Approximation à précision numérique prédéfinie d'une classe de problèmes d'optimisation non-linéaires non-convexes séparables

Claudio Contardo[1] Sandra Ulrich Ngueveu[2]

[1]School of Business, University of Québec in Montreal, Canada
[2]LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
ngueveu@laas.fr

1 Description du problème

Nous considérons le problème de la résolution du programme non-linéaire en nombres entiers (MINLP) suivant avec un objectif séparable :

$$x \in \arg\min \left\{ \sum_{i=1}^{n} f_i(x_i) : Ax = b, x \in \mathcal{X} \subseteq \mathbb{R}_{n-p} \times \mathbb{N}_p \right\}, \tag{1}$$

où les fonctions $f_i : \mathbb{R} \longrightarrow \mathbb{R}^+, i = 1 \ldots n$ sont dérivables par morceaux. Par souci de simplicité, nous supposons également que le problème est bien défini et qu'il admet au moins une solution. Bien qu'il soit possible de gérer efficacement les problèmes pour certaines formes spécifiques de f_i —à savoir lorsqu'elles sont linéaires, affines, quadratiques et/ou convexes —, les formes générales des fonctions f_i rendent le problème d'optimisation beaucoup moins traitable.

Une façon possible d'approcher le problème (1) est de remplacer les fonctions f_i par des fonctions linéaires par morceaux (PWL) et résoudre le modèle résultant à l'aide d'algorithmes issus de la littérature sur la programmation en nombres entiers (MILP). En choisissant bien les PWL, cette approche peut fournir des garanties sur la qualité des solutions obtenues. Son inconvénient est qu'elle se limite généralement à des précisions numériques très grossières car les petites erreurs d'approximation conduisent à des MILP de très grande taille pouvant devenir difficiles à résoudre. La méthode itérative que nous proposons [3] contribue à y remédier.

2 Travaux connexes

[2] s'appuie sur les travaux de [4] et développe un algorithme itératif pour trouver une solution optimale globale du MINLP ayant des nonlinéarités multivariables en résolvant une série de relaxations MIP avec une précision progressivement croissante, basée sur des PWL qui sont affinées de manière adaptative d'une itération à l'autre. Un élément critique concerne la façon dont les PWL sont définies et leur procédure de raffinement. Les auteurs ont besoin de PWL qui interpolent les f_i aux points de cassure et qui contiennent complètement le graphe de la fonction. Ils montrent que la stratégie de raffinement ajoutant uniquement des points avec une erreur d'approximation maximale sur un simplexe de PWL ne remplit pas des conditions de convergence. En revanche, celle ajoutant un point de rupture de linéarisation au milieu de l'arête la plus longue d'un simplexe si. Les auteurs ne calculent aucune borne supérieure. Ils supposent également qu'il existe un oracle qui optimise la différence entre chaque f_i et sa PWL sur un simplexe, afin de calculer les erreurs de linéarisation résultant des raffinements de la PWL. Un tel oracle peut être un solveur NLP si la formule analytique de la solution n'est pas disponible. Dans le cas où les f_i sont polynomiales, [5] proposent un raffinement adaptatif de PWL qui vise à améliorer la qualité de l'approximation au lieu du raffinement uniforme de [2]. Pour ce faire, au lieu de simplement partitionner chaque morceau de PWL à raffiner en

deux parties égales, [5] le divise en trois, la taille de la portion du milieu étant définie par un paramètre utilisateur. Les auteurs utilisent également un solveur NLP pour calculer des bornes supérieures. L'algorithme résultant est disponible comme solveur en JULIA appelé Alpine.

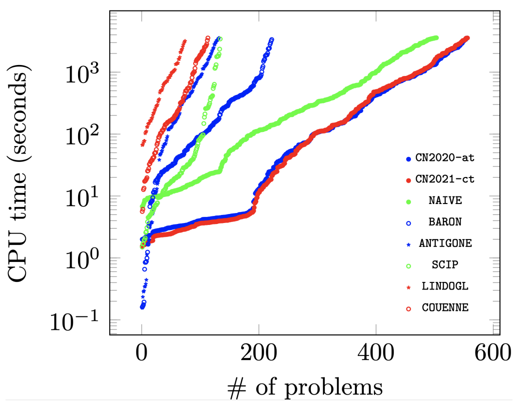
3 Principes de la méthode proposée

Notre méthode itérative [3] vise à résoudre le problème (1) à une tolérance prédéfinie donnée. A l'initiation, elle remplace chaque f_i par une PWL qui la sous-estime avec un nombre de morceaux minimal et une tolérance ϵ_0 supérieure à la tolérance cible ϵ. Une borne duale est calculée après résolution du MILP résultant et une borne primale est calculée en re-évaluant la solution du MILP avec les fonctions f_i. Si la différence entre ces deux valeurs est jugée non satisfaisante, l'approximation est localement raffinée et un nouveau MILP est généré avant résolution. Le raffinement ne concerne que les morceaux de PWL utilisés par la solution du MILP, et se base sur le calcul de la meilleure sous-estimation linéaire par morceaux pour une tolérance plus faible, au lieu d'une partition arbitraire de l'intervalle en deux [2] ou trois [5].

La convergence de la méthode est assurée sous des hypothèses très modérées, et aucun solveur/oracle NLP ou MINLP n'est requis. La méthode est peu sensible à la précision recherchée et peut gérer des tolérances absolues ou relatives, sachant que ces dernières peuvent produire des MILP plus petits pour la même valeur de précision cible. Enfin, nous n'exigeons pas que les PWL soient continues ni d'interpoler les f_i, ce qui peut être exploité efficacement par exemple en utilisant l'algorithme en JULIA LinA [1] pour calculer et resserrer les approximations.

4 Résultats numériques avec une limite de 3600 secondes CPU

Les tests numériques sont effectués, sur des variantes non linéaires des trois problèmes suivants : *transportation*, *capacitated facility location* et *multi-commodity network design*.



Comparaison de [3] vs solveurs MINLP avec même précision 1e-4 sur toutes les instances

Inst.	[5]	CN[3]	Inst.	[5]	CN[3]
2_1	13.43	0.15	3_1	TLIM	29.33
2_2	ERR	0.85	3_2	TLIM	1.1
2_3	25.84	1.05	3_3	TLIM	6.62
2_4	29.34	1.09	3_4	TLIM	4.45
2_5	TLIM	0.22	3_5	TLIM	4.53
2_6	2.13	0.27	3_6	TLIM	11.08
2_7	ERR	0.16	3_7	TLIM	19.37
2_8	1.84	0.16	3_8	TLIM	11.83
2_9	19.92	0.19	3_9	TLIM	2.29
2_10	ERR	0.93	3_10	TLIM	3.95
Moy.	15.4*	0.5	Moy.	TLIM	8.68

CPU(s) d'Alpine ([5]) vs [3] sur de petites instances (2x2 et 3x3) du *transportation problem*

Références

[1] https://homepages.laas.fr/sungueve/piecewiselinearapprox/LinA.html.

[2] R. Burlacu, B. Geißler, and L. Schewe. Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes. *Optimization Methods and Software*, 35(1) :37–64, 2020.

[3] C. Contardo and S. U. Ngueveu. On the approximation of separable non-convex optimization programs to an arbitrary numerical precision. Technical report, LAAS report n°21222, 2021.

[4] B. Geißler, A. Martin, A. Morsi, and L. Schewe. Using piecewise linear functions for solving MINLPs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 287–314, New York, NY, 2012. Springer New York. ISBN 978-1-4614-1927-3.

[5] H. Nagarajan, M. Lu, S. Wang, R. Bent, and K. Sundar. An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs. *Journal of Global Optimization*, 74(4) :639–675, 2019.