

Rapport sur la thèse
The group cumulative scheduling problem

présentée par **Lucas Groleaz**

pour obtenir le grade de Docteur de L'Université de Lyon

Christian Artigues

May 19, 2021



Le manuscrit de thèse présenté par Lucas Groleaz, rédigé en anglais, comporte 150 pages, dont un résumé en français, une introduction, une liste de notations, et 8 chapitres répartis en trois parties, plus un chapitre de conclusion et les références bibliographiques. La thèse de Lucas Groleaz concerne la résolution d'un problème d'ordonnancement à machines parallèles provenant d'une application de préparation de commandes rencontrée par la société Infologic dans le cadre de son progiciel de gestion intégrée (ERP - *Enterprise Resource Planning*) Copilote. L'objectif principal de la thèse est l'adaptation des méthodes d'ordonnancement de la littérature à une contrainte inédite provenant de cette application : la contrainte cumulative de groupes. Deux objectifs secondaires visent l'un de proposer un outil de sélection de l'algorithme le plus adapté selon le problème rencontré et l'autre de prendre en compte les aspects dynamiques de l'ordonnancement.

L'introduction générale pose ainsi le contexte des ERP, et la difficulté de proposer pour un outil générique une méthode adaptée à tous les cas d'applications rencontrés, puis le problème cumulatif de groupes. Le plan de la thèse est présenté ainsi que les principales publications.

La première partie de la thèse propose une revue de littérature. Elle comporte deux chapitres.

Le premier chapitre, intitulé "Définitions", situe la thèse au sein des problèmes d'optimisation sous contraintes puis présente les problèmes d'ordonnancement en se basant sur la notation à trois champs de Graham. La thèse considère ainsi des problèmes à machines parallèles (identiques ou bien à vitesses différentes) avec possiblement des périodes d'indisponibilités, des dates de lancement et de livraison, des temps de préparation dépendant de la séquence des tâches. Les fonctions objectifs à minimiser sont soit la durée de l'ordonnancement (*makespan*), soit la somme des retards vrais (*tardiness*). Des éléments de complexité algorithmique sont ensuite présentés.

Les notations utilisées pour les problèmes d'ordonnancement sont parfois un peu déroutantes (comme B_j pour la date de début de la tâche j). On aurait aimé trouver une présentation ou au moins une mention dès ce chapitre de la contrainte cumulative qui n'est autre que la contrainte de ressource renouvelable présentée au sein du paragraphe sur le RCPSP. Au final, ce chapitre, très bien écrit, comporte les éléments nécessaires à la compréhension de la suite de la thèse et les différents problèmes à machines parallèles sont très bien décrits et illustrés.

Le deuxième chapitre présente différentes approches de résolution de problèmes d'optimisation sous contraintes et leur application à l'ordonnancement. L'accent est mis à juste titre sur les approches utilisées dans la thèse : les algorithmes gloutons et en particulier les algorithmes de liste, l'optimisation par colonies de fourmis, la recherche locale, la programmation linéaire en variables mixtes et la programmation par contraintes.

D'autres méthodes sont très succinctement mentionnées à la fin mais auraient peut être mérité un peu plus d'attention : notamment les approches par encodage SAT qui finalement peuvent être vues comme des approches de programmation par contraintes avec un langage réduit mais qui

sont maintenant incontournables de part leur succès sur les problèmes d’ordonnancement. Il en va de même du principe de recherche dirigée par les conflits qui sont à la base des approches SAT mais aussi d’autres approches par recherche arborescente. Pour la partie programmation linéaire, il manque aussi la mention des modèles à variables indexées par le temps (qui donneraient sans doute des formulations de tailles trop importantes pour le problème considéré). J’aurais enfin trouvé pertinent d’explicitier le lien entre l’apprentissage par renforcement et l’optimisation par colonies de fourmis, ce qui permettrait d’éclairer sous cet angle l’approche de résolution hybride proposée dans le chapitre 4 et faire également écho aux techniques d’apprentissage supervisé du chapitre 7. Il faut enfin souligner le soin apporté à la présentation des algorithmes qui sont détaillés en pseudo-code ainsi que la présence d’illustrations bienvenues des différents opérateurs de voisinage.

La deuxième partie de la thèse est consacrée au problème d’ordonnancement cumulatif de groupes (GCSP) et constitue le cœur des contributions théoriques et algorithmiques de la thèse. Elle comporte deux chapitres.



Le troisième chapitre présente précisément le problème GCSP ainsi que trois problèmes proches : le problème cumulatif, le problème de piles ouvertes et le problème d’ordonnancement hiérarchique. Il est montré que malgré la proximité de ces problèmes, il existe des différences significatives qui empêchent d’utiliser directement leurs approches de résolution sur le GCSP, ce qui justifie pleinement l’étude spécifique de ce problème. Les tâches sont pré-réparties en groupes : un groupe correspond par exemple à un ensemble de préparations pour un même client devant être disposées sur la même palette. La contrainte spécifique du GCSP assure qu’à tout moment le nombre de groupes actifs (c’est à dire qu’une tâche du groupe est commencée et qu’il existe une tâche du groupe non terminée) ne dépasse pas une limite prédéfinie. Le chapitre présente ensuite le principal résultat théorique fondamental de la thèse qui souligne l’extrême difficulté du problème : même si les séquences de tâches sont fixées sur chaque machine, trouver une solution qui respecte la contrainte cumulative de groupes est NP-difficile. Le résultat est obtenu par une réduction de PATHWIDTH très bien illustrée. Il s’ensuit que le GCSP appartient à la classe de problèmes tels qu’on ne connaît pas d’algorithme de liste dominant : le problème ne peut pas se ramener à la recherche d’un ordre total des tâches.

Le quatrième chapitre bâtit des méthodes de résolutions cherchant à pallier ce résultat négatif. Tout d’abord, un modèle de programmation par contraintes est proposé suivant le formalisme du solveur d’IBM CP Optimizer en utilisant la contrainte CUMULATIVE standard et la contrainte SPAN qui permet d’indiquer qu’une variable d’intervalle (une groupe) inclut nécessairement un ensemble d’autres variables d’intervalles (les tâches qui le constituent). Pour la recherche locale, il est nécessaire d’avoir des algorithmes qui lient les séquences obtenues par les mouvements classiques de recherche locale aux dates de début car les ordonnancement aux plus tôt ne sont pas dominants. Deux algorithmes gloutons sont proposés : un premier algorithme est basé sur l’algorithme de liste en tentant de réparer les conflits résultant de l’ordonnancement au plus tôt par décalage de certaines tâches d’un groupe après les tâches d’un autre groupe. Il est montré que cet algorithme peut indéfiniment proposer des décalages et un mécanisme de détection de cycle permet d’interrompre l’algorithme ce qui produit un échec. Un autre algorithme glouton permet d’obtenir à coup sûr une solution réalisable en utilisant le concept de groupe ouvert (un groupe partiellement ordonné : lorsque le nombre de groupes ouverts atteint la limite du nombre de groupes actifs en parallèle, les tâches des groupes non ouverts ne sont plus candidates à l’ordonnancement. Ces conditions sont suffisantes pour obtenir une solution réalisable mais ne sont pas nécessaires et donc il n’existe pas forcément de règle de priorité menant à la solution optimale. Les approches par colonies de fourmis sont ensuite adaptées au problème, notamment par la définition de fonctions de phéromones pour apprendre l’intérêt d’ordonner une tâche à une date précise (puisque la séquence où les positions ne suffisent plus pour le GCSP) avec des mécanismes intelligents de limitation de la mémoire utilisée (il n’est pas raisonnable de mémoriser une phéromone par tâche

et unité de temps). Enfin, une approche hybride intégrant la programmation par contraintes et l'optimisation par colonies de fourmis est proposée.

Les algorithmes sont encore une fois présentés avec le plus grand soin et un souci de détail louable pour la reproduction des résultats. Des illustrations viennent bien expliquer les difficultés de la résolution des conflits par les algorithmes gloutons. Un regret vient de la partie résolution de l'approche de programmation par contraintes. On apprend dans les chapitres suivants que les différentes options de recherche du solveur CP Optimizer (*depth first search, restart...*) ont bien été testées mais il aurait peut être été pertinent d'orienter le solveur en lui proposant de brancher d'abord sur l'ordonnancement relatif des macro-tâches représentant les groupes puis sur les tâches élémentaires, ce qui aurait peut être pu éviter la difficulté expliquée de satisfaire la contrainte cumulative de groupes.

La troisième et dernière partie de la thèse présente les expérimentations effectuées dans la thèse. Cette partie, très riche comporte quatre chapitres.



Le cinquième chapitre présente des jeux de données issu de l'application de préparation de commandes rencontrée par la société Infologic dans le cadre de son ERP Copilote déployé dans l'industrie agro-alimentaire. Les jeux de données comportent ainsi jusqu'à plus de 3000 tâches, 14 machines, 462 groupes. L'ordonnancement se fait sur un horizon d'une journée avec des tâches durant de quelques seconde à une heure et demi.

Le sixième chapitre présente de manière très détaillée et illustrée par de nombreux graphiques les comparaisons des différents algorithmes (notamment l'évolution de la fonction objectif en fonction du temps de calcul) sur 10 problèmes : du plus simple, l'ordonnancement à machines parallèles identiques et minimisation du makespan au plus complexe, avec dates de lancement, de livraison, temps de préparation, machines à vitesses différentes et indisponibilité et contrainte cumulative de groupes. Il est rare de trouver dans une thèse de recherche opérationnelle un si grand soin apporté à une comparaison rigoureuse des résultats. Notamment, on peut apprécier le paramétrage très fouillé des différentes approches avec la considération de paramétrages différents selon les combinaisons de composants algorithmiques (dont l'étude des interactions entre le voisinage choisi et la structure de liste tabou ou encore un paramétrage différent selon qu'on utilise les colonies de fourmis seules ou en combinaison avec d'autres algorithmes, l'importance de la diversification augmentant pour l'utilisation en solo). En ce qui concerne les résultats, l'intérêt de l'hybridation de la méthode tabou avec l'algorithme de colonie de fourmis est finalement la meilleure alternative mais le choix du meilleur algorithme dépend clairement du problème considéré : par exemple l'approche hybride colonies de fourmis / programmation par contraintes est très bonne lorsque les temps de préparation sont combinés avec la contrainte cumulative de groupes. La programmation par contraintes s'effondre pour le problème le plus complexe.

Sur la base de cette constatation, le septième chapitre propose une méthode pour sélectionner le meilleur algorithme face à une instance particulière. Ce chapitre correspond également à une demande industrielle face à la difficulté de performance d'un outil générique comme l'ERP Copilote face à un problème particulier qui nécessiterait une expertise pour choisir la méthode de résolution la plus adaptée. L'approche utilise le package R "Llama" pour apprendre, à partir d'un ensemble bien défini et élargi de caractéristiques de l'instance, quel est l'algorithme à sélectionner en utilisant comme mesure de performance la capacité de l'algorithme à atteindre au bout d'une heure de temps de calcul une solution de référence pour l'instance. Différents algorithmes d'apprentissage sont comparés. L'intérêt de cette approche est démontré pour améliorer les résultats d'une méthode prise isolément avec toutefois des limites identifiées laissant la place à beaucoup de progression dans le domaine de la sélection automatique d'algorithmes.

Le huitième et dernier chapitre présente des expérimentations pour résoudre des problèmes d'ordonnancement dynamique. Là encore, ce chapitre répond à un besoin industriel. Les com-

mandes à préparer ne sont pas toutes connues au début de la journée et arrivent au cours du temps. La problématique est donc de réoptimiser la solution proposée lorsqu'une commande arrive sans toutefois perturber les opérateurs en changeant constamment l'ordre des tâches qu'ils doivent réaliser. Un état de l'art de l'ordonnancement dynamique est effectué. Deux événements sont considérés : l'arrivée d'une tâche et la fixation de la date de début d'une tâche. En effet pour éviter de perturber les opérateurs, une tâche dont le début est trop proche doit être figée dans les plannings ultérieurs. L'algorithme hybride colonies de fourmis / programmation par contraintes est utilisé comme algorithme *anytime* : il est capable de partir de la solution courante et de l'améliorer continuellement en réponse à un événement, jusqu'au prochain. Les résultats montrent un impact important de la période de gel du planning mais aussi un écart considérable entre l'approche statique et l'approche dynamique. Pour contrer cet écart, des régularités dans les données laissent finalement présager la possibilité d'anticipation via des modèles stochastiques.

En résumé cette thèse comporte tous les éléments d'une thèse de recherche opérationnelle réussie : l'extraction des éléments importants d'un cas pratique pour aboutir à des modèles mathématiques avec différentes variantes, une étude de complexité non triviale d'un nouveau problème d'ordonnancement pertinent pour l'industrie, des algorithmes de résolution exacts et approchés et une étude expérimentale très fouillée à la fois pour le problème statique et le problème dynamique. Le problème d'ordonnancement cumulatif de groupe est particulièrement intéressant de part son intérêt industriel et sa difficulté de résolution. Il devrait susciter un réel intérêt de la part de la communauté et les approches proposées par Lucas Groleaz constitueront les algorithmes de références sur ce problème. On peut demeurer un peu sur sa faim en ce qui concerne d'autres approches qui auraient pu être tentées : j'ai déjà évoqué les approches SAT et de recherche dirigée par les conflits mais je pense aussi à des méthodes de décomposition du problème, selon par exemple le schéma d'ordonnancement des groupes dans un problème maître et d'ordonnancement des tâches des groupes dans un sous problème. Mais c'est aussi tout l'intérêt d'une thèse de mettre en évidence de nombreuses perspectives comme celles, nombreuses, évoquées dans la conclusion. Je souligne encore une fois le soin apporté aux expérimentations et à la présentation pédagogique des algorithmes.

Il faut enfin noter que deux publications dans des conférences de premier plan (GECCO et CP) ont été réalisées dans le cadre de la thèse.

Pour toutes ces raisons, je recommande sans réserve d'autoriser la soutenance de la thèse de Lucas Groleaz en vue de l'obtention du grade de Docteur de l'Université de Lyon.

Christian Artigues
Directeur de Recherche au LAAS-CNRS
artigues@laas.fr

