Heuristiques pour des problèmes non linéaires en variables mixtes

Stage de recherche

LocalSolver,
24 avenue Hoche, 75008 Paris,
Julien Darlay, Nikolas Stott, Simon Boulmier
jdarlay/nstott/sboulmier@localsolver.com

1 Contexte

LocalSolver est une entreprise de service, de conseil et d'édition dans le domaine de la recherche opérationnelle. Elle développe un solveur programmation mathématique, traitant les problèmes d'optimisation sous contraintes, avec des variables continues, discrètes (ou ensemblistes) :

$$\min_{x,y} f(x,y) \text{ tels que } \begin{cases} g_i(x,y) = 0\,, \forall i\,, \\ x \in \mathbb{R}^p\,, y \in \mathbb{Z}^q \end{cases}.$$

Le solveur exploite de nombreux outils numériques et algorithmiques tels que la recherche locale, l'optimisation non linéaire, ou encore la programmation par contraintes. Il est utilisé pour résoudre des problèmes opérationnels dans des domaines variés de l'industrie, tels que les télécommunications, la logistique, les transport, l'énergie ou encore l'aérospatial.

Le calcul de solutions faisables de bonne qualité est une tâche souvent difficile lorsqu'un problème d'optimisation comporte des variables mixtes. Dans ce cas, le sous-problème défini par les variables continues est souvent "piloté" par l'ensemble de variables discrètes. Si le problème est linéaire et que les variables discrètes sont fixées, la résolution du problème continu sous-jacent est simple, mais l'espace de recherche des variables discrètes est de taille exponentielle et donc intraitable en pratique. De nombreuses heuristiques ont ainsi été formulées dans le cadre de la programmation linéaire en nombres entiers (PLNE) : heuristiques d'arrondis de relaxations, résolution de sous-problèmes ciblés, etc. Une difficulté supplémentaire se présente lorsque le problème est non-linéaire, puisque le calcul d'une solution pour le sous-problème continu n'est plus facile et peut nécessiter beaucoup plus d'efforts.

2 Sujet

Ce stage propose d'explorer différentes heuristiques afin de renforcer le calcul de solutions dans LocalSolver pour des problèmes d'optimisations combinant des variables continues et discrètes avec des opérations non-linéaires, en particulier des problèmes de flot non-linéaire et d'attribution énergétique. Plusieurs pistes sont déjà envisagées et sont décrites ci-dessous.

2.1 Réparation de la partie continue

Une heuristique de réparation consiste à choisir la valeur des variables discrètes $y := \widehat{y}$ et à "réparer" la partie continue de la solution en résolvant le sous-problème

$$\min_{x} f(x,\widehat{y}) \text{ tels que } \begin{cases} g_i(x,\widehat{y}) = 0\,, \forall i\,, \\ x \in \mathbb{R}^p \end{cases}.$$

LocalSolver intègre déjà un tel composant lorsque la partie continue est linéaire. La même idée pourrait être appliquée dans le cadre non-linéaire, en utilisant l'un des algorithmes d'optimisation continue déjà présents dans LocalSolver. Deux aspects importants pour l'efficacité d'une telle approche sont :

— la capacité de démarrer la résolution à chaud afin de limiter le coût de résolution,
— l'exploitation de l'infaisabilité ou du caractère non-borné des sous-problèmes pour piloter la recherche discrète.

2.2 Résolution limitée de voisinages

Deux heuristiques majeures en PLNE consistent à résoudre un sous-problème mixte ciblé autour de la solution de la relaxation continue du problème, en contraignant les intervalles des variables discrètes autour de la valeur de la relaxation et en effectuant une résolution limitée de ce sous-problème.

Une idée similaire peut être mise en place ici. On commence par résoudre une relaxation du problème : les fonctions f, g_i sont remplacées par des sous-estimations convexes $\widehat{f}, \widehat{g_i}$ et l'intégrité de y est relaxée.

$$\min_x \widehat{f}(x,y) \text{ tels que } \begin{cases} \widehat{g_i}(x,y) = 0\,, \forall i\,, \\ x \in \mathbb{R}^p\,, y \in \mathbb{R}^q \end{cases}.$$

On note $(\widehat{x}, \widehat{y})$ la solution de ce problème. On s'intéresse alors à une restriction du problème initial dans lequel les bornes des variables y_j ont été réduites autour de la valeur de la relaxation $\widehat{y}_j$:

$$\min_{x,y} f(x,y) \text{ tels que } \begin{cases} g_i(x,y) = 0\,, \forall i\,, \\ x \in \mathbb{R}^p\,, y \in \mathbb{Z}^q \\ \lfloor \widehat{y}_j \rfloor \leq y_j \leq \lceil \widehat{y}_j \rceil \end{cases}.$$

De nombreuses questions se posent ici.

— Y-a-t'il un intérêt à garder la relaxation $\widehat{g_i}$?
— Quels composants du solveur doit-on désactiver pour résoudre ces problèmes réduits avec le meilleur équilibre vitesse/fiabilité possible ?
— A quelle fréquence lance-t'on cette résolution ?
— Quelles limites impose-t'on à cette résolution ?
— Que peut-on apprendre des sous-problèmes qui sont prouvés inconsistants ou sous-optimaux ?

2.3 Détection de structures et heuristiques dédiées

La présence de certaines sous-structures dans les données du problème d'optimisation peut motiver le développement d'heuristiques dédiées. Ces structures peuvent par exemple être :

— un ensemble de booléens activant ou désactivant des structures similaires,
— plus génériquement la présence de couplages de variables/contraintes,
— sous-modèle de flot linéaire/non linéaire.