

# Résolution de problèmes continus non-linéaires avec Hexaly

Nikolas Stott<sup>1</sup>

Hexaly, 251 boulevard Pereire, 75017 Paris, France  
nstott@hexaly.com

**Mots-clés :** *solveur, optimisation non-linéaire, optimisation globale, détection de convexité, points intérieurs*

## 1 Introduction

Hexaly est un solveur d'optimisation mathématique basé sur différentes techniques de recherche opérationnelle, combinant des méthodes exactes, telles que la programmation linéaire, non linéaire et par contraintes, et heuristiques, comme la recherche locale. Son but est d'offrir une approche de type "model-and-run" à des problèmes d'optimisation (combinatoires, continus, mixtes...). Nous présentons ici les stratégies heuristiques et exactes employées par Hexaly pour trouver des solutions à des problèmes continus non-linéaires. Le solveur utilise ces approches de façon parallèle et hybride.

Nous nous intéressons plus précisément à des problèmes d'optimisation de la forme

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad h_j(x) = 0 \\ & l_k \leq x_k \leq u_k \end{aligned}$$

où toutes les variables  $x_k$  sont continues et où les fonctions objectif et contraintes sont disponibles sous forme analytiques et modélisées à l'aide d'un graphe d'expressions dans le solveur.

## 2 Nouveaux composants dans Hexaly 14.5

### 2.1 Détection de structure

La connaissance complète du problème d'optimisation sous la forme d'un graphe d'expressions permet d'effectuer des analyses de structure. Hexaly effectue systématiquement une classification du problème selon la nature de son objectif et de ses contraintes (linéaire, quadratique, non-linéaire) et une analyse de sa convexité. Ces informations sont obtenues en propageant des informations simples (degré, monotonie, convexité) des expressions du modèle des variables vers l'objectif et les contraintes. Des analyses supplémentaires plus coûteuses sont faites lorsque le problème est quadratique mais non-trivialement convexe. Les résultats de ces analyses permettent d'orienter la résolution vers les algorithmes les plus efficaces pour un problème donné.

### 2.2 Un algorithme de point intérieurs modulaire

Hexaly contient différents sous-solveurs permettant de résoudre des problèmes non-linéaires. Lorsque les hessiennes sont de grande taille et ne sont pas calculables explicitement, Hexaly n'utilise que des méthodes d'ordre 0 de type recherche locale ou d'ordre 1 de type lagrangien augmenté avec gradient conjugué. Sinon, le solveur lance en priorité une méthode d'ordre 2 de type point intérieur [7]. Cet algorithme a été modularisé, afin de proposer une méthode prédictor-correcteur [6] et une méthode de région de confiance [3] en complément de la méthode de filtre [5] déjà présente. Un module peut alors être préféré aux autres en fonction du

type de problème qui est résolu (contraintes linéaires ou non-linéaires, convexe ou non-convexe). Hexaly effectue également des résolutions multiples par défaut, et différents modules peuvent être utilisés dans des résolutions différentes.

### 2.3 Approche exacte

Hexaly contient un solveur MINLP exact [2]. Ce sous-solveur reformule le modèle sous une forme plus adaptée à la procédure de relaxation linéaire/convexe, ce qui lui permet de calculer des bornes inférieures. La borne inférieure de la relaxation du noeud racine est ensuite améliorée à l'aide de génération de plans coupants (coupes de reformulation et linéarisation RLT) et d'un mécanisme de *branch-and-reduce*. Ce solveur génère également des solutions primales en générant et en résolvant des sous-problèmes continus, plus simples que le problème d'origine.

## 3 Résultats

Nous avons comparé les performances d'Hexaly 14.5 avec les solveurs open-source SCIP 9.2 et Ipopt 3.14 sur différentes familles de problèmes d'optimisation non-linéaires. Nous nous concentrons sur 2 métriques principales : le nombre de solutions faisables obtenues et le nombre de solutions optimales prouvées par chaque solveur (sauf Ipopt pour la deuxième métrique), avec un temps limite de 60 secondes.

Nous avons choisi 3 ensembles d'instances : des instances d'optimisation de portfolio (QP convexes), des instances de *pooling* (QCQP non-convexes) [1] et des instances non-linéaires réexportées de la librairie CUTEst [4]. SCIP et Ipopt ont été lancés à travers Pyomo. Les résultats sont présentés dans la table 1.

Famille d'instances	Nombre	Hexaly		SCIP		Ipopt
		faisables	optimales	faisables	optimales	faisables
Portfolio	270	270	210	180	177	236
Pooling	64	64	11	28	10	12
CUTEst	1059	943	756	889	531	883

TAB. 1 – Solutions faisables et optimales obtenues en 60s

## Références

- [1] <https://github.com/cog-imperial/pooling-network>.
- [2] S. Boulmier. *Optimisation globale avec LocalSolver*. PhD thesis, 2020. 2020GRALM037.
- [3] Richard H. Byrd, Mary E. Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.
- [4] N. Gould, D. Orban, and P. Toint. Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60, 07 2014.
- [5] O. Hinder and Y. Ye. A one-phase interior point method for nonconvex optimization. 01 2018.
- [6] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.
- [7] N. Stott. Intégration d'un algorithme de points intérieurs pour résoudre des problèmes non-linéaires dans localsolver. 2019.