

Optimisation Multi-objectifs

Présentation et algorithmes non-Pareto

Pr Pierre Collet

Laboratoire ICUBE
Campus Numérique des Systèmes Complexes

Université de Strasbourg

[Pierre.Collet](mailto:Pierre.Collet@unistra.fr)Pierre.Collet@unistra.fr

Contexte

- Il s'agit de minimiser plusieurs critères en même temps, qui sont souvent malheureusement antagonistes...
- Concerne la plupart des secteurs de l'industrie, où toutes les solutions sont toujours des compromis, entre 2 objectifs (poids/résistance) ou plus (poids/résistance/prix).
- La notion de solution « optimale » perd son sens... : un boîtier Leica à 5500 euros, c'est très cher, mais aucune vibration, un viseur fantastique, un télémètre précis. Inversement, un appareil jetable a un viseur microscopique, n'a qu'une seule vitesse d'obturation, pas de mise au point, ... mais pour quelques euros, ça fait tout de même des photos.

Les 2 sont optimaux, chacun dans leur genre..

- Comme il y a plusieurs solutions optimales possibles et différentes, en général, les décideurs les veulent toutes, pour choisir ensuite, suivant d'autres critères (clientèle).

Méthodes possibles

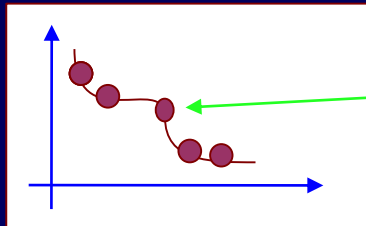
- Traitement séparé des critères.
- Transformation d'un MOP en SOP par agrégation des critères.
- Approches non agrégées, non Pareto.
- Approches Pareto, utilisant la dominance.

Traitement séparé des critères

- C'est possible si les critères sont indépendants entre eux. On commence alors par minimiser un premier critère, puis un autre, ... jusqu'à tous les minimiser.
- Méthode lexicographique (Fourman[85]) : les objectifs sont rangés par ordre d'importance décroissant, puis on commence par minimiser la fonction d'objectif la plus importante, puis la 2è, puis la 3è...
- Premier problème : la minimisation du premier critère emmène l'algorithme vers une zone restreinte, d'où les autres critères ne peuvent plus sortir.
- De plus, dans la vie courante, les critères sont très souvent liés, et surtout, antagonistes...

Agrégation des critères

- Moyenne pondérée :
 - On cherche à minimiser une fonction objectif :
$$f(x) = \sum \alpha_i f_i(x)$$
avec α_i pondération de chaque contrainte.
 - Question : comment pondérer chaque critère ?
 - Comment exprimer l'interaction entre plusieurs critères ?
- Généralement, on ne trouve qu'une seule solution (car le problème est transformé en mono-objectif)
- Ce qu'on optimise est différent de ce qui est recherché.
- Problème : les solutions trouvées seront situées dans des portions convexes de la frontière de Pareto...



Non-trouvable par
agrégation des critères...

Corrélation entre les critères

Autre problème : il y a plusieurs types d'interaction entre les critères... :

- Corrélation positive ou négative (j plus fort que i lorsque i absent, car les 2 sont corrélés).
- Changement d'échelle inopiné :
 - Coût des 200 premiers stylos, 50 euros
 - De 200 à 500 stylos : 10 la centaine supplémentaire.
 - Entre 500 et 1000 : 5 euros la centaine supplémentaire.Comment mélanger un tel critère à un autre qui varie linéairement ?
- Comment gérer des seuils minimaux pour certains critères :
 - $(5+15+15)/3 = 11,66...$ C'est > 10 , mais première note $< 10...$

\square -contrainte

- Minimisation d'un objectif f_i , lorsque tous les autres objectifs $j \neq i$ sont inférieurs à une valeur \square_j .
- Ensuite, on recommence sur un autre critère jusqu'à trouver une solution satisfaisante.
- Pb : il faut déjà trouver une solution où toutes les autres contraintes j sont $< \square_j$.

Méthode lexicographique Fourman[85]

- Les objectifs sont rangés par ordre d'importance décroissant, puis on commence par minimiser la fonction d'objectif la plus importante, puis la 2è, puis la 3è...

Algorithmes utilisables

- Algorithmes d'optimisation standard, faisant évoluer un seul point sur l'espace de recherche.
- Algorithmes à population.

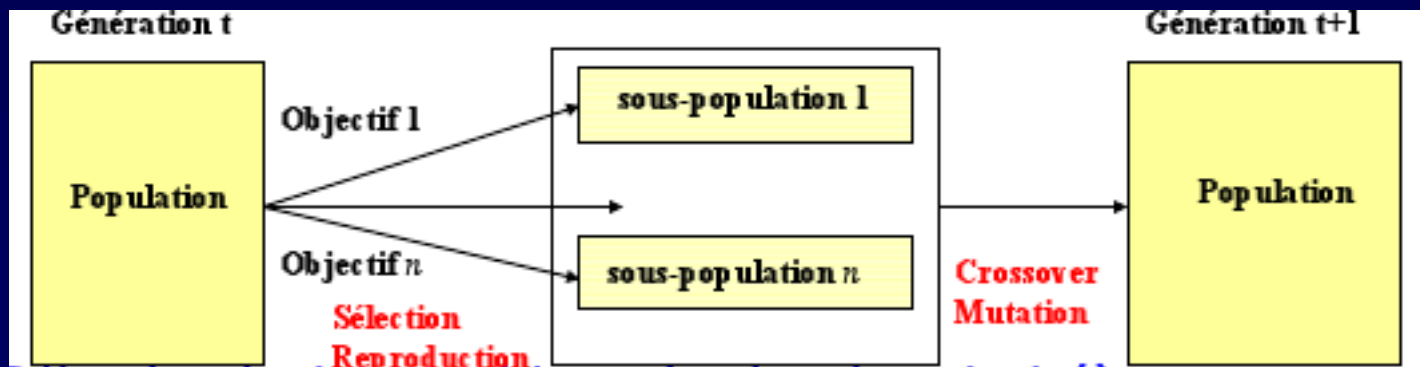
Les algorithmes à population sont bien mieux adaptés à la recherche d'un front de Pareto, d'où le succès des approches évolutionnaires

Approches évolutionnaires

- Dès 1967, Rosenberg a suggéré d'utiliser une méthode génétique pour trouver les principes chimiques d'une population d'organismes unicellulaires selon des propriétés (objectifs) multiples, mais sans implémentation.
- Première implémentation : David Schaffer en 1984, qui modifie un EA standard pour effectuer des sélections indépendantes suivant chaque critère, jusqu'à remplir une population de parents. Ensuite, la population de parents est mélangée (pour mélanger les critères) et 2 parents créent 2 enfants par croisement + mutation (Vector Evaluated Genetic Algorithm: VEGA). Ça fonctionne bien pendant quelques générations, mais après, un individu trouve un bon coin, et la population converge.
- Ensuite, pas grand chose ne se passe jusqu'à ce qu'en 1989, Goldberg suggère d'utiliser une procédure de tri portant sur la non-domination des individus, tout en préservant la diversité. En 2000, Deb propose NSGA-II.

VEGA (non agrégé, non Pareto)

- VEGA (Vector Evaluated Genetic Algorithm)
Schaefer[85]
- Soient i objectifs et une population de P individus. On sélectionne P/i individus pour chaque critère différent pour créer i sous-populations, qui contiennent chacune les meilleurs individus pour chacun des critères.
- Ensuite, on mélange les i populations en gardant l'ordre interclassé des individus et on utilise cette population reclassée pour créer des enfants par croisement/mutation.



VEGA : problèmes

- Pb : à chaque création de population intermédiaire, les individus très bons dans un seul domaine sont privilégiés, au détriment d'individus bons partout, qui ne sont les premiers dans aucune sous-population, alors que justement, c'est eux qu'on recherche !!!
- On aboutit à la création d' « espèces » spécialisées pour chaque critère.

1 genre par critère (Allenson[92], Lis & Eiben[96])

- Allenson [92] : on maintient un nombre identique de « mâles » et « femelles » dans la population, chaque genre étant jugé par l'objectif qui lui est associé.
- A chaque génération, les enfants créés sont aléatoirement d'un genre ou de l'autre.
- Les enfants remplacent les plus mauvais individus du même genre qu'eux.
- Lis&Eiben[96] : extension à n genres, toujours un par critère. Lors de la création d'un individu, on sélectionne n parents, et on fait un croisement multipoints. Celui qui a donné le plus de gènes transmet son genre à l'enfant (aléatoire si nombre de gènes égal).
- Dans ces algorithmes, la diversité par genre est bien maintenue, et la création d'espèces différentes est minimisée car un individu ne transmet pas automatiquement son genre.