



HAL
open science

Un logiciel d'optimisation adapté à la mécatronique

François Schott, Noëlie Di Cesare, Sébastien Salmon, Dominique Chamoret

► **To cite this version:**

François Schott, Noëlie Di Cesare, Sébastien Salmon, Dominique Chamoret. Un logiciel d'optimisation adapté à la mécatronique. 12e Colloque national en calcul des structures, CSMA, May 2015, Giens, France. hal-01516455

HAL Id: hal-01516455

<https://hal.science/hal-01516455>

Submitted on 1 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Un logiciel d'optimisation adapté à la mécanique

F. Schott^{1,2}, N. Di Cesare¹, S. Salmon², D. Chamoret¹

¹ IRTES-M3M-EA7274, Université de Technologie de Belfort Montbéliard, Belfort, France

² Optimization Command & Control Systems, Besançon, France

Résumé — Les applications pratiques de l'optimisation dans toutes les disciplines industrielles, que nous qualifierons de mécatroniques, permettent de concevoir plus rapidement des produits toujours plus performants. Dans ce contexte, nous proposons un logiciel d'optimisation à la fois convivial et performant: *my-OCCS Software*. L'objectif de cet outil est d'accompagner au mieux l'ingénieur dans sa démarche d'optimisation en tenant compte des contraintes de l'univers industriel tout en conservant la rigueur mathématique liée aux méthodes d'optimisation.

Mots clés — optimisation, logiciel, mécanique

1 Introduction

La mécanique n'est pas une discipline nouvelle, mais on assiste ces dernières années à une réelle expansion de ses applications. En effet, dans un contexte industriel extrêmement concurrentiel, avec des cahiers des charges de plus en plus exigeants, les systèmes mécaniques permettent d'envisager des solutions originales et jusqu'à alors inexplorées. Dans ce contexte, il est nécessaire à l'ingénieur de hiérarchiser et de pondérer des critères pour optimiser sa conception et satisfaire au mieux son cahier des charges. Cependant, cette démarche d'optimisation, dans le cadre d'une conception de produit, n'est que très peu appliquée à l'heure actuelle en entreprise, notamment dans les TPE.

Deux freins majeurs semblent intervenir dans l'application de cette démarche. La formation des ingénieurs est un premier frein. En effet, l'optimisation est une discipline à part entière qui peut être difficile à appréhender. L'outil numérique est un second frein. En effet, à l'heure du tout numérique, l'absence d'outils logiciels simples, conviviaux et interfacables avec les outils de l'ingénieur dans un environnement de conception mécanique semble être un point limitatif.

A partir de ce constat, le Laboratoire IRTES-M3M-EA7274 et la start-up Optimization Command & Control Systems (*my-OCCS*, spin-off du laboratoire) ont développé un logiciel d'optimisation convivial et performant s'intégrant dans un environnement de conception mécanique: *my-OCCS Software*. L'objectif de cet outil est d'accompagner au mieux l'ingénieur dans sa démarche d'optimisation en tenant compte des contraintes de l'univers industriel tout en gardant la rigueur mathématique liée aux méthodes d'optimisation.

2 Démarche d'optimisation

A l'heure actuelle, les notions de meilleur rapport/qualité prix et d'optimisation paraissent évidentes. D'un point de vue mathématique, un problème d'optimisation consiste à minimiser ou maximiser une fonction objectif dépendant de variables de conception avec ou sans contraintes. Cependant, la démarche d'optimisation n'est pas naturelle dans un contexte industriel. En effet, il faut, avant toute chose, formuler correctement le problème d'optimisation. Cette formulation du problème d'optimisation passe par trois étapes fondamentales qui nécessitent une parfaite connaissance du problème à traiter. Elles sont la base des données utilisateur du logiciel d'optimisation *my-OCCS Software* et sont décrites dans la partie suivante.

2.1 Définition des variables d'optimisation

Il s'agit ici de déterminer les paramètres les plus influents du problème à traiter. Il est important de connaître et de spécifier la nature des variables. En effet, une variable d'optimisation pourra être continue ou discrète (Figure 1).

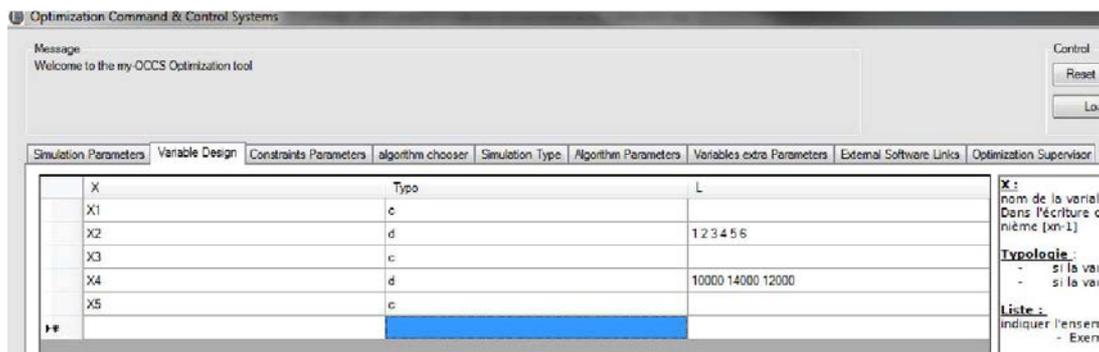


FIG. 1 – Variables d'optimisation

2.2 Définition de la fonction objectif (ou fonction coût)

Il faut ici établir les relations entre les paramètres et identifier le/les critères les plus importants pour le cahier des charges.

Il est fondamental de souligner que cette fonction objectif est, dans la majorité des cas, le résultat d'une simulation numérique, elle est donc étroitement liée à la création d'un modèle. Le logiciel développé propose plusieurs outils d'écriture ou de calcul pour la fonction objectif : appel à une bibliothèque codée en C#, appel à un interpréteur d'équations via le framework NCalc, appel à Excel comme moyen de calcul mais aussi des appels à des outils de calculs usuels tels SciLab, OpenModelica ou encore CodeAster (Figure 2). Cette liste est non exhaustive. En effet, l'aspect modulaire de l'outil proposé offre la possibilité de l'interfacer avec un très grand nombre de logiciels existants, codes éléments finis industriels (Ansys, Abaqus ...), outils de CAO (Catia) suivant les besoins de l'utilisateur. Ce point est un atout majeur du logiciel proposé.

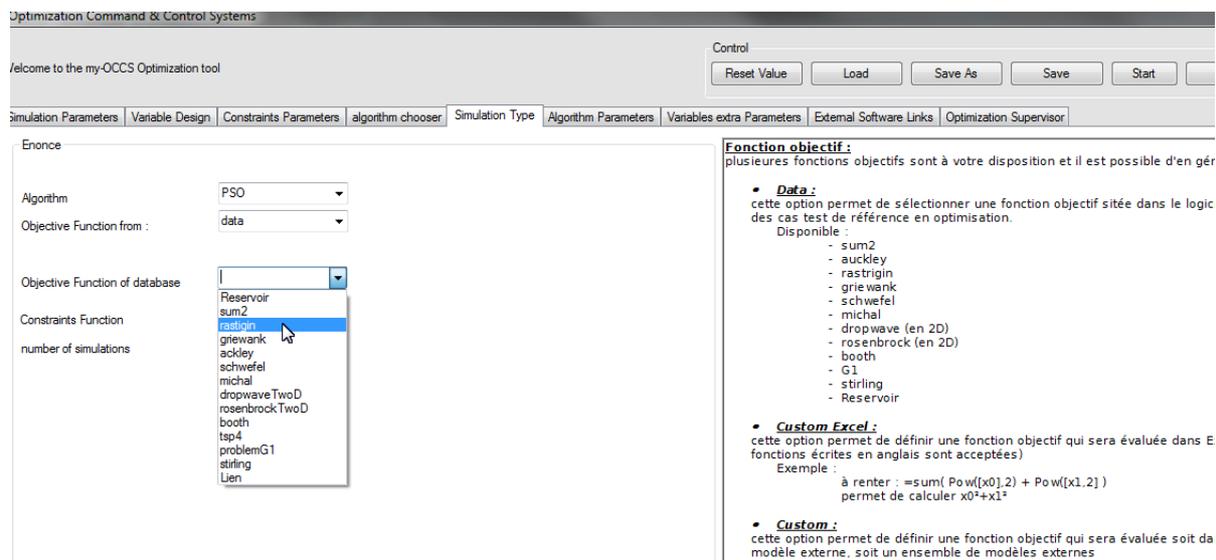


FIG. 2 – Fonction objectif

2.3 Définition des contraintes

Il faut identifier des limitations sur les variables d'optimisation et modéliser d'éventuelles relations entre ces mêmes variables. C'est un point fondamental dans la formulation d'un problème d'optimisation.

tion. D'un point de vue mathématique, les contraintes peuvent être difficiles à traiter et générer de gros problèmes numériques. Dans l'optique de traiter au mieux numériquement ces contraintes sans alourdir la formulation du problème du point de vue de l'utilisateur, *my-OCCS Software* nécessite la définition de trois types de contraintes qui seront pris en compte via une méthode de pénalité (Figure 3).

- Contraintes fortes: sont considérées ici l'ensemble des contraintes qui ne sont pas acceptables physiquement par un ingénieur (ex : une masse négative). Pour une variable donnée x_i , il est nécessaire de définir son intervalle de validité $[C_{inf}, C_{sup}]$. La fonction de pénalité associée est une constante de grande valeur (C_{val}) qui permet d'exclure des points de calculs des plages de résultats admissibles.
- Contraintes faibles: sont considérées ici l'ensemble des contraintes qui génèrent une interaction entre les variables d'optimisation. Ces contraintes sont écrites en utilisant NCalc, ce qui permet de conserver l'équation sous une forme classique. Les pénalités associées sont, ici, écrites sous formes d'équation afin de personnaliser la pénalisation.
- Post-contraintes : sont considérées ici des contraintes qui nécessitent un appel à la fonction objectif pour être évaluées.

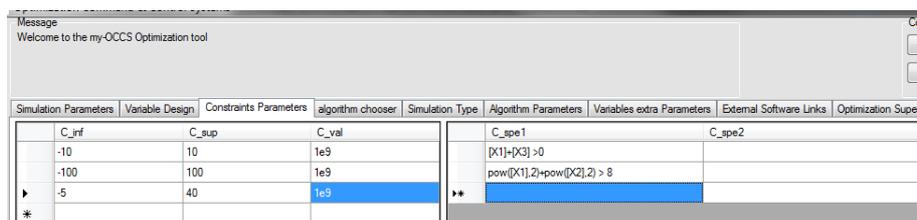


FIG. 3 – Définition des contraintes

3 Résolution du problème

3.1 Les algorithmes implantés

Le logiciel est actuellement en phase bêta et a déjà fait ses preuves sur différents problèmes industriels.

Différents algorithmes classiques et originaux (notamment métaheuristiques) ont été implantés :

- Gradient (simple & conjugué) avec multiples points de départ.
- Simplex (Nelder-Mead) avec multiples simplex de départ.
- CMA-ES, idéal pour une recherche locale [1].
- PSO (inertiel, radius, BSG-Starcraft) [2].

Ils permettent de couvrir la quasi-totalité des besoins en terme d'optimisation dans le domaine de la mécatronique. Nous allons cependant continuer à enrichir cette liste et surtout générer la possibilité de créer des algorithmes hybrides tel que PSO + CMA-ES ou encore Simplex + Gradient local.

3.2 Aide à la décision

Une fois la formulation du problème établie, la difficulté réside dans le choix d'un algorithme d'optimisation pertinent. *my-OCCS Software* est muni d'un outil pour aider l'utilisateur à choisir l'algorithme le mieux adapté à la résolution de son problème (Figure 4). Pour ce faire, l'utilisateur doit répondre à une série de questions relatives à son problème (Nature de la fonction objectif, nature des variables ...). Une analyse de ses réponses via un graphe de décision conduit à une proposition d'algorithme.

3.3 Résultats

Pendant la phase d'exécution du processus d'optimisation, l'utilisateur peut observer l'évolution des variables d'optimisation et de la fonction objectif via un superviseur (Figure 5). Cette surveillance en temps-réel permet de pouvoir arrêter et modifier son optimisation en cas de problème. Il autorise aussi la récupération de résultats intermédiaires, ce qui peut être utile lors de processus d'optimisation longs dans le temps.

Questions

Objectifs

recherche globale ou locale ?

recherche de plusieurs point de fonctionnement ou de l'optimum theorique ?

Première approximation ou recherche fine ?

fonction

Sur le domaine, la fonction est elle derivable ?

Sur le domaine, la fonction possede t elle un ou plusieurs extremes ?

la fonction varie t elle au cours du temps ?

combien de variables y a t il ?

constraints

la fonction est elle contraintes ?

y a t il des contraintes faibles ?

y a t il des variables liées par des contraintes faibles ?

controls

PSO
normal size radius 0.001
!: Radius size choose by default

FIG. 4 – Outils d'aide à la décision

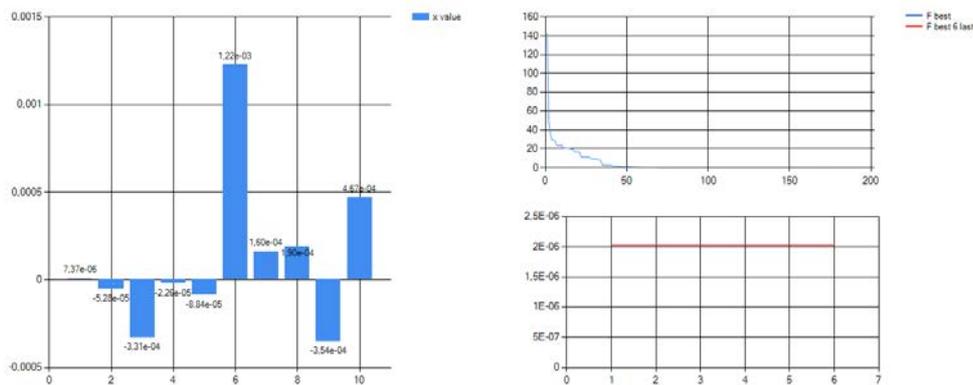


FIG. 5 – Résultats

4 Conclusions

Afin de renforcer l'usage des outils d'optimisation dans le monde industriel, nous avons créé un logiciel à la fois simple, intuitif et performant à destination des ingénieurs. A l'aide d'une conception par module, il est possible de l'interfacer dans un environnement de travail mécatronique. Les algorithmes divers et variés disponibles permettent de couvrir un large champ d'applications. Une version de démonstration du logiciel est disponible sur www.my-occs.fr ou directement sur demande.

Références

- [1] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195, June 2001.
- [2] D. Chamoret, S. Salmon, N. Di Cesare, and Y. J. Xu. Bsg-starcraft radius improvements of particle swarm optimization algorithm: An application to ceramic matrix composites. In Patrick Siarry, Lhassane Idoumghar, and Julien Lepagnot, editors, *Swarm Intelligence Based Optimization*, Lecture Notes in Computer Science, pages 166–174. Springer International Publishing, 2014.