\uparrow$ ) : Restreindre l'ensemble des solutions pour refléter la nouvelle contrainte.

Les requêtes CO, EX et CD sont nécessaires au suivi des décisions d'ordonnement en temps réel et à la prise de décision. Les requêtes  $R\uparrow$  et  $E\uparrow$  permettent de prendre en compte certains aléas, mais beaucoup d'autres requêtes peuvent être utiles en fonction du cas d'application.

### 4 Le langage cible

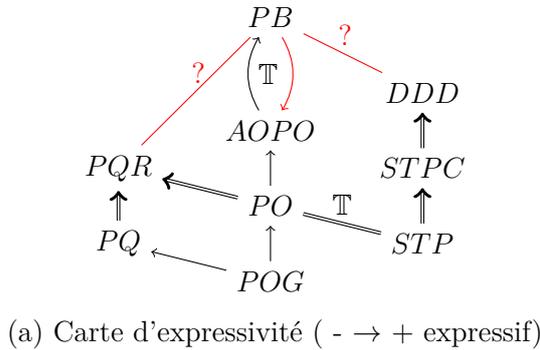
Les séquences de groupes d'opérations permutables (POG) permettent la représentation compacte de larges ensemble de séquences, mais ne sont pas suffisamment expressives. Dans ce papier, nous nous intéressons donc au langage cible que sont les ensembles de POG (SPOG). Ce langage se base sur des **variables de décision de séquence** telles que  $\vec{y}_i = j$  iff la tâche  $i$  est en position  $j$  pour exprimer une séquence solution  $\vec{y}$ . Il peut être défini comme :

- Syntaxe :  $\Phi_{SPOG}$  Chaque SPOG  $\Pi$  est un ensemble de POG.
- Sémantique :  $I_{SPOG}$  : Chaque SPOG  $\Pi$  est associé à la fonction  $f_{\Pi}^{LSPOG} : D_{\chi_{seq}} \rightarrow \mathbb{B}$  telle que  $f_{\Pi}^{LSPOG}(\vec{y}) = \top$  ssi  $\vec{y}$  est représentée par l'un de POG de  $\Pi$ .

### 5 Comparaison des langages et résultats

Les différents langages sont comparés selon trois critères : leur **expressivité** (un langage est plus expressif qu'un autre s'il permet de représenter plus d'objets), leur **compacité** (un langage est moins succinct que l'autre s'il existe des objets pour lequel sa représentation est exponentiellement plus grande que celle dans l'autre langage) et la **satisfaction des requêtes** (un langage satisfait une requête si un algorithme peut y répondre en temps polynomial).

La figure suivante présente l'état de travaux en cours. La figure 1b présente les résultats de complexité pour différentes requêtes (seulement pour les deux langages décrits). La figure 1a présente les résultats d'expressivité pour différents langages en cours d'étude.



Requête	$L_{PB}$	$L_{SPOG}$
CO	o	✓
EX	o	✓
CD( $y_0$ )	✓	✓
$R\uparrow$	✓	•
$E\uparrow$	✓	•

(b) Carte des requêtes

✓ : Requête satisfaite ; • : pas satisfaite ;  
o : pas satisfaite, sauf si P=NP

## Références

- [1] J. Erschler et F. Roubellat An Approach to Solve Workshop Real Time Scheduling Problems, *Advanced Information Technologies for Industrial Material Flow Systems* p. 651, 1989.
- [2] H. Fargier, P. Marquis, et A. Niveau Towards a Knowledge Compilation Map for Heterogeneous Representation Languages *IJCAI* p.877, 2013