CHAPITRE 1: OPTIMISATION COMBINATOIRE ET LES MÉTA-HEURISTIQUES

COURS : MÉTAHEURISTIQUES ET ALGORITHMES ÉVOLUTIONNISTES

RÉALISÉ DR. S. SLATNIA

UNIVERSITÉ DE BISKRA
Master d'informatique
2020-2021

PLAN DU CHAPITRE

1. Définitions
2. Difficulté des problèmes réels
   - 2.1. Taille de l'espace de recherche
   - 2.2. Modélisation du problème
3. Comment traiter les problèmes NP-difficiles ?
   - 3.1. Méthodes de résolution
   - 3.2. Quelques techniques de l'IA (thématiques)
4. Optimisation combinatoire (discrète).
5. Problème d'optimisation
   - 5.1. Formulation d'un PO
   - 5.2. Difficulté d'un POC
6. Classification des algorithmes d'optimisation
   - 6.1. Méthodes exactes
   - 6.2. Méthodes approchées
7. Les méta-heuristique
8. Deux principes de base pour une recherche heuristique, la Diversification et l' Intensification

DÉFINITIONS

I. Problème de recherche (PR)

- Un PR est un problème algorithmique associé à une relation binaire.

- Si **R** est une **relation binaire** telle que pour tout (R) ⊆ Γ+ et T une machine de turing, alors T implante *R* si:

  - Si *x* est tel qu'il existe un y vérifiant *R*(*x*, *y*) alors *T* accepte l'entrée *x* en produisant un résultat *z* tel que *R*(*x*, *z*)
  - Si *x* est tel qu'il n'existe aucune *y* tel que *R*(*x*, *y*) alors *T* rejette l'entrée *x*

  [Wiki].

DÉFINITIONS

❑ But de PR

- Trouver une solution lorsqu'un algorithme n'est pas fourni, mais lorsqu'on a que la spécification de la solution.

❑ Définition de PR

Un problème de recherche est défini par:

- Un ensemble d'états
- Un état de départ
- Un état cible ou état-test
- Une fonction booléenne qui permet de savoir un état donné est l'état-cible
- Une fonction successeur

  Une application d'un état vers l'ensemble des états possibles

DÉFINITIONS

2. Problème de Décision (PD)

- Un **problème de décision** est une question mathématique dont la réponse est soit « oui », soit « non ».
- Les logiciens s'y sont intéressés à cause de l'existence ou de la **non-existence d'un algorithme** répondant à la question posée.

- En théorie de la **calculabilité**, formuler un problème de décision c'est se poser une question de **décidabilité**.
  - Il s'agit en fait de rechercher l'existence d'un algorithme résolvant le problème et s'il existe de l'expliciter.
  - Dans le cas où, il n'existe pas d'algorithme résolvant le problème, celui-ci est dit **indécidable**.
- Il y a des problèmes fondamentaux qui

DÉFINITIONS

3. Problème d'Optimisation (PO)

❑ Optimisation et Wikepedia : Une définition

- L'**optimisation** est une branche des mathématiques cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou maximiser une fonction sur un ensemble.

- L'optimisation joue un rôle important en recherche opérationnelle, dans les mathématiques appliquées, en analyse, en statique pour l'estimation du maximum de vraisemblance d'une distribution, pour la recherche de stratégies dans le cadre de la théorie des jeux, ou encore en théorie du contrôle et de la commande.

DÉFINITIONS

PO

- Aujourd'hui, tous les systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés.
- La qualité des résultats et des prédictions dépend de la pertinence du modèle, de l'efficacité de l'algorithme et des moyens pour le traitement numérique.

❑ **Formellement,**

L'**optimisation** est l'étude des problèmes qui s'expriment de la manière suivante [Wiki].

Problème d'optimisation — Étant donné une fonction $f : A \to \mathbb{R}$ définie sur un ensemble A à valeurs dans l'ensemble $\mathbb{R}$ des nombres réels (éventuellement dans la droite achevée $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$), trouver un élément $\bar{x}$ de A tel que $f(\bar{x}) \leqslant f(x)$ pour tous les x dans A.

DÉFINITIONS

Définition 1. [S. Douir et all]

Classe P

- Un algorithme **en temps polynomial** est un algorithme dont le temps de la complexité est en O(p(n)), où **p** est une fonction polynomiale et **n** est la taille de l'instance (ou sa longueur d'entrée).

- Si k est le plus grand exposant de ce polynôme en n, le problème correspondant est dit être résolu en $O(n^k)$ et appartient à la **classe P**,

  - Un exemple de problème polynomial
    - La connexité dans un graphe.

DÉFINITIONS

Définition 2.

La classe NP

- **La classe NP** contient **les problèmes de décision** qui peuvent être décidés sur une machine non déterministe en temps polynomial.
- C'est la classe des problèmes qui admettent **un algorithme polynomial** capable de tester la validité d'une solution du problème.
- Les problèmes de cette classe sont les problèmes qui peuvent être résolus en **énumérant l'ensemble de solutions** possibles et en les **testant** à l'aide d'un **algorithme polynomial**.

DÉFINITIONS

Définition 3.

- On dit qu'un problème de recherche P1 se réduit polynomialement à un problème de recherche P2 par la réduction de Turing s'il existe un algorithme A1 pour résoudre P1 utilisant comme sous programme un algorithme A2 résolvant P2 , de telle sorte que la **complexité** de A1 est **polynomiale**, quand on évalue chaque appel de A2 par une constante.

DÉFINITIONS

Définition 4.

La classe NP-complet

- **parmi l'ensemble des problèmes appartenant à NP,** il en existe un sous ensemble qui contient les problèmes les **plus difficiles** : on les appelle les problèmes **NP-complets**.

- Un **problème NP-complet** possède la propriété que tout problème dans **NP** peut être transformé (réduit) en celui-ci en **temps polynomial**.
  - **NP-complet** quand tous les problèmes appartenant à **NP** lui sont **réductibles**.
  - Si on trouve **un algorithme polynomial** pour un problème **NP-complet**, on trouve alors automatiquement une résolution polynomiale de tous les problèmes de la **classe NP**.

DÉFINITIONS

Définition 5.

La classe NP-difficile

- **Un problème est NP-difficile s'il est plus difficile qu'un problème** NP-complet,
  - S'il existe un problème **NP-complet** se réduisant à ce problème par une réduction de Turing.

DIFFICULTÉ DES PROBLÈMES RÉELS

- **Nombre important de solutions possibles dans l'espace de recherche** → Recherche exhaustive impossible.
- **Problème complexe** → Utilisation de **modèles simplifiés**.
- **Fonction d'évaluation** qui décrit la qualité des solutions et dynamique (fonction du temps) → Plusieurs solutions à trouver.
- **Problèmes avec des contraintes fortes** → Construire une solution réalisable est difficile.
- **Multi-critère** [E.Talb], ...

TAILLE DE L'ESPACE DE RECHERCHE

- **Problème SAT :** NP-difficile

Problème de satisfaisabilité booléenne (PD).

- Plusieurs applications : emploi du temps, routage, …
- **Trouver** un ensemble de variables booléennes

$$X=(X1, \ldots, Xn)$$

pour qu'une expression booléenne donnée soit= TRUE

- L'expression booléenne doit être une conjonction de clauses;

Les clauses sont représentées par des disjonctions de k variables (k-SAT) : $F(X) = (\overline{X}_1 \vee X_3) \wedge (X_1 \vee \overline{X}_2) \wedge (X_4 \vee X_2) \wedge \ldots$

- Pour n=100, taille espace = $2^{100} \approx 10^{30}$,
- k>2, problème NP-difficile; k=2, Algorithme polynomial [E.Talb].

TAILLE DE L'ESPACE DE RECHERCHE

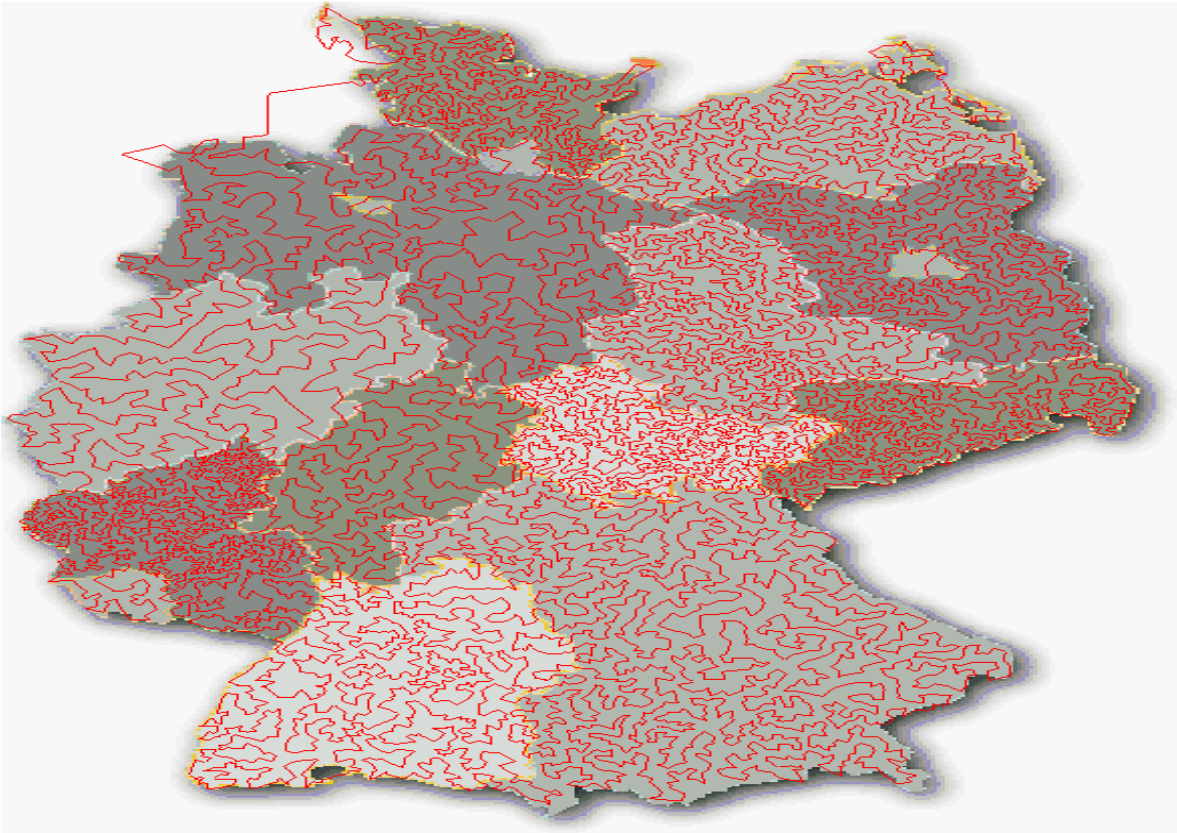
- **Problème TSP :** NP-difficile

n villes, trouver un circuit qui minimise la distance totale parcourue; toutes les villes doivent être visitées une seule fois.

- TSP symétrique dist(x,y)=dist(y,x)
- |S| = n!/2n = (n-1)! / 2;
- 10 villes = 181 000 solutions
- 20 villes = 10 000 000 000 000 000 solutions
- 50 villes = 100 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 solutions

EXEMPLES

PROBLÈME DU VOYAGEUR DE COMMERCE [M. SALO].



Circuit reliant 15112 villes

en Allemagne

MODÉLISATION DU PROBLÈME

- Dans la réalité, nous trouvons une solution à un modèle du problème.
- Tous les modèles sont des simplifications de la réalité.
- Problème ==> Modèle ==> Solution
- SAT, TSP, et NLP sont 3 formes canoniques de modèles qui peuvent être appliqués à différents problèmes.

❖ **<u>Contraintes</u>**

- La plupart des problèmes réels possèdent des contraintes

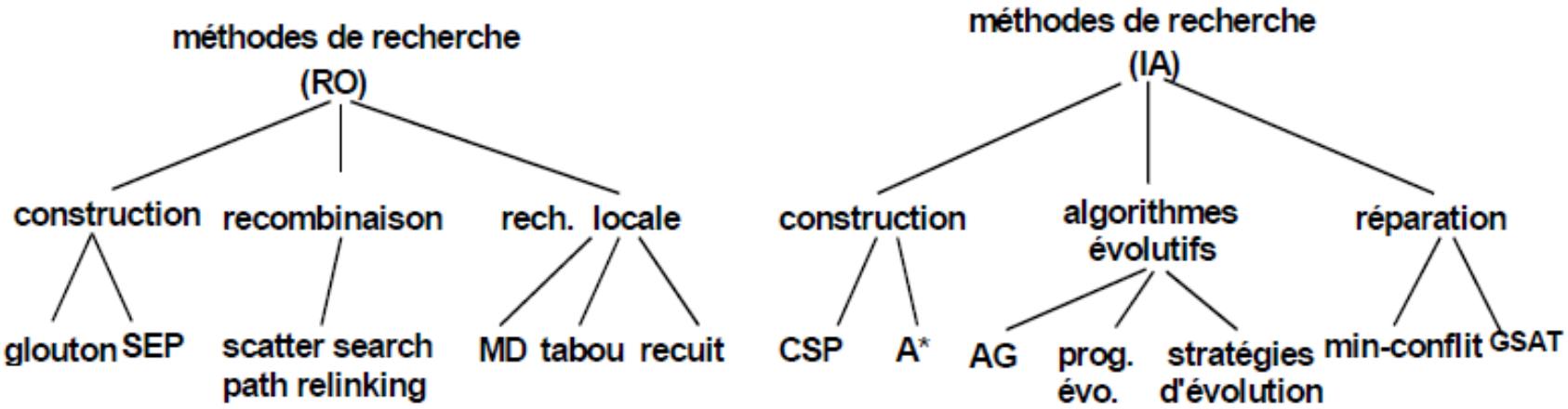
[E. Talb].

COMMENT TRAITER LES PROBLÈMES NP-DIFFICILES ?

Pour des problèmes d'optimisation [H. Fior].

- **Par approximation**
  - Algorithmes en temps polynomial garantissant de trouver des "**bonnes solutions**".
- **Par des heuristiques**
  - Approches "**intelligentes" donnant souvent** des solutions mais pouvant **échouer**.

MÉTHODES DE RÉSOLUTION



Quatre grandes approches : **[J. Hao]**

1. **Approche de construction**
   - L'instanciation successive des variables selon un ordre statique ou dynamique
   - Si exacte (complète), alors complexité exponentielle dans le pire des cas
   - **Exemple** : branch& bound, CSP, méthodes gloutonnes...

MÉTHODES DE RÉSOLUTION

2. Approche de recherche locale ou voisinage

- Réparation itérative d'une config. complète par des modifications locales
- Non-exacte
- **Exemple** : descente, recuit simulé, recherche tabou, mais aussi min-conflit...

3. Approche d'évolution

- Evolution d'une population de solutions par des opérations "génétiques" (sélection, croisement, et mutation)
- Non-exacte
- **Exemple** : algo. génétiques, stratégies d'évolution, programmation évolutive...

4. Hybridation

Combinaison de différentes approches.

QUELQUES TECHNIQUES DE L'IA (THÉMATIQUES)

1. Algorithmes d'exploration : [L. Lamo]

ceci inclut les techniques suivantes :

- **L'exploration sans information**
  - En profondeur,
  - En largeur,
  - Min-Max
- **L'exploration avec information**
  - A*,
  - *hill-climbing,*
  - *Algorithmes Génétiques*
- **Les problèmes de satisfaction de contraintes (*CSP*)**

QUELQUES TECHNIQUES DE L'IA (THÉMATIQUES)

2. **Représentation des connaissances et raisonnement automatique**

   - Logiques,
   - Agents et SMA,
   - Réseaux bayésiens,

3. **Apprentissage**

   - Apprentissage Symbolique Automatique,
   - Algorithmique évolutionnaire,
   - Réseaux de neuronnes,

4. **Perception, action**

   - Planification automatique,
   - Robotique.

OPTIMISATION COMBINATOIRE

- La programmation informatique [S. Canu].

- **Optimisation combinatoire**

Est une branche de l'optimisation en mathématiques appliquées et en informatique, également liée à la recherche opérationnelle, l'algorithmique et la théorie de la complexité.

OPTIMISATION COMBINATOIRE

- L'optimisation combinatoire consiste à trouver dans un ensemble un sous-ensemble contenant les «meilleures solutions ».
- Trouver une solution optimale dans un ensemble discret et fini est un problème facile en théorie,
- Il suffit d'essayer toutes les solutions, et de comparer leurs qualités pour voir la meilleure.
- L'explosion combinatoire de solutions possibles de certains problèmes mathématiques ne permettent pas d'obtenir de solution en un temps « humain ».

OPTIMISATION COMBINATOIRE

- **Problème d'optimisation (PO)**

  Un PO consiste à trouver, parmi un ensemble donné, un élément (un vecteur de IR^P, un vecteur d'entiers, une fonction...) minimisant ou maximisant une fonction donnée de cet ensemble sur IR [S. Canu].

OPTIMISATION COMBINATOIRE

Définition.

Un **problème d'optimisation combinatoire** [S. Douir et all] peut être défini par :

- Vecteur de variables **x = (x1, x2, ..., xn),**
- Domaine des variables **D = (D1, D2, ..., Dn),** où **les (Di)i=1,...,n** sont des ensembles finis,
- Ensemble de contraintes,
- Une fonction objectif f à **minimiser** ou à **maximiser**,
- Ensemble de toutes les solutions réalisable possibles est

  S = {x =(x1, x2, , xn) ϵ D / x satisfait toutes les contraintes}, l'ensemble S est aussi appelé un espace de recherche.

EXEMPLES D'UN PO

Exemples de problèmes d'optimisation

chaine logistique (*suply chain*)

équilibrage d'un réseau (électricité, informatique. . .) (*load balancing*)

planification des vols (emplois du temps) *scheduling*

calcul de trajectoire

Formalisation des problèmes

Classification des problèmes

FORMULATION D'UN PO

❖ Minimisation

étant donné un couple **(S,f)** où

- **S** un ensemble fini de solutions ou configurations (espace de recherche)
- **f : S→R** une fonction de coût (ou objectif)

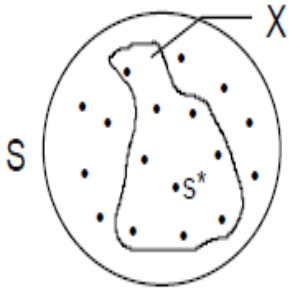
trouver **s*∈X ⊆S** tel que **f(s*) ≤f(s) ∀s ∈X** (config. faisables ou réalisables)

Remarques:

- pour la **maximisation**, il suffit de remplacer

  f(s*) ≤f(s) par f(s*) ≥f(s)

- **S** et **f** ne sont pas nécessairement donnés à l'avance
- la plupart des pbm d'optimisation intéressants sont dans la classe NP-difficile **[J. Hao].**



Exemple. Voyageur de commerce (TSP)

DIFFICULTÉ D'UN POC

- Problème d'optimisation combinatoire (POC) [M. Salo]
  - Espace de recherche fini ou dénombrable, mais non énumérable en un temps « raisonnable »
  - Fonction de coût à minimiser ou maximiser

- Difficulté d'un problème d'optimisation combinatoire
  - **Taille de l'espace de recherche**
  - **« Paysage » de la fonction de coût**

CLASSIFICATION DES ALGORITHMES D'OPTIMISATION

- **Variable suivant le point de vue considéré** [M. Salo]
  - **Algorithmes déterministes / algorithmes stochastiques**
  - **Algorithmes de recherche locale / algo. de recherche globale**
  - **Algorithmes d 'optimisation locale / algo. d'optimisation globale**

❖ **Algorithmes d'optimisation locale**

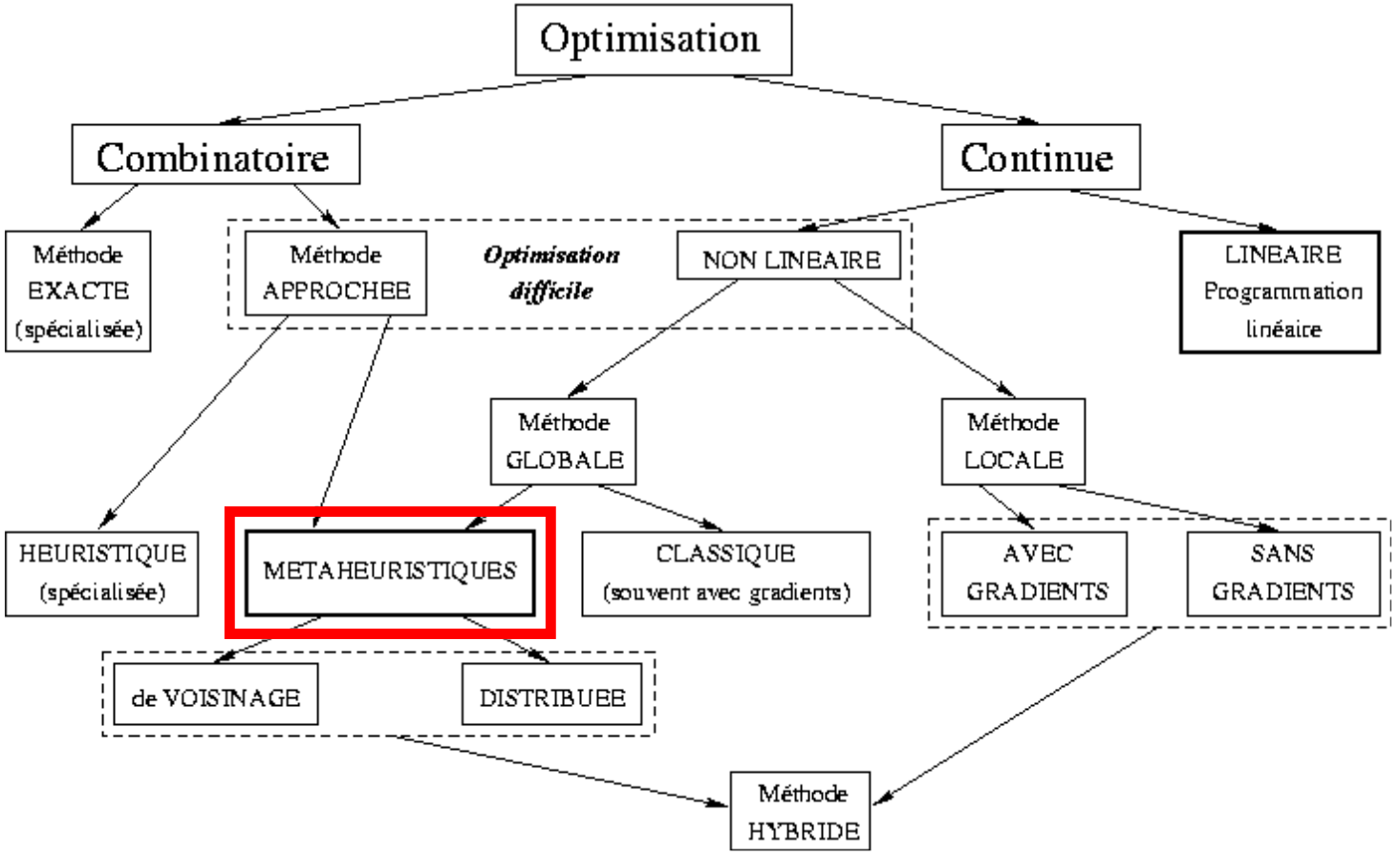
- Tout algorithme piégé par le premier optimum rencontré;
- Ne permettant pas d'obtenir une solution proche de l'optimum global en raison de la trop grande cardinalité de l'espace de recherche

❖ **Algorithmes d'optimisation globale**

- Tout algorithme qui n'est pas sensible aux minima locaux;
- Algorithme permettant d 'obtenir une solution proche de l'optimum global

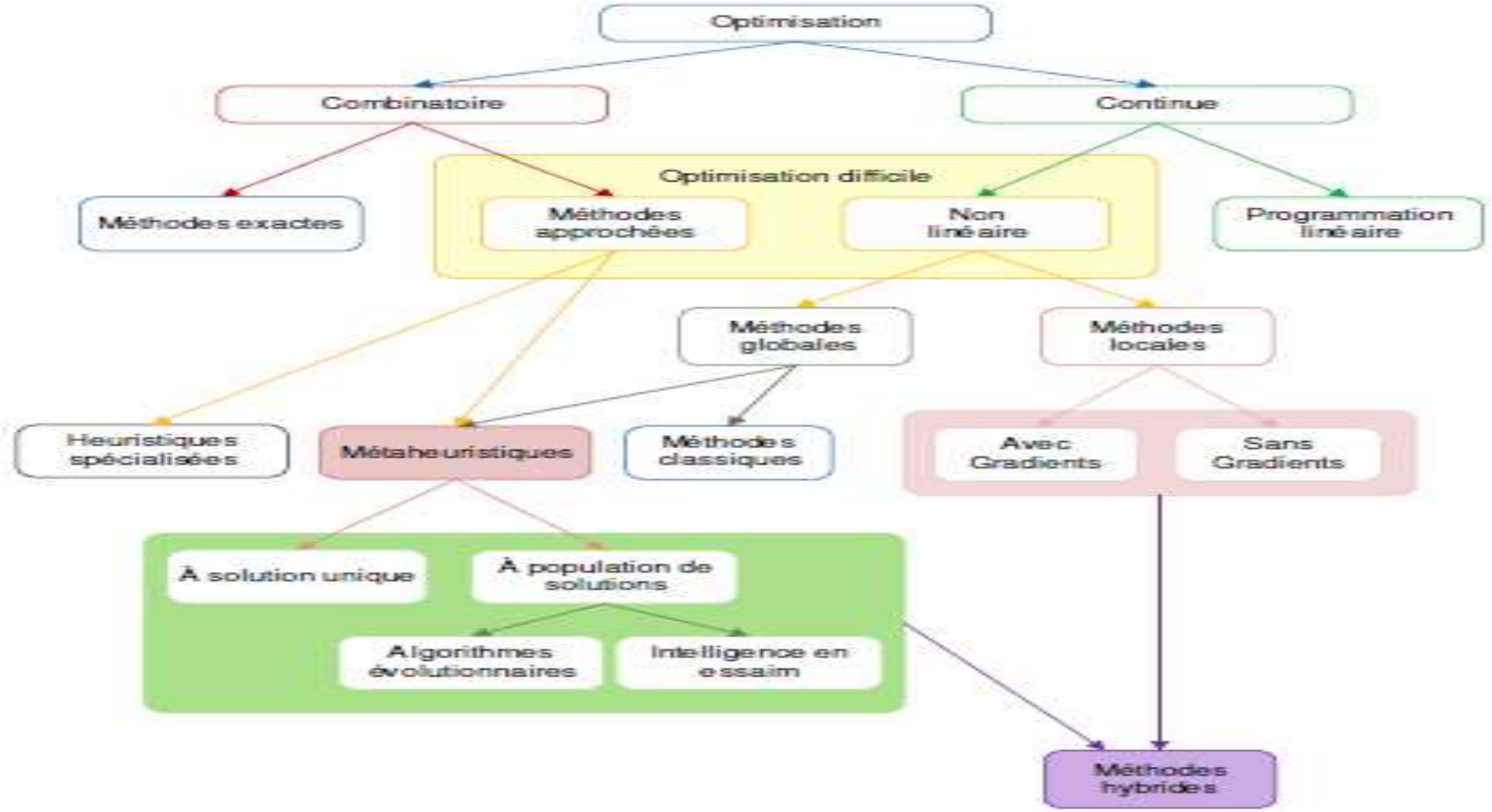
CLASSIFICATION DES ALGORITHMES D'OPTIMISATION

❖ **Classification générale des méthodes [M. Salo].**



CLASSIFICATION DES ALGORITHMES D'OPTIMISATION

❖ **Classification générale des méthodes [I. Bous].**



CLASSIFICATION DES ALGORITHMES D'OPTIMISATION

- **Multitude d'algorithmes d'optimisation combinatoire** [M. Salo]

1. **Méthodes exactes**
   - Programmation dynamique
   - Recherche arborescente

   …
2. **Méthodes approchées - heuristiques / métaheuristiques**
   - Recuit simulé et variantes
   - Algorithmes évolutionnaires
   - Algorithmes de colonies de fourmis

   …

MÉTHODES EXACTES

- Quelques problèmes d'optimisation combinatoire peuvent être résolus (de manière exacte) en **temps polynomial**

Exemple

Un algorithme de programmation dynamique ou en montrant que le problème peut être formulé comme un problème d'optimisation linéaire en variables réelles.

- Si Le problème est **NP-difficile**, pour le résoudre, il faut faire appel à des **algorithmes adaptés.**
  - ❖ En pratique, la complexité physiquement acceptable n'est souvent que polynomiale.

MÉTHODES EXACTES

- Permettent d'obtenir une solutions dont **l'optimalité est garantie** [S. Douir et all],

- Dans certaines situations, on peut cependant chercher des solutions de **bonne qualité**, sans garantie d'optimalité, mais au profit d'un **temps de calcul plus réduit**.

- Pour cela, On applique des méthodes appelées

MÉTHODES APPROCHÉES

- On se contente alors d'avoir une **solution approchée** au mieux, obtenue par une heuristique ou une méta-heuristique.
- Il est important de noter que certaines de ces méthodes fournissent des **optimaux globaux,**

➢ La différence avec les méthodes exactes est le fait de ne pas avoir de preuve formelle (preuve mathématique ou de finalité) de son optimalité globale.

- **Champs d'applications privilégiés**

problèmes de recherche difficiles qu'on ne sait pas traiter autrement

MÉTHODES APPROCHÉES

❖ **Heuristique** [J. Hao]

Une méthode approchée conçue pour un problème produire des solutions non nécessairement optimales (avec un temps de calcul raisonnable).

❖ **Métaheuristique**

- **Contrôle** et **guide** une heuristique interne de recherche locale adaptée au problème à résoudre pour lui permettre de transcender les optima locaux [Ham et all].
- Une stratégie (règle) de choix **pilotant** une heuristique
- Un schéma de calcul heuristique, général et adaptable à un ensemble de problèmes différents
- Adaptées à chaque problème traité, avec cependant l'inconvénient de ne disposer en retour d'aucune information sur la qualité des solutions obtenues.

LES MÉTAHEURISTIQUES/SOLUTION UNIQUE

1. **<u>Méthodes minimum locales</u> (méthode à une solution)**

- Méthode de recherche locale (recherche par voisinage) converge vers un minimum local.

➢ **Principe**

- Partent d'une solution initiale et, par raffinements successives, construisent des suites de solutions de couts décroissants pour un problème de minimisation.
- Le processus s'arrête lorsqu'on ne peut plus améliorer la solution courante ou parce que le nombre maximal d'itérations (fixé au départ) est atteint.

LES MÉTAHEURISTIQUES/SOLUTION UNIQUE

❖ **Avantage**

- Les premières méthodes testées sur les nouveaux problèmes combinatoires émergeant des applications réelles.

❖ **Inconvénient**

- Ne soit pas complète (rien n'assure qu'elles pourront trouver toutes les solutions existantes).
- Ni n'assurent la preuve d'optimalité
- Parviennent très souvent à trouver des solutions de bonne qualité dans des temps de calcul raisonnables.

LES MÉTAHEURISTIQUES/SOLUTION UNIQUE

- ***Exemples***
  - Recuit simulé
  - La méthode de la Descente
  - La recherche Tabou

LES MÉTAHEURISTIQUES/ POP SOLUTION

2. **<u>Méthodes optimum globales</u> (méthode à population de solutions)**

   Atteindre un ou plusieurs optima globaux, sont d'une grande diversité pour Résoudre des problèmes combinatoires

❖ **Exemples**

- Les algorithmes génétiques
- Les algorithmes à évolution différentielle
- La recherche dispersée

LES MÉTAHEURISTIQUES

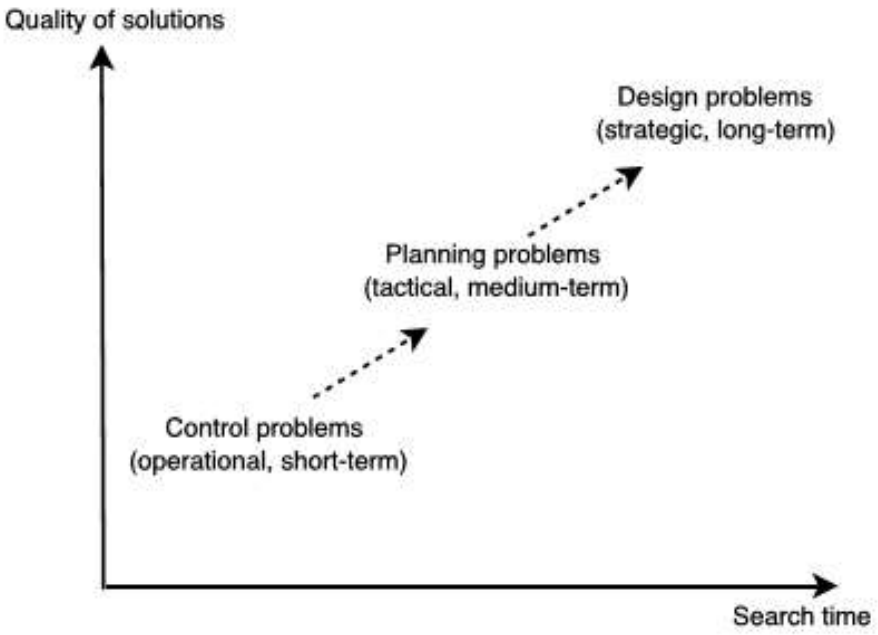


Figure. Different classes of problems in terms of the trade-off between quality of solutions and search time: design (strategic, long-term), planning (tactical, medium-term), control (operational, short-term) [Talbi].

DEUX PRINCIPES DE BASE POUR UNE RECHERCHE HEURISTIQUE

❑ Intensification (ou exploitation)

Permet d'examiner en profondeur une zone particulière de l'espace de Recherche.

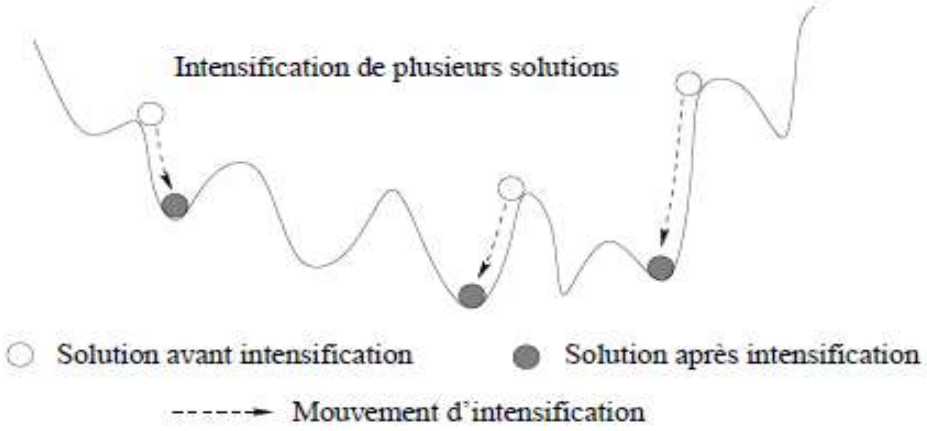


Figure. Processus d'intensification de solutions dans un paysage donné.

❑ Une recherche heuristique efficace

Nécessite un bon compris entre intensification et diversification

- ❖ Métaheuristiques fournissent des moyens différents pour la mise en oeuvre de ces deux principes complémentaires.

DEUX PRINCIPES DE BASE POUR UNE RECHERCHE HEURISTIQUE

❑ Diversification (ou exploration)

permet d'orienter la recherche vers de nouvelles zones (prometteuses) dans l'espace de recherche [J. Hao] .

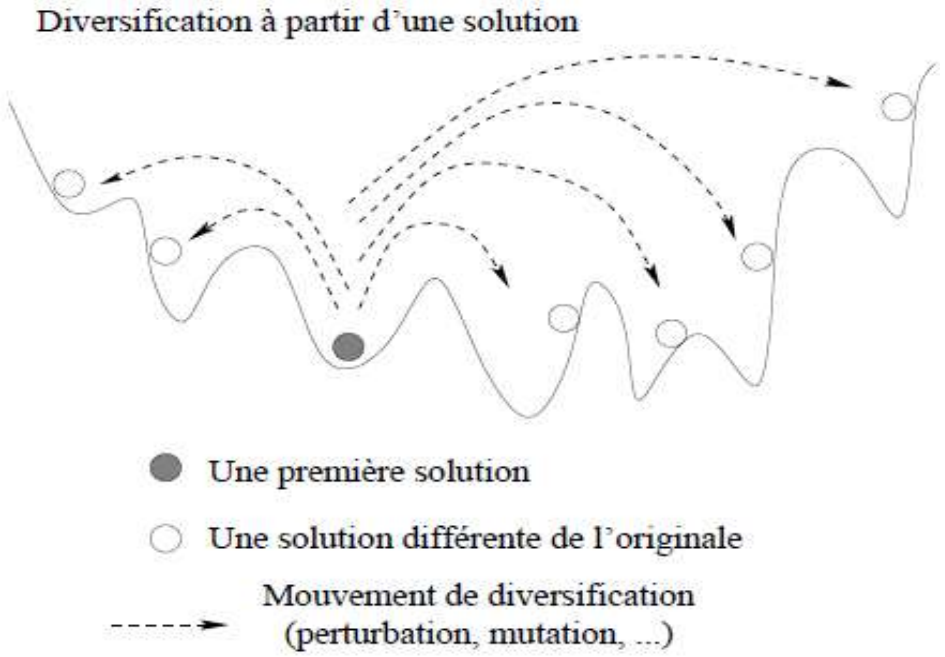
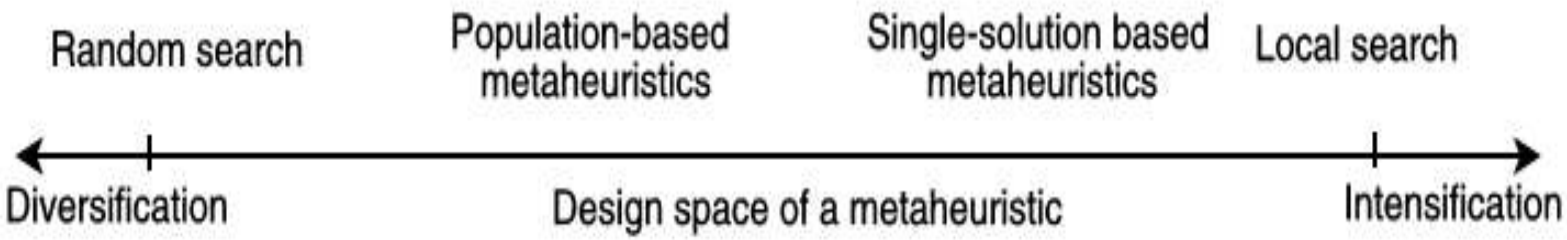


Figure. Processus de diversification d'une solution dans un paysage donné **[J. Bois]**.

DEUX PRINCIPES DE BASE POUR UNE RECHERCHE HEURISTIQUE



- Two conflicting criteria in designing a metaheuristic:
  - **Exploration** (diversification) **versus exploitation** (intensification).
  - In general, basic single-solution based metaheuristics are more exploitation oriented whereas basic population-based metaheuristics are more exploration oriented [Talbi].

RÉFÉRENCES DU CHAPITRE

[J. Bois] Jean-Charles Boisson, Modelisation et resolution par metaheuristiques cooperatives : de l'atome a la sequence proteique, RO. Universite Lille 1, 2008.

[E. Talb] E.G. Talbi, Méthodes d'optimisation avancées, LIFL –CNRS.

[J. Hao] J.K. Hao , Recherche local et mémétique : de la théorie à la pratique, Université d'Angers.

[H. Fior] Humbert Fiorino, Planification Automatique & Techniques d'Intelligence Artificielle, Les fondements de l'informatique et de l'Intelligence Artificielle, LIG.

[L. Lamo] Applications de techniques d'intelligence artificielle aux jeux, *Concepts* avancés pour systèmes intelligents, 2009.

[M. Salo] Michel Salomon, Techniques d'optimisation, les métaheuristiques.

[S. Douir et all] Sidi Mohamed Douiri, Souad Elbernoussi, Halima Lakhbab, Cours des Méthodes de Résolution Exactes Heuristiques et Métaheuristiques, master codes, cryptographie et sécurité de l'information , Rabat.

RÉFÉRENCES DU CHAPITRE

[S. Canu] Stéphane Canu, Introduction à l'optimisation,ASI 4 - Introduction à l'optimisation pour l'ingénieur, 2016.

[Y. Coll] Collette, Patrick Siarry, Optimisation multi-objectif, Livre d'éditions EYROLLE

[I. Bous] Ilhem Boussaid. Perfectionnement de m_etaheuristiques pour l'optimisation continue. Autre. Universit_e Paris-Est, 2013. Fran_cais.

[L. Ali] LEMOUARI Ali, Introduction aux Métaheuristiques, Introduction Aux Métaheuristiques 2014. master codes, cryptographie et sécurité de l'information , Rabat.

[Talbi] METAHEURISTICS, from design to implementation, El-Ghazali Talbi, University of Lille – CNRS, INRIA.

[Wiki] https://fr.wikipedia.org/wiki/