

Intégration d'un algorithme de points intérieurs pour résoudre des problèmes non-linéaires dans LocalSolver

Nikolas Stott

Localsolver, Paris, France

nstott@localsolver

Mots-clés : *Optimisation numérique, Optimisation non-linéaire, Méthode de points intérieurs*

1 Introduction

Les méthodes de points intérieurs ont été introduites par Karmarkar [3] pour la résolution de programmes linéaires (f et g_i affines dans (LP)) :

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s. c.} \quad g_i(x) \geq 0, \forall i \quad (\text{LP})$$

Ce problème est transformé en déplaçant les contraintes affines dans la fonction objectif, au moyen d'une fonction barrière logarithmique et d'un paramètre $\mu > 0$ pour ajuster sa force, ce qui donne le nouveau problème (avec variables de *slack*) :

$$\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_i \log s_i \quad \text{s. c.} \quad g_i(x) = s_i, s_i \geq 0 \quad (\text{IP}_\mu)$$

Le paramètre de barrière μ est décroissant vers 0 au cours des itérations et la solution optimale du problème initial (LP) comme limite des solutions optimales des problèmes (IP_μ) . En pratique, nous recherchons des points KKT du système (IP_μ) , lorsque μ est petit – c'est à dire des vecteurs x, λ, s tels que

$$\nabla f(x) + \sum_i \lambda_i \nabla g_i(x) = 0 \quad s_i \lambda_i = \mu \quad g_i(x) = s_i \quad s_i \geq 0 \text{ et } \lambda_i \geq 0, \quad (\text{KKT})$$

qui fournissent ainsi des approximations de la solution optimale du problème (LP), lorsque les équations ci-dessus sont satisfaites à ε (10^{-6} ou 10^{-9}) près.

Cette famille d'algorithmes repose sur une méthode du second-ordre (itération de Newton) et converge rapidement en pratique (typiquement moins de 50 itérations). De plus, les variantes les plus efficaces ne nécessitent pas de point faisable pour démarrer l'itération [4], ce qui est rarement à la disposition de l'utilisateur dans le cas de problèmes fortement contraints.

Nous étudions des problèmes d'optimisation où les fonctions f et g_i sont non-linéaires et possiblement non-convexes. La généralisation de ces méthodes à ces problèmes introduit de nombreuses difficultés : possibilité de non-terminaison en restant bloqué dans une région infaisable, instabilité numérique des multiplicateurs λ proche de l'optimum local et une dégradation du conditionnement du système linéaire en présence d'opérations fortement non-linéaires.

2 Implémentation d'une nouvelle approche

Une nouvelle approche par Hinder et Ye [2] propose de *décaler* la barrière dans le sens de l'infaisabilité à l'aide d'un vecteur $w \geq 0$ et de la rapprocher de la vraie barrière lorsque μ tend vers 0, résolvant la suite de problèmes

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) - \mu \sum_i \log [\mu w_i + s_i] \\ \text{s. c.} \quad & g_i(x) = s_i, s_i \geq -\mu w_i \end{aligned} \quad (\text{IPshift}_\mu)$$

Nous avons implémenté cette méthode dans LocalSolver, car elle a trois avantages pratiques.

2.1 Une seule phase

L'algorithme évite les dangers mentionnés dans l'introduction en décidant de réaliser un pas de Newton dans une direction de descente agressive, visant à se rapprocher de l'optimalité, ou une direction stabilisante, cherchant à optimiser le sous-problème à barrière fixée. Nous évitons ainsi les écueils des méthodes bi-phasées, tel l'échec de l'algorithme lorsque la "phase de secours" est appelée trop proche de l'optimum. L'implémentation de ces phases est unifiée et ne se sépare qu'en deux points : choix de la perturbation du système KKT à résoudre et détermination de l'acceptation (ou non) d'un pas.

2.2 Contrôle des erreurs numériques

Le décalage de la barrière permet de contrôler la taille des multiplicateurs de Lagrange, ce qui retire une source d'erreurs numériques. Ce phénomène nous permet également de transformer les contraintes du type $a = b$ en deux contraintes $a \leq b$ et $a \geq b$ sans dégrader le conditionnement du problème, et de traiter ainsi des problèmes généraux.

Les erreurs numériques pouvant apparaître dans l'algorithme sont atténuées par l'utilisation de sommes de Kahan, qui compensent ces erreurs lorsque de nombreuses valeurs flottantes sont sommées.

2.3 Un traitement plus stable des non-convexités

La recherche du pas de Newton est pénalisée quadratiquement par la taille du pas dans la variable primale x , pour compenser l'effet de non-convexités locales. L'algorithme se distingue ainsi de solveurs comme IPOPT [5] et KNITRO [1], qui nécessitent également une pénalisation par la taille du pas dans la variable d'écart s . L'ajustement par un seul paramètre (par recherche linéaire) améliore grandement la stabilité de l'algorithme.

3 Résultats expérimentaux

Nous avons comparé les performances de l'algorithme de points intérieurs avec les autres approches déjà présentes dans LocalSolver sur un benchmark expérimental de 558 instances numériques non-linéaires de petite taille (moins de 250 variables). L'approche par recherche locale (*ordre 0*) résout 61 % des instances, l'algorithme de Lagrangien augmenté (*ordre 1*) en résout 80% et les points intérieurs (*ordre 2*) trouvent une solution localement optimale dans 90% des cas.

Références

- [1] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. *Knitro : An Integrated Package for Nonlinear Optimization*, pages 35–59. Springer US, Boston, MA, 2006.
- [2] O. Hinder and Y. Ye. A one-phase interior point method for nonconvex optimization. *ArXiv e-prints*, January 2018.
- [3] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4) :373–395, Dec 1984.
- [4] Irvin J. Lustig. Feasibility issues in a primal-dual interior-point method for linear programming. *Mathematical Programming*, 49(1) :145–162, Nov 1990.
- [5] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1) :25–57, Mar 2006.