

Article invité

Introduction à l'optimisation globale

Marcel Mongeau ¹

mongeau@cict.fr

Nous présentons ici le problème d'optimisation globale en nous concentrant plus particulièrement sur les problèmes du type boîte noire et survolons les principales méthodes pour l'aborder.

1 Généralités

On considère le problème d'optimisation général :

$$\min_{x \in S} f(x)$$

où S est un ensemble de candidats satisfaisant certaines contraintes dits *solutions réalisables* et la fonction f , dite *fonction-objectif* est définie sur S .

Les méthodes mathématiques traditionnelles d'optimisation numérique se contentent de chercher un minimum local x^* , c'est-à-dire une solution réalisable telle que $f(x^*) \leq f(x)$, pour tout x dans un voisinage de x^* . Le problème d'*optimisation globale* est le «vrai» problème d'optimisation : trouver une solution réalisable x^* telle que $f(x^*) \leq f(x)$, pour tout $x \in S$.

Les problèmes d'optimisation globale sont très répandus dans la modélisation mathématique de problèmes pratiques et cela pour un large spectre d'applications. On se concentrera ici sur le cas sans contraintes. On se situera néanmoins dans le contexte du problème d'optimisation du type *boîte noire* : la fonction-objectif n'est pas donnée par une forme mathématique *explicite*. De tels problèmes, très fréquents dans l'industrie et les services, sont particulièrement difficiles. Leurs fonctions-objectifs ne peuvent être évaluées que, par exemple, par une coûteuse simulation (informatique ou expérimentale), un processus itératif, ou par une série de programmes informatiques parfois anciens, non-modulaires, que l'on n'est pas autorisé à modifier et/ou requérant l'interaction d'un druide expérimenté, en bref : inutilisables par un logiciel d'optimisation mathématique traditionnelle.

Un exemple d'un tel problème est le problème de mise au point moteur que nous avons rencontré chez Renault. Il s'agit de trouver les réglages du moteur (tel l'avance à l'injection) minimisant sa consommation en carburant tout en respectant les

normes de pollution. L'évaluation d'une proposition de solution requiert alors un essai sur banc moteur.

Généralement, dans nos cursus, on se contente d'enseigner l'optimisation locale ou *convexe* (cas où f et S sont explicites, différentiables et convexes) qu'on traite avec des algorithmes cherchant un point *stationnaire* (satisfaisant les conditions nécessaires de premier ordre de Karush-Kuhn-Tucker : voir le site [13] pour le logiciel associé).

On imagine aisément l'attrait de l'optimisation globale pour aborder, par exemple, un problème économique où l'on cherche à minimiser le coût : on préférera clairement un bon minimum local à un moins bon. Dans certaines applications cependant, l'optimisation globale est tout simplement incontournable. C'est le cas du problème d'équilibre des phases qui se pose lors de l'étude des réservoirs pétroliers. Seul un minimum *global* du problème d'optimisation associé donne une information utile sur ce qui se passe à l'équilibre. Dans [10], on tire parti d'une structure mathématique particulière de ce problème pour obtenir un algorithme convergeant vers un optimum global en un nombre fini d'itérations.

Un autre exemple où l'on **doit** trouver un optimum global est le problème de structure moléculaire [4] : trouver la configuration tridimensionnelle d'une molécule de façon à ajuster au mieux des distances inter-atomiques mesurées expérimentalement. L'obtention d'un optimum local (et il y en a un nombre qui est exponentiel en la taille du problème), n'apporte aucune information utile à la compréhension de la structure de la molécule. Ce problème est crucial pour l'industrie pharmaceutique lors de la conception d'un médicament. Dans [2] on propose d'exploiter la séparabilité partielle de ce problème d'optimisation numérique à l'aide de la théorie des graphes.

Alors que l'optimisation *locale*, née avec le calcul différentiel, a plus de 300 ans, les termes «optimisation globale» n'apparaissent pour la première fois qu'en 1975 dans un titre [3]. Depuis, une revue, le *Journal of Global Optimization*, est née (1991) et des conférences spécialisées ont lieu régulièrement. Cette récente ferveur s'explique par des demandes pressantes des utilisateurs ; par des contributions de

¹Université de Toulouse, Institut de Mathématiques

mathématiciens pour des problèmes spécialement structurés; et surtout par le développement de la puissance de calcul. Ce dernier entraîna une algorithmique foisonnante inventée par des ingénieurs et des spécialistes de la RO pour des applications spécifiques.

Notons que le cas très particulier du problème d'*optimisation quadratique* (fonction-objectif quadratique et domaine réalisable défini par un nombre fini d'(in)équations affines) est déjà un problème NP-difficile si l'on cherche un optimum global. C'est cette difficulté intrinsèque qui a fait que l'optimisation globale est longtemps restée un champ d'étude marginal et ce, même dans le domaine spécialisé de l'optimisation numérique.

2 Survol des méthodes

Comment procède-t-on pour aborder un problème d'optimisation globale? Les méthodes génériques de résolution peuvent être subdivisées en deux classes : les méthodes déterministes et les méthodes stochastiques.

2.1 Méthodes déterministes

Elles regroupent les méthodes qui ne comportent aucun aspect aléatoire ni stochastique, telles la séparation et évaluation progressive (*branch & bound*), les méthodes de plans de coupe, l'analyse par intervalles, l'homotopie (méthode de lissage), les méthodes lipschitziennes et les méthodes de décomposition. Chacune est adaptée à une classe particulière de problèmes d'optimisation globale. Par exemple, l'**analyse par intervalles**, requiert une forme explicite de la fonction-objectif. Cette méthode subdivise récursivement l'espace de recherche jusqu'à ce qu'on arrive à montrer pour chaque sous-domaine qu'il ne peut contenir d'optimum global ou qu'il ne contient que des solutions optimales (à ϵ près). Pour y arriver, l'analyse par intervalles utilise un *branch & bound* couplé par exemple avec des calculs de bornes pour la fonction-objectif et son gradient dans chaque sous-domaine. Cette méthode peut être appliquée à des problèmes de taille modérée (disons en deçà de 50 variables). Des exemples d'application fructueuse de l'analyse par intervalles sont [11] pour les moteurs électriques ou encore [9] en génie chimique.

Systématiques, les méthodes déterministes citées plus haut requièrent généralement des temps de calculs excessifs pour les problèmes de grande taille. Aussi, les ingénieurs confrontés aux problèmes pratiques se contentent souvent de solutions approxi-

matives (réalisables et ayant une «bonne» valeur de f) à l'aide d'**heuristiques**. Les exemples les plus triviaux sont les **algorithmes gloutons** et les méthodes **par amélioration locales**.

Il existe une grande demande pour des méthodes d'optimisation numérique n'exigeant pas que l'utilisateur programme le gradient de la fonction-objectif. Cette demande est satisfaite en général par des variantes «approximations par différences» des méthodes classiques d'optimisation (locale) numérique. Le principe est le même que celui qui consiste à approcher la dérivée d'une fonction unidimensionnelle par $\frac{f(x) - f(x+h)}{h}$ avec h petit. Pour un domaine à n dimensions, cela nécessite non plus 2 évaluations de f mais $n+1$. Les désavantages sont la perte de précision due aux erreurs d'annulation (différences de nombres presque identiques) et aux divisions par de petits nombres, ainsi que le grand nombre d'évaluations de fonction-objectif que cette technique nécessite (particulièrement critique dans le cas d'applications pour lesquelles une évaluation de f est coûteuse). Une autre objection aux approximations par différences est qu'on y remplace chaque gradient par un bouquet serré d'au moins $n+1$ évaluations de fonction-objectif au lieu de disperser ces points où l'on évalue f (d'autant plus si on cherche à optimiser globalement!). De plus, dans le cas où f est bruitée, la contribution du bruit aux taux de changement prédits serait moindre si les points d'évaluation étaient plus espacés.

Ce dernier argument milite en faveur des **méthodes directes** (sans dérivées). Ces méthodes sont en fait des méthodes d'optimisation locale mais sont souvent utilisées comme heuristiques en optimisation globale. Elles remontent au début des années 1960 (**Hooke-Jeeves** [5]). Certaines construisent un modèle interpolant la fonction-objectif, d'autres sont du type «simplexe», comme la méthode du **polytope mouvant** [14] (dite aussi du **simplexe non-linéaire** ou encore de **Nelder-Mead**) que nous décrivons ici. C'est la méthode d'optimisation (locale) sans contraintes la plus utilisée de NAG (bibliothèque commerciale de programmes d'analyse numérique et d'optimisation)!

Un *simplexe* est l'enveloppe convexe de $n+1$ points dans \mathbb{R}^n , formant un volume non-nul. Selon l'article [14] introduisant la méthode du polytope mouvant, le simplexe s'adapterait au paysage local, s'allongeant vers le bas sur les plans inclinés, changeant de direction en rencontrant une vallée avec un angle et se contractant au voisinage d'un minimum. La méthode du polytope mouvant est initialisée par l'évaluation de f en $n+1$ points. À chaque itération, on engendre un nouveau point sur la droite allant

du pire point au centre des autres points et on évalue f en ce nouveau point. Ensuite, selon différents cas de figure, ce nouveau point remplacera le pire point, ou bien on allongera le simplexe dans une direction, ou encore on contractera le simplexe dans la direction du meilleur point, etc. L'algorithme se termine lorsque les arêtes du simplexe deviennent de longueur suffisamment petite.

Cet algorithme est mis en échec notamment par le cas d'une fonction *lisse* (dont la dérivée est continue) strictement convexe en dimension 2, ce qui rend très discutable l'utilisation massive de la méthode du polytope mouvant et, du même coup, sert de motivation aux recherches actives en optimisation sans dérivées depuis 1990 (voir le *survey* [15]). Néanmoins, on trouvera dans le livre [7] des propositions pour l'utilisation pratique de la méthode du polytope mouvant.

2.2 Méthodes stochastiques

Cette deuxième classe regroupe les méthodes comportant une forme de recherche aléatoire, généralement un échantillonnage de la fonction-objectif en des points choisis aléatoirement dans le domaine réalisable. Nous distinguons trois types de méthodes stochastiques : les méthodes à deux phases, les méta-heuristiques et, enfin, les méthodes basées sur les surfaces de réponse.

2.2.1 Méthodes à deux phases

Une *phase globale* évalue f en un nombre de points engendrés aléatoirement et une *phase locale* manipule ces points, par exemple par l'entremise d'améliorations locales. On note que, et cela reste valable pour les autres méthodes stochastiques que nous présenterons, engendrer ces points aléatoirement dans le domaine réalisable n'est pas une tâche facile, même dans le cas simple d'un polyèdre.

La méthode d'**initialisations multiples** (*multistart*) applique une procédure de recherche locale L à chaque point engendré aléatoirement et garde en mémoire le meilleur point trouvé. Idéalement, la procédure L ne serait pas appelée plus d'une fois dans chaque *bassin d'attraction* (le bassin d'attraction d'un minimum local x^* étant l'ensemble des points du domaine réalisable S à partir desquels L convergera vers x^*).

Cette remarque conduit à la **méthode de regroupement en grappe** (*clustering method*) qui cherche à regrouper les points de l'échantillon qui sont près les uns des autres afin de n'initialiser la procédure de recherche locale qu'une seule fois dans

chacun de ces groupes. Le **multi level single linkage** [16] par exemple engendre à chaque itération N points aléatoirement. Puis, elle considère un à un ces points. Si le point considéré est le meilleur dans le voisinage (une boule de rayon évoluant au fur et à mesure des itérations), alors on applique la procédure de recherche locale à partir de ce point. Cet algorithme satisfait certaines propriétés théoriques comme par exemple : la probabilité que L soit appliquée à l'itération k tend vers zéro lorsque k tend vers l'infini, etc.

2.2.2 Méta-heuristiques

Les méta-heuristiques sont considérées avec un certain mépris par une partie de la communauté mathématique (« méthodes vaudous »), en raison du manque de fondements théoriques pour expliquer leur succès. Leur performance dépend fortement de l'ajustement de paramètres correspondant à un bon compromis entre *intensification* (exploitation des meilleures solutions trouvées) et *diversification* (exploration de l'espace de recherche). De plus, leur mise en œuvre devra exploiter au maximum la structure particulière du problème d'optimisation globale considéré. On retrouve en optimisation globale les méthodes classiques que sont la **recherche avec tabous**, le **recuit simulé** et les **algorithmes génétiques** (et autres algorithmes évolutionnaires).

2.3 Méthodes basées sur les surfaces de réponse

Ces méthodes sont apparues avec le **krigeage** en géostatistiques, du nom de l'ingénieur minier Krige qui a développé cette méthode pour prédire la localisation des réserves de minerai de fer. Elle s'appuie sur l'hypothèse suivante : les valeurs de la fonction-objectif en des points rapprochés ont un certain degré de corrélation spatiale alors qu'en des points éloignés les valeurs de f sont statistiquement indépendantes. De manière plus générale, les méthodes basées sur les surfaces de réponse supposent que la fonction-objectif peut être modélisée par la réalisation d'un processus stochastique gaussien. En conséquence on suppose que la valeur de f en un point x non encore évalué peut être interprétée comme une variable aléatoire $Y(x)$ ayant une certaine distribution. Avant tout, on évalue la fonction-objectif en un ensemble de points initiaux. On interpole ensuite ces données avec une combinaison linéaire de fonctions de base (de régression), afin d'obtenir un modèle approchant f . Les fonctions de base comportent des paramètres à ajuster.

Une fonction d'erreur reflète l'incertitude dans le modèle. Une compromis entre la minimisation du modèle (intensification de la recherche) et la maximisation de la fonction d'erreur (diversification de la recherche) constitue une fonction critère permettant de déterminer un nouveau point candidat pour l'évaluation de f . Des variantes de cette méthode sont étudiées dans [6, 8].

Enfin, notons qu'on peut inclure les **réseaux neuronaux** [17] dans la catégorie des méthodes basées sur les surfaces de réponse.

3 Conclusion

Nous avons tenté d'introduire le lecteur au difficile problème d'optimisation globale et de présenter un panorama des méthodes pratiques pour sa résolution. Nous ne nous sommes attardés que sur des méthodes *génériques*, qui s'appliquent à tout problème d'optimisation globale. L'article [12] tente de faire un point sur différentes approches, testées de façon indépendante et sur des problèmes-tests issus de la pratique. Dans leur conclusion, les auteurs restent cependant sceptiques quant à la possibilité de développer des méthodes efficaces pour le problème *général* d'optimisation globale qui n'exploiterait ni la connaissance métier propre au domaine d'application ni la structure particulière d'une classe de problème d'optimisation globale donnée. Les méta-heuristiques permettent, dans une certaine mesure, d'intégrer la connaissance propre au domaine d'applications via l'ajustement (souvent laborieux) des nombreux mécanismes et paramètres en jeu définissant ces méthodes. Pour une classe particulière de problèmes, on pourrait considérer «optimiser» la valeur de ces paramètres sur un échantillon d'instances de cette classe. On s'attendrait alors à ce que la méthode ainsi construite fonctionne bien lorsqu'appliquée à une nouvelle instance de cette même classe de problèmes. Ce travail préliminaire d'ajustement de paramètres est cependant considérable puisqu'il s'agit là, en soi, d'un problème, non-trivial, d'optimisation globale de type boîte noire!

Pour en savoir plus sur l'optimisation globale, le lecteur intéressé se référera à l'ouvrage collectif [1] paru à l'occasion du 25^{ème} anniversaire du GÉRAD.

Références

- [1] C. Audet, P. Hansen, G. Savard (éd). *Essays and Surveys in Global Optimization*. Springer, 2005.
- [2] A. Conn, M. Mongeau. Partial separability, global optimization, and graph theory, 2002. Présentation, *Thematic Year Optimization Visitor*, Fields Institute, Toronto. Audio : <http://www.fields.utoronto.ca/audio/01-02/optvisitors/mongeau/>.
- [3] L. C. W. Dixon, G. P. Szegö (éd). *Towards Global Optimization*. North Holland, 1975.
- [4] B. Hendrickson. The molecule problem : Exploiting structure in global optimization. *SIAM Journal on Optimization*, 5(4) :835–857, 1995.
- [5] R. Hooke, T. A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, 8 :212–229, 1961.
- [6] D. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21 :345–383, 2001.
- [7] C. T. Kelley. *Iterative Methods for Optimization*. SIAM, 1999.
- [8] T. Mayer. *Efficient Global Optimization : Analysis, Generalizations and Extension*. PhD thesis, School of Mathematics, University of Edinburgh, Royaume-Uni, 2003.
- [9] K. I. M. McKinnon, C. G. Millar, M. Mongeau. Global optimization for the chemical and phase equilibrium problem using interval analysis. Dans C. Floudas, P. Pardalos (éd), *State of the Art in Global Optimization*, pages 365–382. Kluwer, 1996.
- [10] K. I. M. McKinnon, M. Mongeau. A generic global optimization algorithm for the chemical and phase equilibrium problem. *Journal of Global Optimization*, 12(4) :325–351, 1998.
- [11] F. Messine. A deterministic global optimization algorithm for design problems. Dans C. Audet, P. Hansen, G. Savard (éd), *Essays and Surveys in Global Optimization*. Springer, 2005.
- [12] M. Mongeau, H. Karsenty, V. Rouzé, J.-B. Hiriart-Urruty. Comparison of public-domain software for black box global optimization. *Optimization Methods & Software*, 13(3) :203–226, 2000.
- [13] J. J. Moré, J. Wright. *Optimization Software Guide*. SIAM, 1993. Voir aussi : <http://www-fp.mcs.anl.gov/OTC/Guide/SoftwareGuide/>.
- [14] J. A. Nelder, R. Mead. A simplex method for function minimization. *The Computer Journal*, 7 :308–313, 1965.
- [15] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7 :287–336, 1998.
- [16] A. H. G. Rinnooy Kan, G. T. Timmer, L. Stougie. Stochastic global optimization methods Part II : Multi level methods. *Mathematical Programming*, 39 :57–78, 1987.
- [17] M. Van Grieken - Garcia. *Optimisation pour l'apprentissage et apprentissage pour l'optimisation*. Thèse de doctorat, Institut de Mathématiques, Université Paul Sabatier, Toulouse, avril 2004. <http://tel.archives-ouvertes.fr/>.