

AMS : PROJET INTÉGRATEUR

**Ordonnancement sur une machine unique :  
étude comparative de formulations MIP**

Xavier GANDIBLEUX

## 1 Contexte

Les problèmes d'ordonnancement d'ateliers représentent une classe importante de problèmes d'optimisation discrets. Ils permettent de modéliser énormément de situations réelles rencontrées dans les systèmes de production de biens et services. On s'intéressera au cas particulier dans lequel le problème comporte une unique machine qui est toujours disponible. Les travaux à réaliser sur la machine s'appellent des tâches. La machine considérée peut traiter qu'une seule tâche à la fois ; on dit que la machine a une capacité unaire. Lorsqu'une tâche est engagée sur la machine, elle ne peut pas être interrompue ; on dit que les tâches sont non-préemptives. Des contraintes spécifiques peuvent apparaître, comme par exemple des dates de disponibilité des tâches ; on dit que l'on est en présence de *release date*  $r_i$ . Les problèmes d'ordonnancement disposent de fonctions objectif, communément appelés critères, qui leur sont spécifiques : minimiser l'avance, minimiser le retard, minimiser le temps passé dans l'atelier, minimiser le nombre de tâches en retard, etc. Des poids peuvent accompagner les tâches, ceux-ci intervenant dans l'évaluation du critère ; on parle de variante pondérée du problème d'optimisation. Un séquençement donne l'ordre de passage des tâches sur la machine. Un ordonnancement est un séquençement pour lequel la date d'engagement de la tâche sur la machine est fixée. Résoudre un problème d'ordonnancement consiste à établir l'ordonnancement de valeur optimale sur le critère considéré.

## 2 Mission

La configuration de l'atelier, les contraintes à prendre en compte et le critère retenu particularisent le problème d'ordonnancement à considérer. Ce dernier s'exprimera par la notation de Graham, qui consiste en un triplet  $\alpha \mid \beta \mid \gamma$ , chaque élément représentant respectivement la configuration de l'atelier ( $\alpha$ ), les contraintes ( $\beta$ ), le critère ( $\gamma$ ). On parle par exemple de l'optimisation du retard pondéré d'un problème d'ordonnancement sur une machine sans contrainte, qui se représente par  $1 \mid . \mid T_W$ . Les problèmes d'ordonnancement forment une classe particulière de problèmes d'optimisation. Les problèmes étudiés seront exprimés à l'aide du formalisme de la programmation mathématique, on est donc en présence d'un modèle MIP. Il sera donc envisageable de rechercher une solution optimale au problème d'ordonnancement en ayant recours à un solveur MIP comme GLPK ou encore Gurobi. Il n'existe pas en général une formulation MIP unique d'un problème donné. Une formulation peut s'avérer meilleure qu'une autre lorsque l'on examine sa relaxation linéaire, et par conséquent l'arbre de recherche conduisant à l'identification d'une solution optimale par un solveur MIP. Après avoir pris en main cette classe de problème d'optimisation, l'objectif principal de ce projet intégrateur est d'étudier l'impact de la formulation d'un même problème d'ordonnancement sur une machine. Selon l'avancement, plusieurs problèmes pourront être investigués, mais au moins le problème  $1 \mid . \mid T_W$  sera considéré. Les arguments seront notamment construits expérimentalement au regard d'une collection variée de d'instances numériques. Ces résultats expérimentaux seront obtenus à l'issue de résolutions mises en oeuvre sous le langage de programmation Julia et le langage algébrique JuMP. Les solveurs MIP seront GLPK et Gurobi, respectivement solveur open source sous licence GNU et solveur commercial pour lequel le monde académique peut obtenir une licence gratuite pour mener des travaux de recherche.

### 3 Aptitudes scientifiques mobilisées

Ce projet intégrateur mobilise des connaissances et aptitudes scientifiques sur les structures de données avancées, sur la modélisation (formalisme de graphes ; formalisme de programmation mathématique), sur l'optimisation des modèles élaborés (algorithmes génériques ou algorithmes spécifiques), sur la représentation d'un MIP sous un langage algébrique, sur le codage efficace d'algorithmes à l'aide d'un langage de programmation scientifique, sur l'utilisation utilisation de solveurs (GLPK ou Gurobi), sur la mise à disposition en open-source d'une solution logicielle opérationnelle.