

Optimisation et recherche opérationnelle

Olivier Grunder

Département Génie Informatique

Laboratoire Systèmes et Transport

Université de Technologie de Belfort-Montbéliard

`olivier.grunder@utbm.fr`

Résumé

Présenter les principales méthodes et algorithmes de recherche opérationnelle permettant d'aborder la plupart des problèmes d'optimisation connus.

Plan du cours

- Introduction à la recherche opérationnelle
- Programmation linéaire
- Programmation en nombres entiers : méthode des coupes
- Application des graphes à la RO : programmation dynamique, recherche arborescente
- Méthodes heuristiques : algorithmes de descente et du kangourou, recuit simulé, méthode tabou
- Optimisation distribuée : colonie de fourmis, intelligence en essaim
- Optimisation de processus stochastiques
- Théorie des jeux
- Algorithme génétique

Introduction à la recherche opérationnelle (RO)

Technique récente datant de la seconde guerre mondiale appliquée dans le cadre des “opérations” militaires de logistique.

Problèmes de la RO soulevés bien avant :

- XVIIème : décision dans l’incertain (Pascal, Fermat, Bernoulli, Waldegrave),
- XVIIIème : pb économique combinatoire des déblais/remblais (Monge),
- 1917 : théorie des files d’attente (Erlang),
- 1921-25 : théorie mathématique des jeux (Borel),
- 1936 : systématisation des graphes (König),
- veille de la guerre 39-45 : programmation linéaire (Kantorovich).

Introduction à la RO

Première équipe de RO dirigée par le physicien anglais Blackett en 1940. Equipe *hétérogène* qui disposait de toutes les *informations* nécessaires, de *données* fiables et qui réservait la *décision* finale à l'amirauté britannique.

Exemples de problèmes traités :

- Implantation optimale des radars de surveillance britanniques,
- protection des convois de navires marchands entre la grande-bretagne et les USA.

Après la guerre, les méthodes de RO ont ensuite été appliquées à l'économie industrielle.

De nombreuses travaux scientifiques ont alors été menés pour développer l'ensemble des méthodes et techniques de la RO.

Introduction à la RO

Raisons de l'apparition tardive de la RO :

- Economistes initialement opposés aux modèles mathématiques simplistes et abstraits non adaptés à la vivacité et la complexité de l'économie.
- Complexité des problèmes est récente (nombre et taille des firmes et relations entre elles complexes).
- Théorie de la RO n'est rien sans des moyens de calcul adaptés (premiers ordinateurs commercialisés en 1955-1956).

Tentative de définition de la RO

La RO peut être vue comme un arsenal de méthodes mathématico-informatiques destiné à l'optimisation des processus de production et de diffusion des produits.

Quand des hommes, des machines et des produits se trouvent en relations actives, on parle de *phénomène d'organisation*. Ces problèmes sont en général fortement combinatoire au départ. Si on y rajoute du hasard et de la concurrence, c'est encore plus complexe.

Robert FAURE (1918-1982), pionnier de la RO en France, donne cette définition ("Précis de recherche opérationnelle", ed. Dunod, 2000) :

La recherche opérationnelle est *l'ensemble des méthodes et techniques rationnelles d'analyse et de synthèse des phénomènes d'organisation utilisables pour élaborer de meilleures décisions.*

Domaines d'application de la RO

La RO s'occupe des problèmes complexes pour lesquels le bon sens ne peut plus être utilisé.

- Problèmes combinatoires : investissements, niveaux d'activité, affectation, transport, ordonnancement ;
- Domaines de l'aléatoire : files d'attente, fiabilité, sûreté de fonctionnement, gestion de production ;
- Situation de concurrence : politiques d'approvisionnement, de vente, etc.

Ces différentes catégories de problèmes peuvent se retrouver dans l'informatique, les télécommunications, l'économie, la production, la logistique, etc...

Forme générale d'un problème de RO

Problème (P) :

- Optimisation (max ou min) d'un critère $f(x_1, x_2, \dots, x_n)$ (ft de coût, objectif),
- Tout en respectant un ensemble de contraintes pouvant être formulées sous forme d'équations algébriques : $g_i(x_1, x_2, \dots, x_n) \leq 0$ pour $i = 1..m$

On appelle :

- *solution* de (P) : tout vecteur x qui vérifie les contraintes du problème.
- *optimum global* de (P) : toute solution x de (P) qui optimise (max ou min) le critère.
- *optimum local* de (P) : toute solution x^0 de (P) telle qu'il existe un voisinage $V(x^0)$ dans lequel x^0 optimise le critère (max ou min).

Différentes classes de pbs de RO.

Catégorie 1 : Problèmes d'optimisation combinatoires (POC)

Exemple : combien faut-il de temps à une famille de 8 personnes , prenant en commun 2 repas par jour, pour épuiser les diverses configurations possibles de se grouper autour de la table familiale ?

A raison de 2 repas/jr (1x2), 12 mois/an (3x4), 30 jrs/mois (5x6), il faut 56 ans (7x8) pour y arriver : $8! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$ (40320).

Le principe fondamental pour les POC est d'éviter toute énumération.

L'énumération de $20!$ solutions (affectation de 20 personnes à 20 postes) à raison d'un million d'affectation par seconde, prendrait 77096 années. Les compagnies aériennes sont confrontées à de tels problèmes mais à la dimension 100.

Catégorie 2 : Problèmes stochastiques

Exemple : dans une entreprise, une personne entre en moyenne toutes les 4 mn dans un bureau B. L'employé de ce bureau peut servir en moyenne 1 personne toutes les 3 mn 18 s. Peut-on conclure que l'employé suffit à ce service car il reçoit $8 \times 60 / 4 = 120$ clients/jr et leur consacre $3 \text{mn} 18 \times 120 = 6 \text{h} 36 \text{mn}$.

Non pour l'entreprise, car le temps perdu par le personnel en attente peut atteindre des heures ce qui peut revenir plus cher à l'entreprise que l'embauche d'un deuxième employé.

Il est nécessaire pour faire apparaître cette solution de considérer le phénomène aléatoire dans sa globalité et minimiser l'espérance mathématique du coût global des opérations.

Catégorie 3 : Problèmes concurrentiels

Les situations de duel sont plus délicates. Le choix d'une stratégie dépend des concurrents : problème à la fois combinatoire et stochastique.

Soit, par exemple, un fabricant A qui doit décider d'une stratégie de ses prix de vente sachant que la concurrence dispose des mêmes stratégies. Supposons que les gains qu'il réaliserait pour chaque paire de choix sont :

	B+	B=	B-
A+	10	-1	-4
A=	6	0	-2
A-	10	-1	2

La solution est pour A de maintenir ses prix 3 fois sur 5 et de les baisser 2 fois sur 5 pour ne perdre en moyenne que 0,4 à chaque coup.

Limites de la RO : approche mono-critère

Un entrepreneur qui minimise ses investissements, optimise sa fabrication (productivité maximale et coût minimal), détermine le prix de vente le plus intéressant, pourrait imaginer réaliser le profit maximal.

Malheureusement les techniques de RO visent à optimiser une seule et unique fonction. C'est pourquoi tout doit être exprimé sous forme de contraintes sauf la seule et unique fonction à optimiser.

Les contraintes limitent un espace de solutions à n dimensions (si n variables).

Limites de la RO : optimisation partielle

Modèle complet d'une entreprise trop complexe.

Pbs de RO souvent limités à une partie de l'entreprise (par ex. gestion des stocks) ce qui donne une sous-optimisation de l'entreprise qui peut perturber les autres parties (non optimisées).

Par exemple : soient A, B, C, 3 entreprises.

- A fournit B
- B fournit C
- Si B optimise ses stocks, on peut très bien assister à des perturbations importantes chez A ou chez C.

Limites de la RO : critères relatifs et temporels

Un critère d'optimisation à une période donnée peut ne plus être pertinent à une autre période, voire être complètement périmé. Les raisons sont nombreuses :

- Le monde économique bouge
- la technique progresse
- les produits deviennent obsolètes
- la législation évolue
- les modes changent
- etc.

Objectifs et critères

Ce n'est pas à l'analyste en RO mais à l'entrepreneur de fixer les contraintes (valeurs limites) et le critère de choix.

Le chercheur opérationnel n'est pas là pour remplacer le décideur dont la fonction est de choisir quelle solution lui semble la plus adaptée au contexte de son entreprise.

La RO n'est pas là pour prendre des décisions, son rôle est de fournir une connaissance approfondie du problème de façon à aider à la prise de décision.

La RO : une discipline carrefour

La RO est plus une pratique qu'une science. Elle cherche avant tout à agir sur l'évolution des phénomènes plutôt qu'à les comprendre. Un modèle n'est intéressant que s'il permet d'obtenir de meilleurs résultats.

La RO est une discipline carrefour entre :

- l'économie (économie d'entreprise, analyse économique) : modèle initial,
- les mathématiques (théorie des systèmes, méthodes d'optimisation ou statistiques) : choix d'une voie pour atteindre une solution,
- l'informatique (algorithmique, structures de données, BD) : mise en oeuvre pratique.

Rentabilité de la RO

La RO n'est pas gratuite. Par conséquent, la question de la rentabilité se pose tout naturellement.

Deux exemples empruntés à A. Kaufmann :

1. Transport aérien : 400 vols, 600 liaisons entre 13 villes. Objectif : améliorer le plan des vols. Nombreuses contraintes : temps max de pilotage, temps supplémentaire passé au sol, repos obligatoire, retour périodique des appareils et équipages, maintenance, indemnités de déplacement, etc. Gain de 18%.
2. Usine sidérurgique : 3 chaînes de laminoirs avec une très moderne et une vétuste. Un simple programme linéaire à fait gagner 6% du coût total de production.

Notions de complexité : complexité des algorithmes

Un même problème peut être résolu par plusieurs algorithmes. Important de comparer les performances de différents algorithmes.

Les 2 critères sont : temps de calcul et espace mémoire, même si en pratique le temps de calcul est le plus important car on est souvent limité dans le temps.

Par abus de langage, on appellera complexité d'un algorithme, sa complexité en temps. C'est la relation de proportionnalité existant entre le nombre d'opérations élémentaires à effectuer pour trouver une solution et n la taille des données de départ.

Complexité des algorithmes : notation O

Soit f une fonction croissante sur l'ensemble des entiers positifs.

Définition : $O(f)$ est la classe des fonctions g définies sur l'ensemble des entiers positifs, telles que : $\exists c > 0, \exists n_0 > 0$ tels que $\forall n \geq n_0, g(n) \leq c.f(n)$.

On note $O(1)$ l'ensemble des fts majorées par une constante. Exemple : $g(n) = 1/n \in O(1)$

A une constante multiplicative près, de nombreuses classes sont équivalentes : $O(10000n^2) = O(0,001n^2) = O(n^2)$ et $O(e^n) = O(2^n)$.

Pour tout polynôme $P(n) = a_p n^p + \dots + a_1 n + a_0$, on a $P \in O(n^p)$.

Complexité des algorithmes : notation O

Les relations d'inclusion pour les classes de fts les plus usuelles sont :
 $O(1) \subset O(\log \log n) \subset O(\log n) \subset O(\sqrt{n}) \subset O(n) \subset O(n \log n) \subset O(n^2) \subset O(n^3) \subset \dots \subset O(n^{10}) \subset \dots \subset O(2^n) \subset O(n!)$

Temps d'exécution d'un algorithme en $O(f)$ pour une donnée de taille n ($f(n)$ opérations) sur un ordinateur exécutant 10^9 ops/secondes.

	$n = 100$	$n = 1000$	$n = 10^6$
$f(n) = n$	$< 1s$	$< 1s$	$< 1s$
$f(n) = n^3$	$< 1s$	$1s$	32 ans
$f(n) = n^5$	$10s$	11 jours	$3 \cdot 10^{13} \text{ ans}$
$f(n) = 2^n$	$3 \cdot 10^{14} \text{ ans}$	10^{281} siècles	$10^{3 \cdot 10^5} \text{ siècles}$

Age de la terre : $4,5 \cdot 10^9$ années

Complexité des algorithmes : temps de calcul

Temps nécessaire pour la multiplication de matrices : $C = A \times B$.

1. pour $i = 1$ à n
2. pour $j = 1$ à n
3. $C(i, j) = 0$
4. pour $k = 1$ à n
5. $C(i, j) = C(i, j) + A(i, k) \times B(k, j)$

Certains algorithmes ne donnent pas la même complexité suivant les données de départ. On parle alors de :

- complexité dans le pire des cas : plus grande complexité pour toutes les données de taille n (plus importante),
- complexité dans le meilleur des cas : plus petite complexité,
- complexité en moyenne : difficile.

Problèmes de décision et complexité

Problème de décision = données + question “oui” / “non”.

Exemple :

- parité d'un nombre
- plus court (resp. long) chemin sous la forme “existe-t-il un chemin d'origine a et d'extrémité b de longueur inférieure (resp. supérieure) à D ?”
- pb hamiltonien : “le graphe G admet-il un cycle hamiltonien ?” (passant une fois par chaque sommet)

Complexité d'un problème = minimum des complexités de tous les algorithmes pouvant le résoudre.

Complexité des problèmes de décision : P et NP

Classe P (pbs de décision Polynomiaux) : un pb appartient à la classe P , s'il peut être résolu par un algorithme A de complexité $O(n^k)$ où k est une constante et n la taille des données. On dit que A est *polynomial*.

Exemple : parité ($O(1)$), plus court chemin dans un graphe ($O(n^2)$)

Classe NP (pbs de décision pouvant être résolus en temps polynomial par un *algorithme non déterministe*) : un pb appartient à la classe NP sissi pour toute instance du pb ayant pour réponse "oui", il existe un algorithme polynomial permettant de vérifier que la réponse est effectivement "oui".

Exemples : plus court (resp. long) chemin, cycle hamiltonien (vérification facile qu'un chemin a une longueur inférieure (resp. supérieure) à D , ou qu'un cycle est hamiltonien).

Note : $P \subset NP$

Problèmes de décision NP -complets

Certains pbs de décision de NP sont plus difficiles que les autres (Cook, 70). Ces problèmes forment la classe des problèmes NP -complets de complexité au moins en $O(2^n)$.

Difficile de résoudre optimalement des pbs NP -complets de grande taille.

Exemple :

- Pb de *satisfiabilité* :
 - n variables booléennes
 - m clauses : sous-ensemble de $x_1, \dots, x_n, \overline{x_1}, \dots, \overline{x_n}$
 - Question : existe-t-il une affectation des x_i de sorte que pour chacune des clauses, au moins l'un des littéraux soit *vrai* ?
 - Ex : $\{x_1, \overline{x_2}, \overline{x_3}\}$ et $\{\overline{x_1}, x_3\} \Rightarrow x_1 = x_3 = \text{faux}$ et $x_2 = \text{vrai}$.
- Plus long chemin élémentaire,
- Cycle hamiltonien.

Problèmes d'optimisation NP -difficiles

Problème d'optimisation = trouver une meilleure solution (suivant un critère d'optimisation) parmi un ensemble de solutions réalisables.

On peut associer un pb de décision à tout pb d'optimisation :

- Pb d'optimisation : trouver $s^* \in S / f(s^*) = \min\{f(s), s \in S\}$
- Pb de décision : existe-t-il $s^* \in S / f(s^*) < a$

Quand le problème de décision est NP -complet, le problème d'optimisation est dit NP -difficile.

La résolution exacte du problème d'optimisation ne peut se faire que par :

- des procédures énumératives de séparation et évaluation (PSE : recherche arborescente),
- la programmation dynamique pour certaines sous-classes de problèmes.

Problèmes d'optimisation NP -difficiles

Exemples :

- Problème du Voyageur de Commerce (PVC, TSP)
- Tournées de véhicules (VRP, PDP)
- Problèmes d'ordonnancement avec ressources,
- Emploi du temps,
- Programmation linéaire en nombres entiers.

Résolution exacte non envisageable pour problèmes de grande taille =>
on utilise des résolutions approchées :

- algorithmes polynomiaux avec garantie relative de performance (mesure de l'écart des solutions par rapport à une solution optimale),
- métaheuristiques : fournissent rapidement de bonnes solutions mais sans garantie de performances (recuit simulé, méthode tabou, algorithmes génétiques, etc.)