Technique de reformulation affine appliquée à l'optimisation globale

Jordan Ninin[1], Frédéric Messine[1], Pierre Hansen[2]

[1] IRIT ; Université de Toulouse ; 2, rue Charle Camichel, 31071 Toulouse, France
{jordan.ninin,frederic.messine}@n7.fr
[2] GERAD ; HEC Monréal ; 3000, chemin de la Côte-Sainte-Catherine, H3T 2A7 Montréal, Canada
LIX ; école Polytechnique ; 91128 Palaiseau, France
pierre.hansen@gerad.ca

Mots-Clés : *arithmétique affine, arithmétique d'intervalles, reformulation linéaire, Branch-and-Bound, optimisation globale.*

1 Introduction

Depuis quelques années, les algorithmes de branch and bound par intervalles ont montré leur intérêt pour la résolution exacte de problèmes d'optimisation globale ; en effet, le caractère robuste aux erreurs numériques de ces algorithmes permet de garantir l'obtention de l'optimum global à la précision souhaitée par l'utilisateur (si cette solution est atteignable par de tels algorithmes de complexité exponentielle). Ces algorithmes ont d'ailleurs prouvé leur efficacité en résolvant de nombreux problèmes de dimensionnement de machines électriques [2]. Malheureusement, ces méthodes nécessitent parfois beaucoup de temps CPU, c'est pourquoi l'intégration de méthode d'accélération s'avère généralement indispensable.

La technique de reformulation que nous proposons, nommée *ART* (pour *Affine Reformulation Technique*), permet de générer de manière automatique un problème linéaire relâché afin d'en déduire un meilleur minorant du problème original ou un certificat d'infaisabilité. Contrairement à la méthode classique de *RLT (Reformulation-Linearization Technique)*, notre technique ne génère aucune variable supplémentaire et la taille du problème linéaire généré reste petite. Ceci est idéal pour l'intégration d'*ART* à chaque itération de notre algorithme de branch and bound par intervalles, nommé *IBBA (pour Interval Branch and Bound Algorithm)* et développé à l'IRIT depuis 1995. Effectivement grâce à la petite taille des problèmes linéaires ainsi générés, la résolution de ceux-ci à chaque itération d'IBBA est très efficace.

2 Technique de Reformulation Affine : *ART*

Considérons un problème d'optimisation globale non convexe (1), notre technique de reformulation affine *ART* consite à générer de manière automatique le problème linéaire (2) de tel sorte qu'il existe une transformation affine T de $X \subset \mathbb{R}^n$ dans $[-1,1]^n$ tel que pour tout x solution réalisable du problème (1), $T(x)$ soit une solution réalisable du problème (2).

$$\left\{\begin{array}{lll} \min\limits_{x\in X\subset\mathbb{R}^n} & f(x) & \\ \text{s.t.} & g_k(x)\leq 0\ , & \forall k\in\{1,\ldots,p\}, \\ & h_l(x)=0\ , & \forall l\in\{1,\ldots,q\}. \end{array}\right. \quad (1) \qquad \left\{\begin{array}{ll} \min\limits_{y\in[-1,1]^n} & c^T y \\ \text{s.t.} & Ay\leq b. \end{array}\right. \quad (2)$$

Notre approche est basée sur des extensions de l'arithmétique affine, notées *AF1* et *AF2* [1]. Ces nouvelles arithmétiques affines ainsi générées sont des extensions de l'arithmétique d'intervalles standard, obtenue en considérant des formes affines à la place d'intervalles. Notons $\widehat{f}$ la forme affine *AF1* d'une fonction f alors :

$$\widehat{f}(x) = f_0 + \sum_{i=1}^{n} f_i \epsilon_i + f_{n+1}\epsilon_{\pm},$$
$$\text{avec } \forall i \in \{0; n\}, f_i \in \mathbb{R}\ ,\ f_{n+1} \in \mathbb{R}^+, \forall i \in \{1; n\},\ \epsilon_i = [-1, 1] \text{ and } \epsilon_{\pm} = [-1, 1].$$

Cette arithmétique permet par surcharge des opérateurs de générer une forme affine d'intervalles de n'importe quelle fonction différentiable. Ainsi en utilisant la notion de fonction d'inclusion, la proposition suivante peut en être déduite et nous permet de construire de manière automatique la relaxation linéaire (2) du problème (1).

Proposition *Considérons* $(f_0, \ldots, f_{n+1})$ *la reformulation de* f *sur* X *utilisant AF1, ainsi si* $\forall x \in X, f(x) \leq 0$ *alors* $\forall y \in [-1, 1]^n$, $\sum_{i=1}^{n} f_i y_i \leq f_{n+1} - f_0$.
et si $\forall x \in X, f(x) = 0$ *alors* $\forall y \in [-1, 1]^n$, $\sum_{i=1}^{n} f_i y_i \leq f_{n+1} - f_0$ *et* $-\sum_{i=1}^{n} f_i y_i \leq f_{n+1} + f_0$.

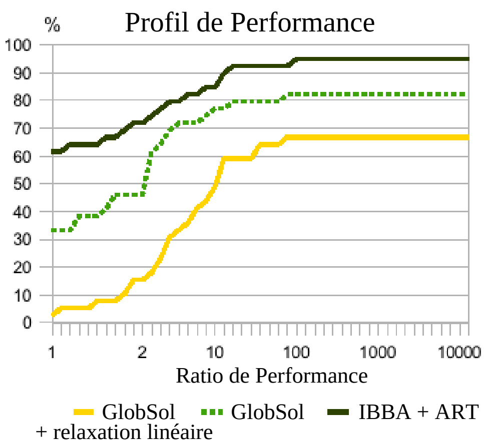
Ainsi, en intégrant cette génération automatique à chaque itération de notre algorithme *IBBA*, nous pouvons déduire les deux propositions suivantes après résolution du problème linéaire (2) :

– Si le problème linéaire (2) relâché du problème (1) n'a pas de solution réalisable, alors le problème (1) n'a pas de solution réalisable.
– Soit y_{sol} une solution qui minimise le problème linéaire (2), alors pour toute solution réalisable x du problème (1), $f(x) \geq c^T y_{sol} + f_0 - f_{n+1}$.

De plus, cette technique peut devenir complètement robuste aux erreurs numériques et être étendue pour prendre en considération des variables binaires.

3 Résultats numériques

74 tests numériques extraits de la librairie 1 du site internet de COCONUT ont été réalisés. 64 d'entre eux ont été résolus avec un temps moyen d'environ 1min 50s, et 32 ont été résolus pour la première fois avec une borne certifiée sur l'erreur relative de 10^{-8}. De plus, cet échantillon comprend 39 exemples pour lesquels l'algorithme *GlobSol* a récemment été appliqué. Celui-ci a trouvé des solutions dans 32 cas. Notre méthode permet de résoudre 31 de ces 32 cas et 5 de plus, soit un total de 37 sur 39 avec un temps CPU ne dépassant jamais 2 minutes. Le profil de performance de ces résultats est représenté ci-contre (cf E. Dolan et J. Moré), nous y comparons notre algorithme *IBBA* avec *ART* et *GlobSol* avec et sans sa relaxation linéaire [3].



Références

[1] F. Messine, *Extentions of affine arithmetic : Application to unconstrained global optimization*, Journal of Universal Computer Science, Vol. 8, pp. 992–1015, 2002.

[2] F. Messine, *A Deterministic Global Optimization Algorithm for Design Problems*, chapter in Essays and Surveys in Global Optimization, pp. 267–294. Springer US, 2005.

[3] R.B. Kearfott, *Discussion and empirical comparisons of linear relaxations and alternate techniques in validated deterministic global optimization*, Optimization Methods and Software, Vol. 21, No. 5, pp. 715–731, 2006.