Optimisation multicritère partie 2

Lennard, Consultant Ingénieur Go Concept Lyon,

1 Introduction

Dans le premier article consacré à l'optimisation multicritère, nous avons présenté les bases de l'optimisation monocritère et multicritère. Le présent article s'intéressera à la manière dont les conflits d'objectifs existants peuvent être résolus. Pour ce faire, nous présenterons les deux classes dominantes auxquelles les méthodes d'optimisation multicritère peuvent être attribuées en fonction du moment où les préférences sont introduites. Nous présenterons ensuite brièvement deux méthodes possibles pour chaque classe.

2 Procédures d'optimisation multicritères

Selon COHON et MARKS, les méthodes d'optimisation multicritères peuvent être réparties en trois classes différentes. Dans les méthodes a-priori, les décisions relatives à la pondération des fonctions cibles concurrentes sont déterminées avant que l'optimisation ne soit effectuée. Dans les méthodes a-posteriori, en revanche, le décideur (Decision Maker : DM) ne doit pas fixer de pondération. L'algorithme recherche tous les points efficaces et la décision sur la pondération des différentes fonctions cibles se fait après le déroulement de l'optimisation par la sélection d'un point du front de Pareto par le DM. La troisième possibilité consiste à alterner la recherche par l'algorithme et une réaction par le DM et est appelée prise de décision progressive [1]. Dans la suite de cet article, nous nous limiterons aux méthodes a-priori et a-posteriori, qui seront expliquées plus en détail dans les paragraphes suivants.

Pour faciliter la compréhension des méthodes présentés nous allons regarder le même exemple d’un problème d’optimisation multicritère (POM) comme dans l’article précédente : À l'école, nous avons deux jours pour nous préparer à un examen de mathématiques et à un examen de français. Plus nous étudions une matière, meilleure sera la note que nous obtiendrons. Avec le temps limité dont nous disposons, la note de français sera donc moins bonne si nous consacrons plus de temps à la préparation de l'examen de mathématiques et inversement. Nous sommes confrontés à ce que l'on appelle un conflit d'objectifs.

2.1 Méthodes a-priori

Lors de l'utilisation de méthodes a-priori, le POM est réduit à un problème d'optimisation monocritère dont la fonction cible contient les différentes fonctions cibles du POM [2]. Le DM doit donc exprimer sa préférence avant de procéder à la recherche d'un optimum. Une solution peut ensuite être trouvée à l'aide d'une optimisation monocritère.

Les problèmes liés à l'utilisation de méthodes a-priori résident dans le fait que le DM a souvent du mal à définir ses préférences a-priori. En particulier pour les problèmes complexes dans le domaine de l'ingénierie, les limitations du problème et les effets d'interaction des fonctions cibles ne sont pas toujours connus à l'avance. Or, la qualité des solutions trouvées dépend fortement des préférences du DM, de sorte que leur formulation insuffisante peut entraîner des résultats inefficaces ou mal interprétés [3]. Deux méthodes a-priori sont brièvement présentées ci-dessous.

2.1.1 Méthode des sommes pondérées

La méthode la plus simple pour résoudre un POM est sans doute la méthode des sommes pondérées, dans laquelle chaque fonction cible du POM est pondérée selon la formule (2.1) par un paramètre λ_i qui caractérise le poids de chaque fonction cible [2, 4].

$$\min \sum_{i=1}^{m} \lambda_i \cdot f_i(x) \tag{2.1}$$

En revenant à notre exemple avec les deux examens, nous pourrions par exemple dire que la note en mathématiques est deux fois plus importante pour nous que la note en français et nous chercherions alors la répartition du temps qui donne la solution optimale pour ce problème d'optimisation.

2.1.2 Méthode des contraintes d'Epsilon

La méthode de la contrainte epsilon est l'une des méthodes les plus connues pour résoudre les POM, dont le développement remonte à HAIMES [5]. L'idée est de minimiser une seule des fonctions cibles du POM, tandis que les autres fonctions cibles sont transformées en contraintes de la forme $f_k \leq \epsilon_k$. Les valeurs de ϵ_k définissent alors des valeurs limites pour les fonctions cibles réduites à des conditions secondaires, qui ne doivent pas être dépassées [2].

Si nous reprenons l'exemple précédent, nous pourrions décider que le français est nettement plus important que les mathématiques, mais que nous voulons tout de même réussir les deux examens. Dans ce cas, nous essaierions de répartir notre temps de manière à réussir de justesse les mathématiques et d'obtenir la meilleure note possible en français.

2.2 Méthodes a-posteriori

Contrairement aux méthodes a-priori, dans les méthodes a-posteriori, le DM ne fait part de ses préférences qu'après l'optimisation effectuée. La POM n'est donc pas réduite à un problème d'optimisation monocritère comme c'est le cas avec les méthodes a-priori. Les objectifs de la résolution d'un POM avec des méthodes a-posteriori sont de trouver des points non-dominés dans l'espace de résolution [6].

Avec les méthodes a-posteriori, nous chercherions donc dans notre exemple toutes les répartitions temporelles qui ne sont pas dominées par d'autres stratégies d'apprentissage. Une partie des solutions recherchées comprend donc aussi bien la simple focalisation sur l'un des deux examens que différentes stratégies d'apprentissage possibles qui accordent une part de temps à la préparation des deux examens.

L'avantage par rapport aux méthodes a-priori réside dans le fait qu'il n'est pas nécessaire de disposer de connaissances spécifiques sur le domaine, souvent nécessaires pour donner des préférences a-priori, ni de connaître les limites de la POM. L'inconvénient réside en revanche dans l'important travail de calcul nécessaire pour trouver un ensemble de points Pareto optimaux. Cela vaut en particulier pour les fonctions cibles, dont l'évaluation nécessite une grande puissance de calcul. De plus, il peut être difficile pour le DM de traiter correctement la quantité d'informations si la POM a plus de deux fonctions cibles [3].

Possibles méthodes

Deux méthodes a-posteriori possibles pour résoudre des problèmes d'optimisation multicritère sont l'algorithme *paretosearch* [7] et le *NSGA-II* (Non-dominated Sorting Genetic Algorithm II) [8]. L'algorithme *paretosearch* fait varier systématiquement les valeurs des variables de conception, tandis que le *NSGA-II* fait varier les variables de conception de manière aléatoire par ce que l'on appelle une mutation et une recombinaison. Le *NSGA-II* est l'algorithme de référence le plus utilisé dans la littérature pour d'autres algorithmes multicritères évolutifs, car il représente un bon compromis en termes d'applicabilité universelle, d'effort de mise en œuvre et de temps de calcul nécessaire. Ces caractéristiques en font une méthode appropriée pour les questions techniques [9]. En revanche, l'algorithme *paretosearch* a été jugé avantageux pour l'optimisation d'une structure par [10] et par le projet qui a été traité pendant le travail de master de l'auteur de cet article à l'Institut Fraunhofer LBF de Darmstadt.

3 Résumé

La résolution de problèmes d'optimisation multicritère peut se faire soit avec des préférences définies à l'avance (a-priori), soit après coup (a-posteriori). Les méthodes a-priori présentent l'avantage de nécessiter moins de puissance de calcul pour la recherche d'un point unique. L'inconvénient réside dans le fait que les interdépendances entre les objectifs ne sont souvent pas connues à l'avance et qu'il est par conséquent difficile de définir les préférences a-priori. Les méthodes a-posteriori recherchent en revanche l'ensemble du front de Pareto (l'ensemble des points non-dominés). Cela nécessite plus de puissance de calcul, mais suppose moins de connaissances sur la nature du problème d'optimisation et donne au décideur un aperçu de plusieurs solutions possibles. Il n'est pas possible de répondre de manière générale à la question de savoir quelle est la meilleure méthode d'optimisation, car elle dépend de nombreux facteurs. Il faut donc toujours tester différentes méthodes et algorithmes pour trouver la meilleure solution à un problème d'optimisation multicritère.

4 Literaturverzeichnis

[1] **Cohon, J. L. und Marks, D. H. (1975):** A review and evaluation of multiobjective programing techniques. In: *Water resources research*. 11, 208–220.

[2] **Ehrgott, M. (2005):** Multicriteria optimization. Springer, Berlin, New York.

[3] **Thiele, L., Miettinen, K., Korhonen, P. J. und Molina, J. (2009):** A preference-based evolutionary algorithm for multi-objective optimization. In: *Evolutionary computation*. 17, 411–436.

[4] **Das, I. und Dennis Jr, J. E. (1996):** A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems.

[5] **Haimes, Y. (1971):** On a bicriterion formulation of the problems of integrated system identification and system optimization. In: *IEEE transactions on systems, man, and cybernetics*, 296–297.

[6] **Chiandussi, G., Codegone, M., Ferrero, S. und Varesio, F. E. (2012):** Comparison of multi-objective optimization methodologies for engineering applications. In: *Computers & Mathematics with Applications*. 63, 912–942.

[7] MathWorks**. (2023):** Paretosearch Algorithm. https://de.mathworks.com/help/gads/paretosearch-algorithm.html, besucht am 10.05.2023.

[8] **Srinivas, N. und Deb, K. (1994):** Muiltiobjective optimization using nondominated sorting in genetic algorithms. In: *Evolutionary computation*. 2, 221–248.

[9] **Krettek, J. (2012):** Ein multikriterieller evolutionärer Algorithmus mit interaktiver Präferenzintegration – Angewendet zur Optimierung von Hydraulikventilreglern. Dissertation, Technische Universität Dortmund.

[10] **Ihmels, F. (2022):** Multikriterielle Optimierung von Strukturdynamik und Lebensdauer für Leichtbaustrukturen mit diskreten schwingungstechnischen Maßnahmen. Masterthesis, Technische Universität Darmstadt.