



Mixed-Integer/Linear and Constraint  
Programming Approaches for a Multi-Skill  
Project Scheduling Problem with Partial  
Preemption

---

Oliver Polo Mejía, Christian Artigues and Pierre Lopez

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 18, 2020

# Approches par PLNE et PPC pour un problème d’ordonnancement partiellement préemptif

Oliver Polo Mejía, Christian Artigues, Pierre Lopez

LAAS-CNRS, Université de Toulouse, CNRS, France

{oliver.polo-mejia, christian.artigues, pierre.lopez}@laas.fr

**Mots-clés :** *Ordonnancement de projet multi-compétences, Prémption partielle, Programmation linéaire en nombres entiers, Programmation par contraintes.*

## 1 Contexte

Les problèmes d’ordonnancement préemptif supposent que toutes les ressources sont libérées pendant les périodes d’interruption des tâches et qu’elles peuvent être utilisées pour effectuer d’autres activités. Toutefois, dans certains cas, des contraintes imposent qu’un sous-ensemble de ressources reste affectées à l’activité lorsqu’elle a été interrompue, par exemple pour assurer la sécurité. Supposons que l’on doive exécuter une activité qui suppose une atmosphère inerte pour son exécution. En pratique, on peut arrêter cette activité et permettre à des techniciens et à une partie de l’équipement d’être utilisés pour d’autres activités. Cependant, les contraintes de sécurité et d’exploitation peuvent obliger à préserver l’atmosphère inerte même lorsque l’activité est arrêtée (avant sa fin). En d’autres termes, on ne peut pas libérer l’équipement qui assure l’atmosphère inerte pendant les périodes de prémption. Les modèles traditionnels d’ordonnancement préemptif ne peuvent pas représenter ce contexte puisqu’ils supposent que toutes les ressources sont libérées pendant les périodes de prémption. Jusqu’à présent, la seule façon de modéliser cette activité, tout en respectant les exigences de sécurité, était de la déclarer “non préemptive”. Cependant, cette décision peut augmenter la durée du projet, surtout lorsque les activités ont des fenêtres temporelles serrées et que la disponibilité/capacité des ressources varie dans le temps.

La possibilité de ne libérer qu’un sous-ensemble de ressources pendant les périodes de prémption est ce que nous appelons la *prémption partielle*. Nous nous intéressons ici au problème d’ordonnancement de projets multi-compétences (MSPSP) [1] et à l’étude d’une nouvelle variante du MSPSP qui utilise le concept de prémption partielle (MSPSP-PP). A notre connaissance, ce problème n’a pas encore été étudié dans la littérature scientifique.

Notre objectif dans le MSPSP-PP est de trouver un ordonnancement réalisable qui minimise la durée totale du projet. Trouver une solution consiste à déterminer les périodes pendant lesquelles chaque activité est exécutée et aussi les ressources qui exécuteront l’activité dans chaque période, tout en respectant la capacité des ressources et les caractéristiques des activités. Nous devons programmer ces activités sur des ressources renouvelables à capacité limitée ; il peut s’agir de ressources mono-spécialisées cumulatives (machines ou équipements) ou de ressources multi-compétences disjonctives maîtrisant des compétences (techniciens). Les ressources polyvalentes peuvent répondre à plus d’une exigence de compétences par activité et peuvent l’exécuter partiellement (sauf pour les activités non interruptibles où les techniciens doivent exécuter l’ensemble de l’activité). Une activité est définie par sa durée, ses relations de précédence, ses besoins en ressources, ses besoins en compétences, le nombre minimum de techniciens nécessaires à son exécution et le sous-ensemble des ressources préemptives. Une fenêtre temporelle peut être associée à l’exécution d’une activité. La figure 1 illustre un exemple d’instance de MSPSP-PP et une solution réalisable.

	Duration	Skill needed	Machine	Dead line	Release date	Activity type
Act 1	4	(S1,1)	(M1,1)	-	-	Preemptive
Act 2	2	(S3,1),(S4,1)	(M1,1)	5	3	Non-preemptive
Act 3	4	(S2,1)	(M1,1)	-	-	Partially preempt (M1 can not be released)

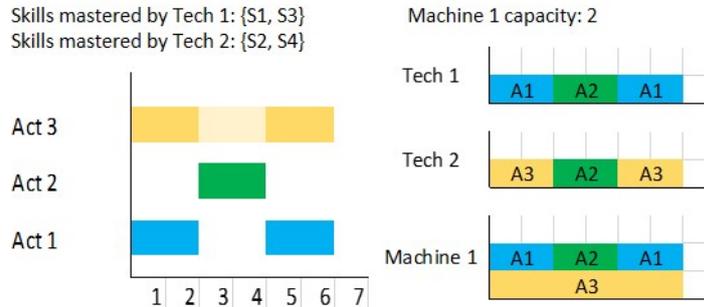


FIG. 1 – Exemple d’une instance de MSPSP-PP.

## 2 Complexité, approches, résultats

La complexité du MSPSP-PP peut être établie en prenant comme point de départ le RCPSP classique. Pour chaque instance du RCPSP, nous pouvons faire correspondre une instance du MSPSP-PP, où toutes les ressources sont mono-compétence, et aucune des ressources ne peut être préemptée. Ainsi, nous pouvons définir le RCPSP comme un cas particulier du MSPSP-PP. Le RCPSP étant NP-difficile au sens fort [2], le MSPSP-PP est aussi NP-difficile au sens fort.

Pour résoudre le MSPSP-PP, nous proposons deux formulations, l’une basée sur la programmation linéaire en nombres entiers, l’autre basée sur la programmation par contraintes. L’approche PLNE utilise une formulation indicée par le temps et impliquant un ensemble de variables binaires associées à la préemption. L’approche PPC s’appuie sur le concept de variable d’intervalle de CP Optimizer.

Des expériences ont été menées pour valider et comparer les approches. Nous utilisons CPLEX 12.7 et CP Optimizer 12.7 [3] pour résoudre respectivement les modèles PLNE et PPC. Nous avons généré quatre ensembles de problèmes, chacun d’eux de 50 instances, se différenciant par la proportion d’activités interruptibles (taux de préemption).

L’analyse des résultats montre que la PLNE est meilleure que la PPC lorsque le pourcentage d’activités interruptibles est élevé. En revanche, la PPC donne de meilleurs résultats lorsque ce pourcentage est faible. Les deux approches sont donc complémentaires.

## Références

- [1] Odile Bellenguez-Morineau. *Méthodes de résolution pour un problème de gestion de projet multi-compétence*. PhD thesis, Université François Rabelais, Tours, 2006.
- [2] Jacek Błazewicz, Jan Karel Lenstra, and Alexander H.G. Rinnooy Kan. Scheduling subject to resource constraints : classification and complexity. *Discrete applied mathematics*, 5(1) :11–24, 1983.
- [3] Philippe Laborie, Jérôme Rogerie, Paul Shaw, and Petr Vilím. IBM ILOG CP optimizer for scheduling. *Constraints*, 23(2) :210–250, 2018.