

# Optimisation de fonctions boîtes noires avec et sans contraintes

Emeline Tenaud

LocalSolver, 24 Avenue Hoche, 75008 Paris, France  
etenaud@localsolver.com

**Mots-clés** : *optimisation de boîtes noires, modèle de substitution, fonction de base radiale (RBF), solveur.*

## 1 Introduction et contexte

LocalSolver est un solveur d’optimisation globale, mêlant des techniques de recherche opérationnelle exactes et des techniques heuristiques, comme la recherche locale [2]. De type “*model-and-run*”, il permet de modéliser divers problèmes d’optimisation (combinatoires, continus, mixtes, ...) et de les résoudre sur des instances de grande taille.

Nous étudions ici l’optimisation de fonctions dites boîtes noires, qui sont des fonctions dont on ne dispose pas de leur formulation analytique mais uniquement d’un oracle permettant de les évaluer. En pratique, ces fonctions sont coûteuses en temps de calcul et/ou en mémoire, ainsi le nombre d’évaluations de l’oracle est souvent limité.

## 2 Optimisation de fonctions boîtes noires

### 2.1 Méthode de résolution générale

Sachant que le budget d’évaluation de la fonction est limité, chaque nouveau point à évaluer par l’oracle doit être judicieusement choisi. L’approche utilisée dans LocalSolver pour résoudre ces problèmes est basée sur la méthode de substitution utilisant des fonctions de base radiale (RBF) introduite par Gutmann [3], et sur l’implémentation proposée par Costa et Nannicini [1]. Dans cette méthode, la fonction objectif est approchée par une fonction de base radiale grâce à divers mécanismes d’ajustements et un algorithme de validation croisée permettant de choisir le meilleur modèle à chaque itération. Ensuite, l’algorithme alterne entre des phases d’exploitation, où le modèle choisi est optimisé, et d’exploration, permettant d’explorer l’espace de recherche et éviter ainsi les optima locaux. Les sous-problèmes obtenus sont ensuite résolus grâce au solveur non linéaire de LocalSolver, permettant de définir un nouveau point à évaluer à chaque itération.

### 2.2 Intégration des contraintes

#### 2.2.1. Contraintes explicites

Depuis la version 10.0 de LocalSolver, il est possible d’ajouter des contraintes explicites au modèle de boîte noire. Les contraintes explicites sont des contraintes dont on dispose de la formulation analytique. Ces contraintes peuvent être écrites directement dans le formalisme de modélisation de LocalSolver, celui-ci offrant de nombreux opérateurs dont des opérateurs mathématiques non linéaires. Peu de solveurs de fonctions boîtes noires offrent cette capacité

d'exprimer des contraintes explicites dans leur modèle, alors qu'en pratique ce sont des contraintes que l'on retrouve fréquemment dans des problèmes industriels. Le solveur ayant une meilleure connaissance de l'espace de recherche, il évite ainsi d'évaluer des points ne respectant pas les contraintes analytiques, ce qui permet d'économiser le budget limité mis à notre disposition.

### 2.2.2. Contraintes boîte noire

Dans la prochaine version 11.0 de LocalSolver, les fonctions boîtes noires pourront retourner plusieurs valeurs qu'il sera ensuite possible de contraindre, tout en optimisant un objectif pouvant lui aussi être une valeur de retour de la fonction ou un objectif analytique défini avec les opérateurs du solveur. L'approche retenue pour intégrer ces contraintes est celle décrite par Regis [4]. De même que pour la fonction objectif, les contraintes sont approchées par un surrogate de type RBF. Un critère de marge est défini pour chaque contrainte pour aider à satisfaire la faisabilité. Ce critère est ensuite affiné à chaque itération en fonction de la faisabilité du point précédemment évalué, ce qui permet d'augmenter les chances d'atteindre l'optimum même si le point optimal est situé aux limites de l'espace des solutions.

## 3 Résultats et perspectives

L'implémentation effectuée au sein de LocalSolver s'appuie fortement sur les capacités non-linéaires du solveur et permet d'obtenir de très bon résultats, comparables à ceux obtenus par d'autres solveurs d'optimisation de boîte noire comme RBFOpt [1] et NOMAD<sup>1</sup>. Ces résultats sont illustrés sur le tableau suivant, obtenus sur un benchmark de 153 instances issues de la littérature (dont 24% avec contraintes), lancées sur 5 graines différentes et avec un budget de 150 évaluations.

	Optimalité (%)	Faisabilité (%)	Gap moyen (%)	Evaluations (#moyen)
<b>LocalSolver</b>	49,3	99,5	23,9	86
<b>RBFOpt</b>	48,9	97,0	23,8	88
<b>NOMAD</b>	36,2	95,7	26,9	107

TAB 1 – Résultats de solveurs d'optimisation de boîtes noires sur le benchmark

Afin de couvrir un maximum de problèmes dans ce domaine, les perspectives sont à terme d'offrir la possibilité d'optimiser plusieurs objectifs issus des valeurs de retour de la fonction boîte noire.

## Références

- [1] A. Costa, G. Nannicini *RBFOpt : an open-source library for black-box optimization with costly function evaluations*. Optimization Online 2014-09-4538, 2014.
- [2] F. Gardi, T. Benoist, J. Darlay, B. Estellon, R. Megel. *Mathematical Programming Solver based on Local Search*. FOCUS Series in Computer Engineering, ISTE Wiley, 2014.
- [3] H.M. Gutmann *A radial basis function method for global optimization*. Journal of Global Optimization 19, 201-227, 2001.
- [4] R.G. Regis. *Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points*. Engineering Optimization, 46(2):218–243, 2014.

<sup>1</sup> <https://www.gerad.ca/nomad/>