Aide à la décision - R5.A.11

Méthodes d'optimisation pour l'aide à la décision

Introduction générale

Michel Salomon

IUT Nord Franche-Comté
Département d'informatique

Variété de problèmes d'optimisation

De nombreux secteurs y sont confrontés - Quelques exemples

- Planification de traitement en dosimétrie
- Traitement d'images (recalage d'images ; etc.)
- Problèmes de tournées de véhicules
- Électronique (placement et routage des composants ; etc.)
- Conception de réseaux mobiles (placement des antennes)
- Techniques d'apprentissage en **Intelligence Artificielle**

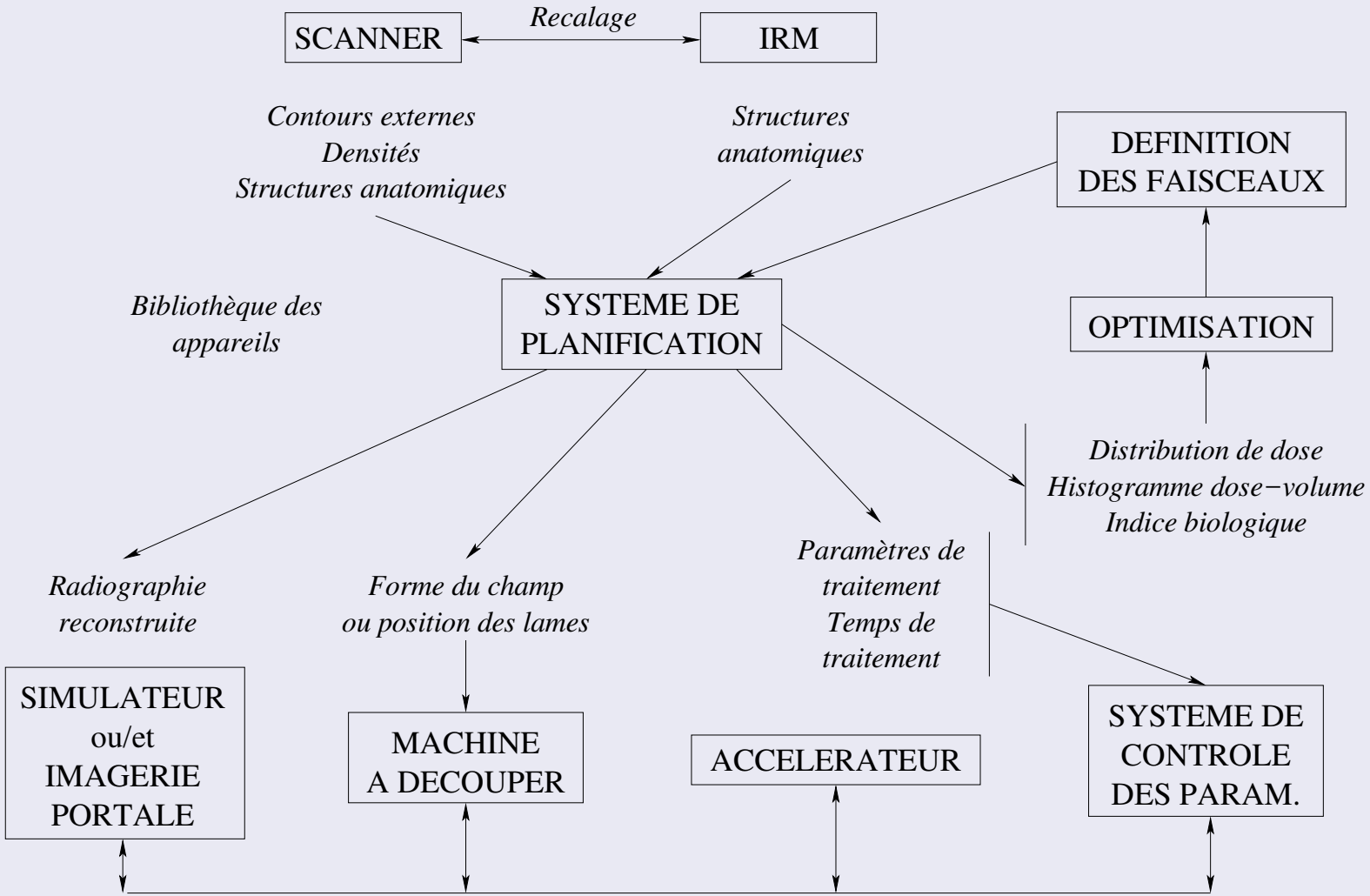
À quoi est liée la difficulté d'un problème d'optimisation ?

- Au nombre de variables $\rightarrow$ quels paramètres faire varier ?
- À l'espace de recherche $\rightarrow$ limites de variation des paramètres ?
- À la (aux) fonction(s) objectif(s) $\rightarrow$ objectif(s) à atteindre et comment le(s) formuler mathématiquement ?

Développement de nombreuses méthodes de résolution

Exemple en dosimétrie

Processus d'optimisation en planification de traitement



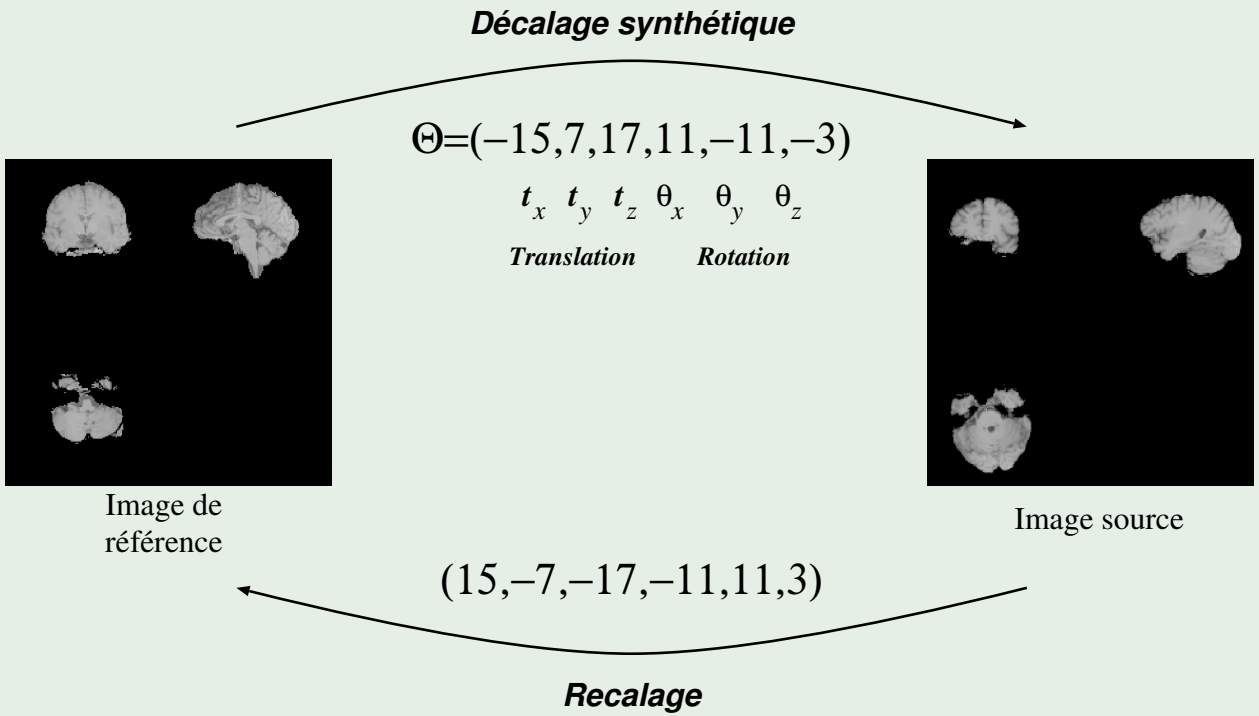
Exemple en imagerie médicale

Recalage d'images monomodales 3D

- D'une dizaine à plusieurs millions de variables à optimiser
  - Recalage rigide $\rightarrow$ intra-patient
    - Rotation (3 paramètres)
    - Translation (3 paramètres)
  - Recalage déformable $\rightarrow$ inter-patient
- "Paysage" complexe des fonctions objectif à minimiser
- Volume de données important
  - IRM 128^3, 256^3, voire plus
  - Dépend de la taille de l'unité élémentaire
    - Standard : 1 mm^3 $\rightarrow$ de 10 à 20 millions ;
    - Haute résolution : $0,5$ mm^3 $\rightarrow$ de 80 à 150 millions
- Utilisation possible en routine clinique
  - Temps de calcul acceptable

Exemple en imagerie médicale

Illustration 1 - Recalage rigide d'IRM 128^3 voxels



Temps de calcul sur processeur MIPS R12000 (500 MHz) - 2001

- 1 processeur = 700,0 secondes
- 16 processeurs = 52,16 secondes (soit un *speedup* de 13,31)

Exemple en imagerie médicale

Illustration 2 - Recalage déformable d'IRM 256^3 voxels

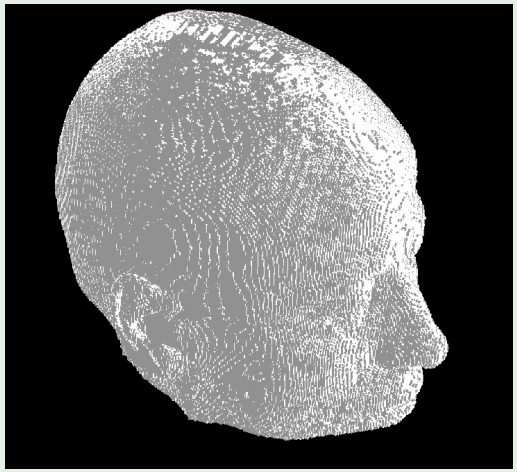
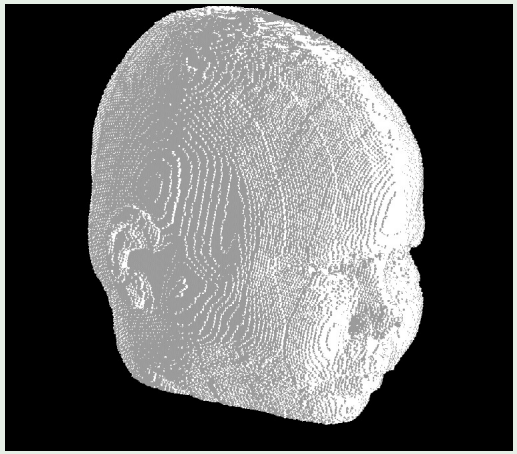


Image source



Résultat

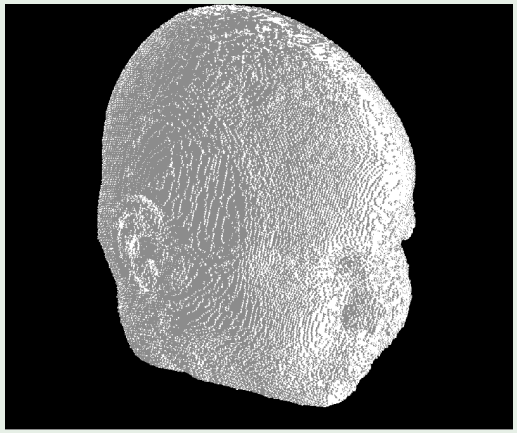


Image de référence

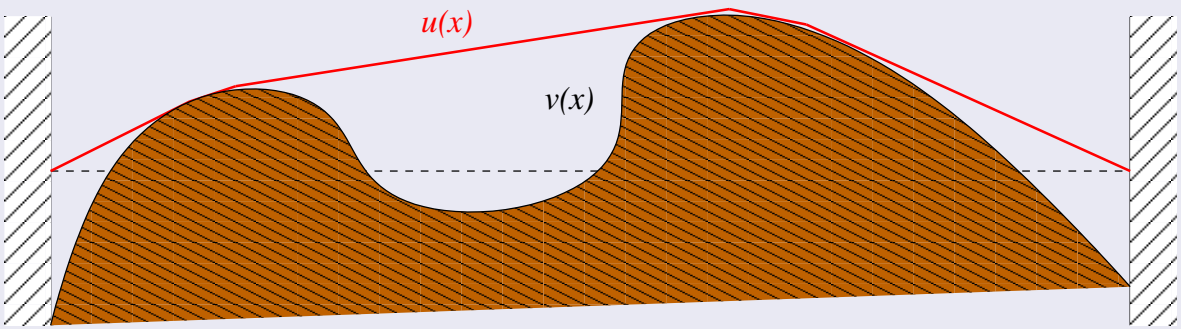
Temps de calcul sur processeur MIPS R12000 (500 MHz) - 2001

- 1 processeur = 13897,44 secondes (un peu moins de 4 heures) ;
- 16 processeurs = 2293,80 secondes (moins de 40 minutes)

Exemple en mécanique

Déformation d'une corde induite par un obstacle

- On considère un système comportant une corde de longueur 1 tendue avec une tension τ
  - Soit $u(x), x \in [0,1]$ le décalage par rapport à la position d'équilibre de la corde, l'énergie associée étant
    $$E(u) = \frac{1}{2} \int_0^1 \tau \left(\frac{du}{dx}\right)^2 dx$$
  - Soit $v(x) \geq 0$ la fonction modélisant un obstacle
- Déterminer la fonction de déformation $u(x)$ associée à $v(x)$
  - c'est trouver la fonction u minimisant $E(u)$;
  - sous la contrainte $u(x) \geq v(x), x \in [0,1]$



Formulation - terminologie

Description générale d'un problème d'optimisation (minimisation)

- Étant donné un ensemble Ω de
  - configurations ;
  - ou solutions admissibles

  du problème à résoudre et une fonction objectif f ;
- trouver le minimum x' de f par rapport à l'ensemble Ω
- Soit x' qui vérifie

$$x' \in \Omega \text{ et } f(x') = \min_{x \in \Omega} f(x)$$

Remarques

- La configuration x' peut ne pas être unique
- La fonction objectif est également appelée
  - fonction de coût ; fonctionnelle
  - fonction de perte (*loss function*) dans le domaine de l'IA
- Ω est aussi appelé l'espace de recherche

Formulation - terminologie

Passer de la minimisation à la maximisation

$$\max_{x \in \Omega} g(x) = -\min_{x \in \Omega}(-g(x))$$

Minimum local

- Une configuration x' vérifiant

$$\forall x \in V(x') \text{ on a } f(x') \leq f(x)$$

- Voisinage de taille ϵ

$$V(x') = \{x \in \Omega | ||x - x'|| < \epsilon\}$$

Minimum global

- Une configuration x' vérifiant

$$\forall x \in \Omega \text{ on a } f(x') \leq f(x)$$

Optimum ou extremum : un min. ou max., local ou global

Classification des méthodes d'optimisation

Dépend du point de vue considéré

Méthodes de recherche locale *vs.* de recherche globale

- Méthode de recherche locale
  - Repose sur la notion de voisinage
  - Exploration de proche en proche
- Méthode de recherche globale
  - Exploration par échantillonnage aléatoire

Méthodes d'optimisation locale *vs.* d'optimisation globale

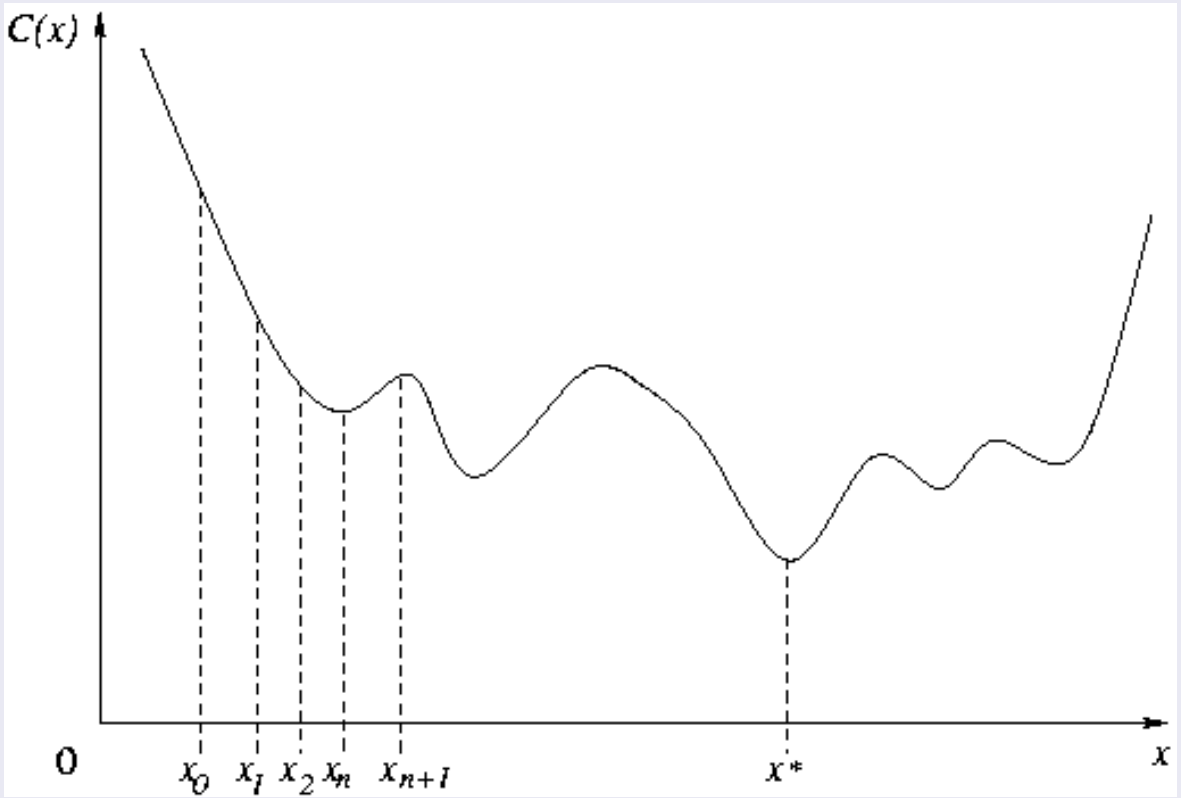
- Méthode d'optimisation locale
  - Piégée par le premier optimum rencontré
  - Une solution proche de l'optimum global n'est pas garantie en particulier quand l'espace de recherche est très grand
- Méthode d'optimisation globale
  - Toute méthode qui n'est pas sensible aux extrema locaux

Classification des méthodes d'optimisation

Dépend du point de vue considéré

Méthodes d'optimisation locale *vs.* d'optimisation globale (suite)

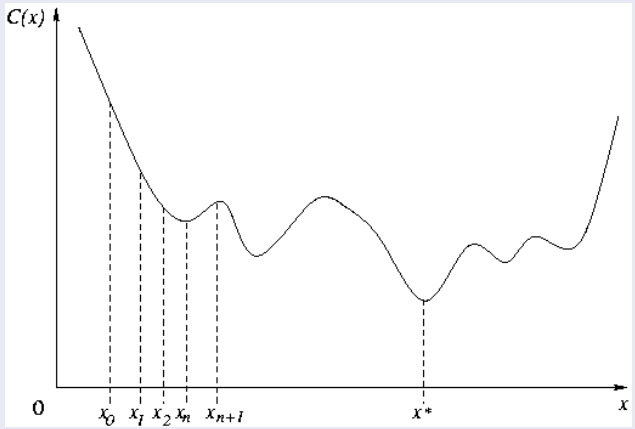
Piégeage d'un algorithme itératif de type gradient



Classification des méthodes d'optimisation

Méthodes d'optimisation locale *vs.* d'optimisation globale (suite)

Piégeage d'un algorithme itératif de type gradient



Méthodes déterministes *vs.* stochastiques

- Méthode déterministe
  - Efficace quand l'évaluation de la fonction objectif est rapide
  - ou que la forme de la fonction est connue
- Méthode stochastique
  - Efficace lorsqu'il y a de nombreux extrema
  - en cas de fonction objectif non dérivable ou bruitée

Classification des méthodes d'optimisation

Méthodes déterministes

- La recherche des extrema d'une fonction revient souvent à résoudre un système d'équations linéaires ou non-linéaires
- Exemples
  - Méthodes du gradient
  - Méthodes de relaxation (Gauss-Seidel, Jacobi)
  - etc.

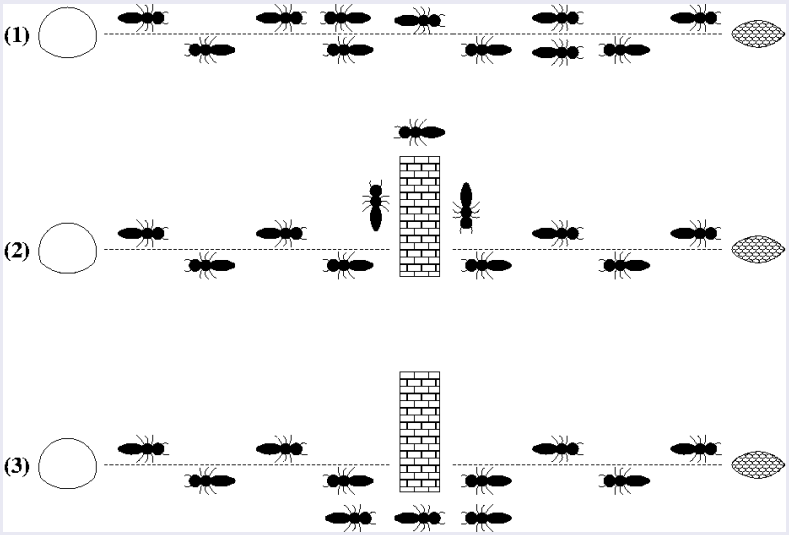
Méthodes stochastiques

- Utilisation de tirages aléatoires pour guider la recherche
- Modélisent souvent des phénomènes physiques ou biologiques
- Exemples - Métaheuristiques
  - Recuit simulé et variantes
  - Algorithmes évolutionnaires
  - Algorithmes de colonies de fourmis
  - etc.

Classification des méthodes d'optimisation

Méthodes stochastiques (suite)

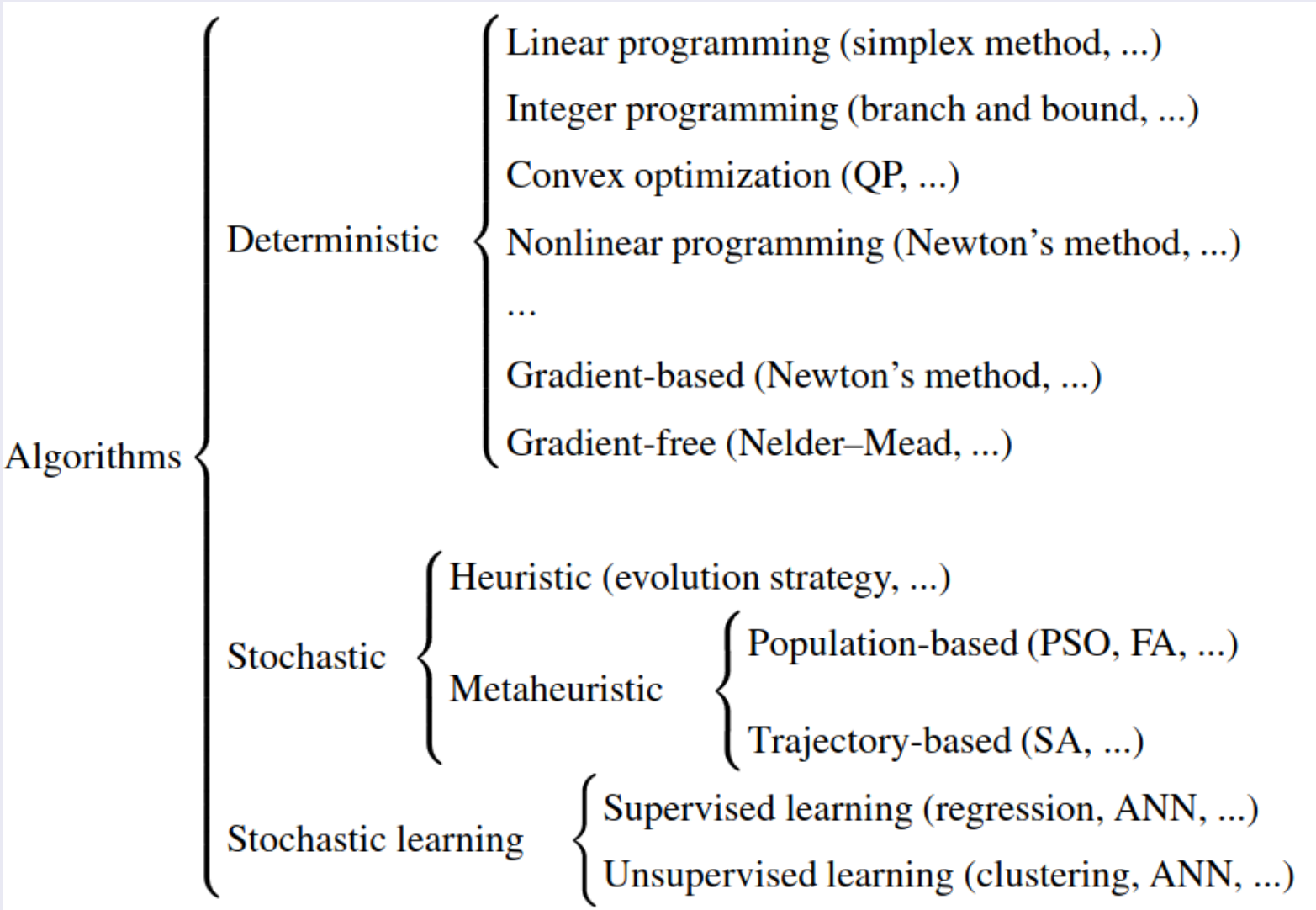
Une colonie de fourmis peut retrouver le chemin le plus court



1. Les fourmis suivent un chemin entre le nid et la nourriture
2. Obstacle $\rightarrow$ les fourmis prennent avec des probabilités égales un des deux chemins, la phéromone est déposée plus vite sur le chemin le plus court (plus d'allers-retours / de passages)
3. Toutes les fourmis choisissent le chemin le plus court

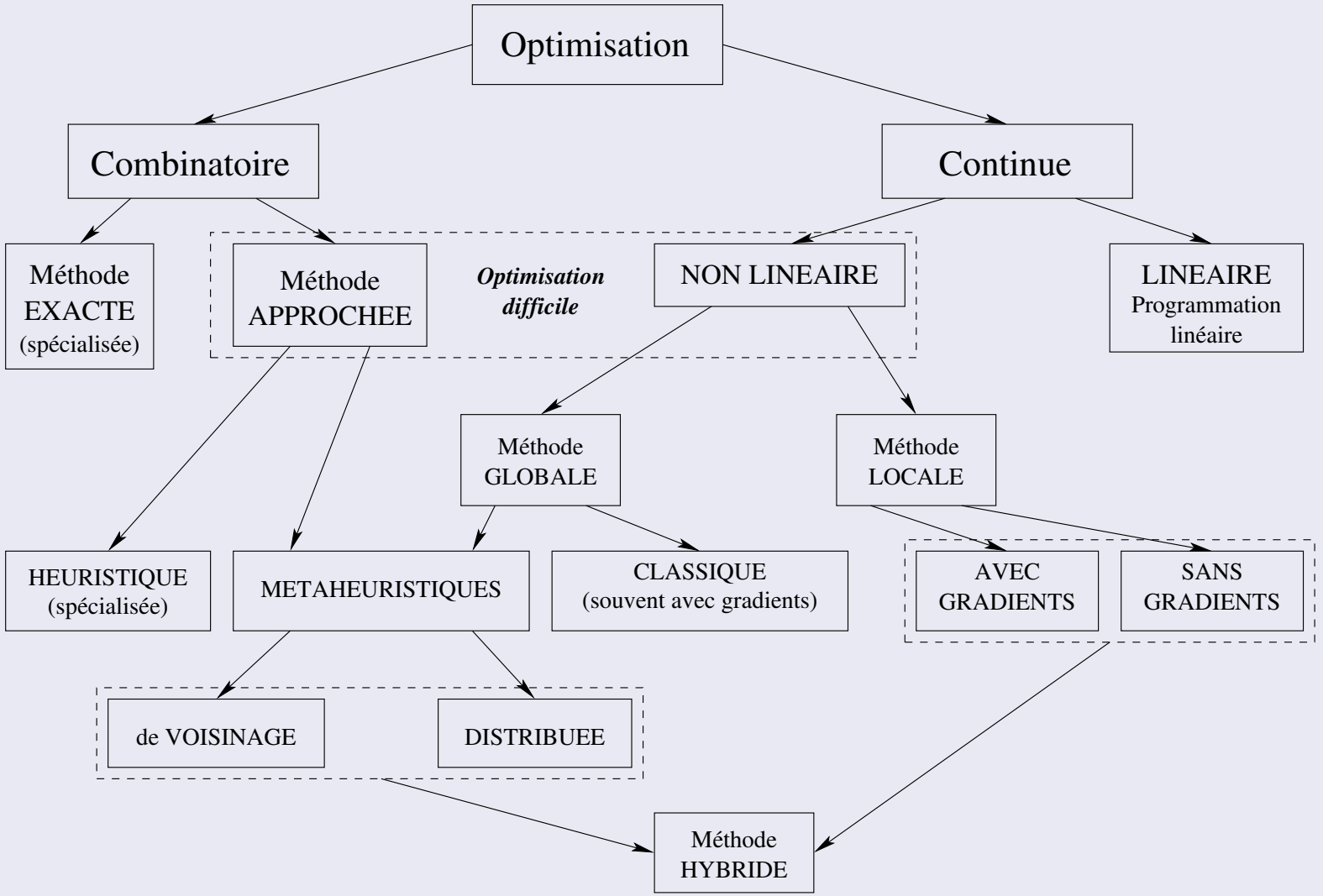
Classification des méthodes d'optimisation

Classification déterministe *versus* stochastique



Classification des méthodes d'optimisation

Classification générale

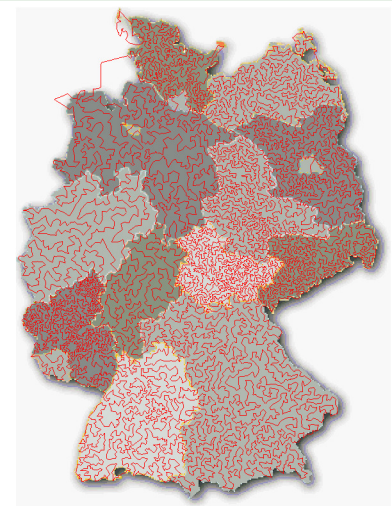


Optimisation combinatoire *versus* optimisation continue

Probléme d'optimisation combinatoire

- Ω est un sous-ensemble de $\mathbb{N}^n$ ou $\mathbb{N}^p \times \mathbb{R}^q$
- L'espace de recherche est fini ou dénombrable, mais non énumérable en un temps "raisonnable"

Problème combinatoire classique : le voyageur de commerce



n villes $\Rightarrow \frac{(n-1)!}{2}$ circuits possibles

- Circuit reliant 15112 villes en Allemagne, *Applegate et al.* 2001 (Princeton University)
- Temps de calcul cumulé sur 1 proc. Alpha EV6 (500 MHz) = 22,6 années

Circuit de 85900 villes calculé en 2006

Problème d'optimisation continue

- Ω est fréquemment un sous-ensemble de $\mathbb{R}^n$

Problème d'optimisation avec ou sans contraintes

Formulation générale des problèmes d'optimisation non linéaire

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} f(x) \\ \text{sous les contraintes} \\ \quad g(x) \leq 0 \qquad (A) \\ \quad h(x) = 0 \qquad (B) \end{cases} \tag{1}$$

- où les fonctions f, g et h sont non linéaires ;
- l'inéquation 1.A spécifie des contraintes d'inégalité ;
- l'équation 1.B désigne des contraintes d'égalité

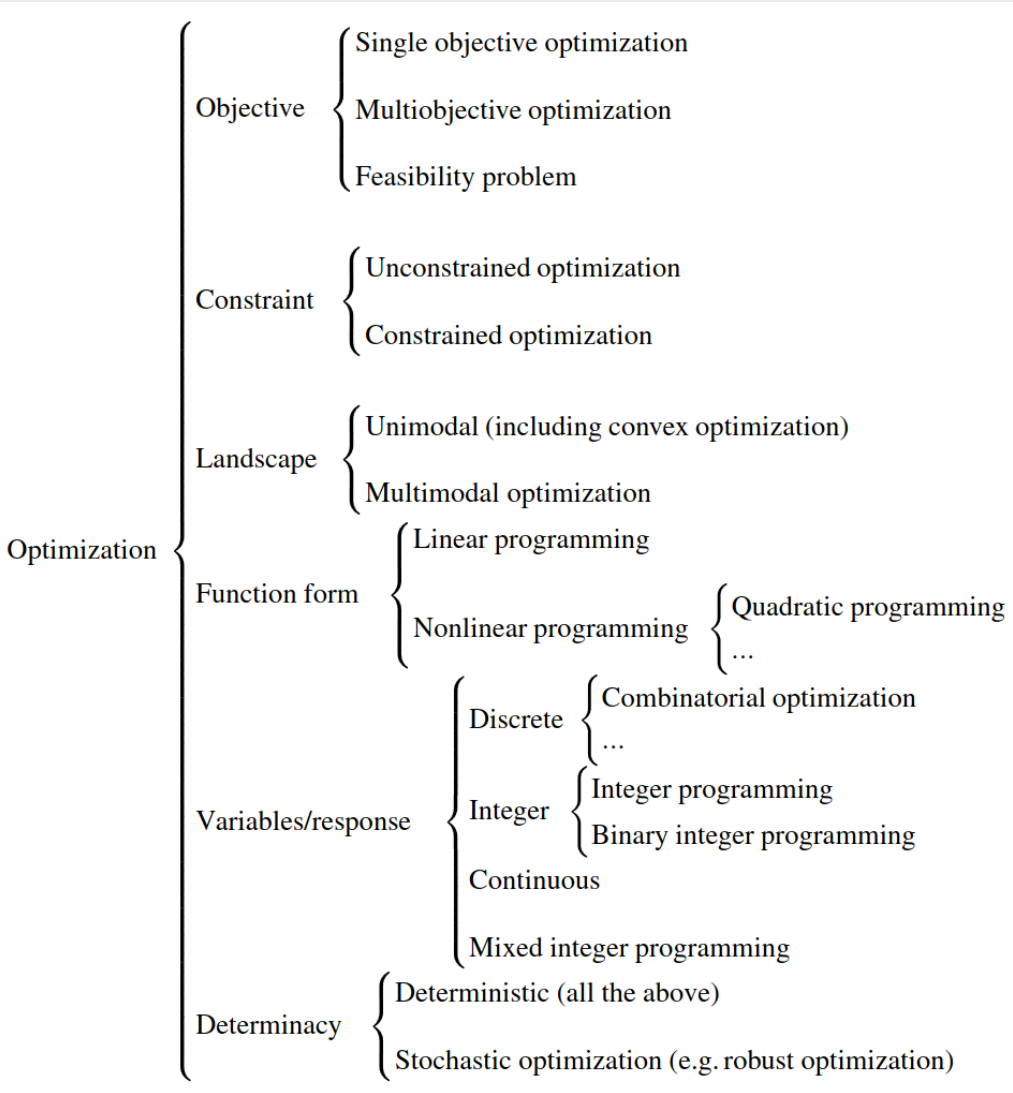
Différents types de problème

- Problème **avec contraintes d'inégalité et d'égalité**
- Problème **avec contraintes d'inégalité ou d'égalité**
- Problème **sans contraintes**

Le problème initial doit être mis sous une de ces formes

Classification des problèmes d'optimisation

Classification générale



Généralités sur les métaheuristiques

- Généralement des algorithmes stochastiques itératifs
- Grande variété d'algorithmes
  - Algorithme du type recuit simulé
    - Recherche monopoint (une seule solution potentielle)
    - Preuve de convergence
      - Recuit simulé classique
      - Équation de la diffusion
      - Recuit adaptatif
  - Recherche tabou
  - Algorithmes évolutionnaires (ou évolutionnistes)
    - Recherche multipoint (plusieurs solutions potentielles)
    - Preuve de convergence
      - Algorithmes génétiques
      - Stratégies d'évolution
      - Évolution différentielle
  - etc.

Généralités sur les métaheuristiques

- Stratégies adoptées pour s'échapper d'un minimum local
  - Métaheuristiques dites de voisinage (*Trajectory-based*)
    - Algorithmes d'optimisation reposant sur la notion de voisinage
      • Exemple : recuit simulé
    - Principe :
      • autoriser, de temps en temps, une dégradation temporaire lors du changement de la configuration courante ;
      • un mécanisme de contrôle de la dégradation est prévu pour éviter la divergence de l'algorithme
  - Métaheuristiques distribuées (*Population-based*)
    - Algorithmes d'optimisation dits "distribués"
      • Exemple : algorithmes évolutionnaires
    - Principe :
      • le contrôle d'une population de solutions potentielles permet d'éviter les pièges de minima locaux ;
      • intégration de mécanismes permettant à l'algorithme de générer de nouvelles solutions par "reproduction"

Généralités sur les métaheuristiques

- Adaptation aux problèmes à variables continues
  - Définir une stratégie de discrétisation des variables si besoin
  - Idéalement le pas de discrétisation devrait s'adapter au cours de l'optimisation
  - Inversement, il est possible de résoudre un problème discret avec une méthode continue
- Optimisation multiobjectif
  - Problème nécessitant la considération de différents objectifs contradictoires $\rightarrow$ pas d'optimum unique
- Méthodes hybrides
  - Combiner des métaheuristiques complémentaires
- Parallélisation
  - Traitement de problèmes de grande taille
  - Réduction des temps de calcul

Adéquation algorithme (ou méthode) / problème

Étant donné un problème d'opt., comment choisir une méthode de résolution efficace ?

- Capable de produire une solution "optimale" ou acceptable ;
- avec un temps de calcul raisonnable

Sujet ouvert

- Pas de "recette miracle" : pas de règles
  - pour le choix d'un algorithme d'optimisation ;
  - ni, en général, pour le réglage optimal de ses paramètres
- Résultats théoriques inexistants ou inapplicables (hypothèses)

Un algorithme d'optimisation meilleur que tous les autres ?

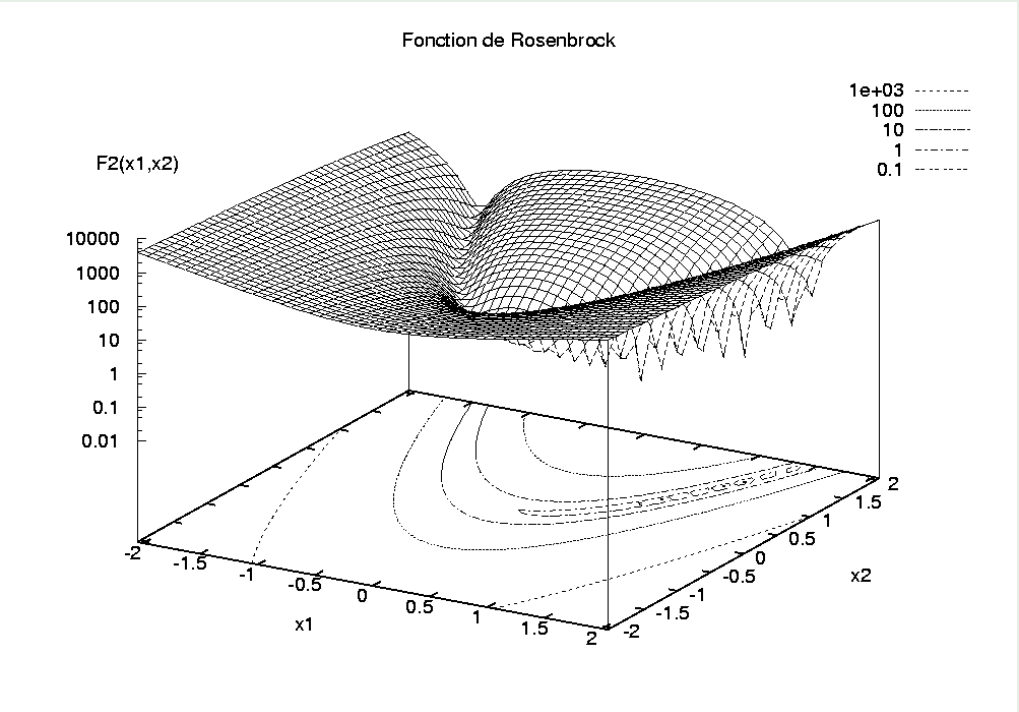
- Comparaison sur des fonctions de coût "synthétiques", c'est-à-dire ne modélisant pas de problème réel
  - Variété de fonctions tests classiques (Rosenbrock, etc.)
  - Choix des fonctions tests pas aisé
  - Approche intéressante **uniquement** si une fonction test approxime convenablement la fonction de coût du problème

Adéquation algorithme (ou méthode) / problème

Un algorithme d'optimisation meilleur que tous les autres ?

- Exemples 2D de fonctions tests (ou synthétiques)
  - Fonction de Rosenbrock

$$F2(x_1, x_2) = 100 \times \left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 ; x^* = (1, 1)$$



Adéquation algorithme (ou méthode) / problème

Un algorithme d'optimisation meilleur que tous les autres ?

- Exemples 2D de fonctions tests (ou synthétiques)
  - Fonction de Rosenbrock

$$F2(x_1, x_2) = 100 \times \left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 \, ; x^* = (1, 1)$$

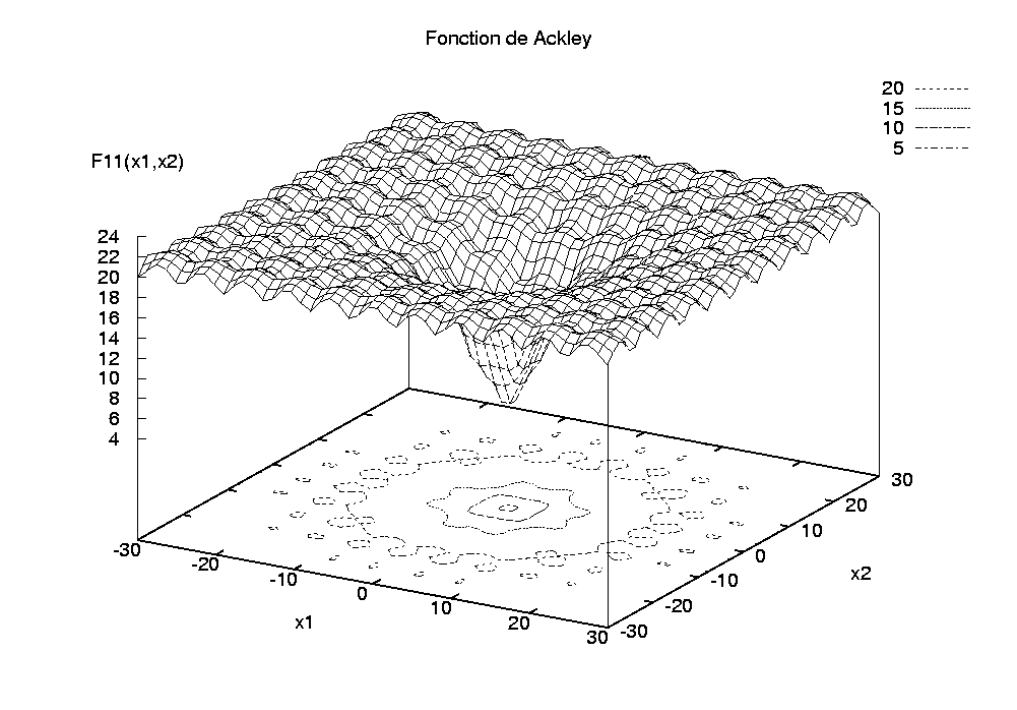
  - Fonction d'Ackley

$$\begin{aligned} F11(x_1, x_2) &= -20 \times \exp\left(-0,2 \times \sqrt{\frac{1}{2}\left(x_1^2 + x_2^2\right)}\right) \\ &- \exp\left(\frac{1}{2}\left(\cos\left(2\pi x_1\right) + \cos\left(2\pi x_2\right)\right)\right) \\ &+ 20 + e; \\ x^* &= (0, 0) \end{aligned}$$

Adéquation algorithme (ou méthode) / problème

Un algorithme d'optimisation meilleur que tous les autres ?

- Exemples 2D de fonctions tests (ou synthétiques)
  - Fonction d'Ackley



Adéquation algorithme (ou méthode) / problème

Un algorithme d'optimisation meilleur que tous les autres ?

- *No Free Lunch Theorem* (Wolpert et Mac Ready - 1997)

  En "moyenne", sur toutes les fonctions de coût possibles, tous les algorithmes d'optimisation ont des performances équivalentes avec un temps d'exécution fini

  - Soit :
    - pas de meilleur algorithme d'optimisation quel que soit le problème considéré ;
    - en revanche, pour un problème donné il existe bien un "meilleur" algorithme
- Optimisation efficace

  Choisir un algorithme d'optimisation appréhendant la structure mathématique du problème

  - de l'espace de recherche ;
  - de la fonction de coût à optimiser

Adéquation algorithme (ou méthode) / problème

Conclusion

- Pas de règle établie
- Appel au savoir-faire et à l'expérience de l'utilisateur
- Essayer plusieurs algorithmes est souvent nécessaire avant de trouver le "bon"
  - Comparaison implicite d'algorithmes sur le problème considéré

Approche la plus prometteuse

Guider le choix par l'analyse du "paysage"
de la fonction de coût (régularité, dérivabilité, etc.)