

# Chapitre 6: Optimisation bio-inspirées

## INF889B — Algorithmes d'optimisation combinatoire

Alexandre Blondin Massé

Université du Québec à Montréal

Hiver 2020

# Plan

- 1 Introduction
- 2 Colonies de fourmis
- 3 Algorithmes génétiques
- 4 Optimisation par essaim particulaire

# Introduction

# Exemples

- **Vie**: automates cellulaires
- **Biologie**: algorithmes génétiques, aussi appelés « évolutionnaires » ou « mémétiques », calcul moléculaire
- **Plantes**: système de Lindenmayer
- **Animaux**: colonies de fourmis, essaims d'abeilles,
- **Humains**: intelligence artificielle, réseaux de neurones, systèmes immunitaires artificiels
- **Physique**: recuit simulé, mécanique, gravité, informatique quantique

# Idée générale

## Optimalité

- La nature atteint souvent un état d'**équilibre**
- Cet équilibre est un **optimum local**
- En même temps, la nature a tendance à se **diversifier**
- Ce qui permet de trouver d'autres **optimums**
- Et se rapproche potentiellement d'un **optimum global**

## Règles

- Les systèmes paraissent **complexes**
- Mais sont souvent régis par des **règles simples**
- Potentiellement représentables d'un point de vue **informatique**

# Colonies de fourmis

# Contexte

- Dans les années **40**, Pierre-Paul Grassé s'est intéressé au comportement d'une espèce de **termites**
- Il a remarqué que ces insectes réagissaient à ce qu'il a appelé des « **stimuli significatifs** »
- Qui activent des **réactions** encodées génétiquement
- Il a utilisé le terme « **stigmergie** » pour décrire ce type de communication indirecte

## 2 caractéristiques principales

- L'information est **non symbolique** et ne fait que modifier l'état de l'environnement visité par les insectes
- L'information est **locale** et ne peut être consultée que par les insectes qui visitent son voisinage immédiat

# Phéromone

- Dans le cas des **fourmis**, il a été observé qu'elles ont un comportement similaire
- Leur stratégie consiste à déposer une substance sur leur chemin appelée **phéromone**
- Les autres fourmis peuvent **sentir** la phéromone laissée par les autres fourmis
- Et elles s'y fient afin de **choisir** le meilleur chemin pour aller chercher de la nourriture
- Plus **forte** est la concentration de phéromone, plus **haute** est la probabilité de choisir le chemin

# Expériences

## Expérience de Deneubourg et al.

- Expérience du **pont binaire**
- Deux ponts de **même longueur**
- Au début, comportement plus ou moins **aléatoire**
- Mais après un certain temps, un des ponts a une concentration **plus élevée** de phéromone
- On observe une **convergence** vers l'un des ponts
- Plusieurs expériences menées: chaque point est choisi avec environ **50%** de probabilité

## Expérience de Goss et al.

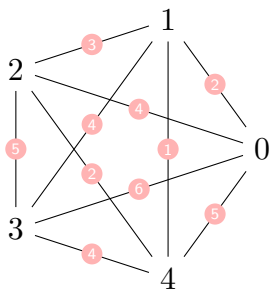
- Avec des ponts de **longueur différente**
- Convergence rapide vers le **plus court** des deux ponts
- Met en évidence l'**évaporation** de phéromone

# Pseudocode

- Le comportement des fourmis a inspiré un **algorithme**
- Différentes **variantes** possibles

```
1: fonction COLONIEDEFOURMIS
2:   Initialisation de la phéromone
3:   tant que la condition d'arrêt n'est pas vérifiée faire
4:     Construire les solutions partielles
5:     Améliorer ou modifier la solution (optionnel)
6:     Mettre à jour la phéromone
```

## Exemple: le voyageur de commerce



- On considère  $m$  **fourmis artificielles**
- Qui se promènent le long des **arêtes** du graphe
- Et qui laissent des traces de **phéromone** sur les **arêtes**
- À chaque sommet, elles vont choisir une arête avec probabilité dépendant de la **concentration** de phéromone

# Construction de la solution

- On associe à chaque **fourmi** une **tournée**
- Au début, on choisit un sommet de départ  $u$  au hasard

On choisit une arête avec probabilité

$$P(u, v) = \frac{\tau(u, v)^\alpha \cdot \eta(u, v)^\beta}{\sum_{w \in N(u)} \tau(u, w)^\alpha \cdot \eta(u, w)^\beta}$$

où

- $\alpha, \beta \geq 0$  sont des paramètres à déterminer
- $\tau(u, v)$  est la valeur de **phéromone** associée à l'arête  $\{u, v\}$
- $\eta(u, v)$  est une heuristique qui pénalise le fait de choisir un voisin qui a déjà été visité, par exemple

$$\eta(u, v) = \begin{cases} 0, & \text{si } v \text{ a déjà été visité;} \\ 1, & \text{sinon.} \end{cases}$$

# Mise à jour des phéromones

Deux éléments à prendre en compte

- **Évaporation**: les phéromones disparaissent au fur et à mesure
- **Dépôt**: les « bonnes » solutions reçoivent plus de nouvelle phéromone que les autres

Mise à jour

$$\tau(u, v) \leftarrow (1 - \rho)\tau(u, v) + \rho \sum_{S \in \mathcal{S}} F(s)$$

où

- $\rho \in (0, 1]$  est le **taux d'évaporation**
- $S$  est l'ensemble des **fourmis** utilisées pour la mise à jour
- $F : \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$  est une **fonction** appelée *fitness function* qui a la propriété que  $f(S) < f(S')$  implique  $F(S) \geq F(S')$  pour tout  $S \neq S'$

# Démonstration

Voir vidéo sur [YouTube](#)

# Algorithmes génétiques

# Introduction

- Aussi appelés **algorithmes évolutionnaires**
- Ou **stratégies évolutionnaires**
- Ou encore **algorithmes mémétiques**

## Théorie de l'évolution

- Inspiré de la théorie de **Darwin**
- On se concentre sur une **population**
- Il y a un mécanisme de **reproduction**
- Par **croisement** ou **recombinaison**
- Entre 2 ou plusieurs **individus**
- Il peut y avoir des **mutations**
- Puis on **sélectionne** les individus les **mieux adaptés**

# Pseudocode

- 1: **fonction** ALGORITHMEGÉNÉTIQUE
  - 2:     Initialiser la population  $P$
  - 3:     **tant que** condition d'arrêt pas encore rencontrée **faire**
  - 4:         Soit  $P'$  obtenue de  $P$  par sélection
  - 5:         Soit  $P''$  obtenue de  $P'$  par reproduction
  - 6:          $P \leftarrow$  combinaison de  $P'$  et  $P''$
- Plusieurs façons d'**initialiser**
  - Plusieurs façons de **sélectionner**
  - Plusieurs façons de faire la **reproduction**
  - Plusieurs façons de **recombinaison**

# Vocabulaire de base

- Un **individu** est une solution admissible
- On accepte parfois de créer des individus **non admissibles**, par exemple par mutation ou par croisement, et des les **corriger**
- Une **population** est un ensemble d'individus
- Une **génération** est une population à une itération donnée
- La **fonction objectif** est la fonction qu'on souhaite optimiser sur tous les individus possibles
- La fonction *fitness* est une mesure de la **qualité** d'un individu (souvent fortement reliée à la fonction objectif)
- L'étape de **reproduction** est l'étape de **croisement** et de **mutation**
- Le ratio du nombre de **nouveaux individus** à une génération subséquente sur le nombre total est appelé **écart générationnel**

# Représentation d'un individu

## Plusieurs représentations possibles

- Une chaîne **binaire** (sac à dos)
- Une chaîne de **caractères**
- Une **permutation** (voyageur de commerce)
- Une suite de valeurs **réelles** (optimisation continue)
- Une structure **arborescente** (arbre de décision)
- Un **réseau de neurones**

## Considérations

- Représentation simple  $\Rightarrow$  opérations efficaces
- Choix dépend fortement de l'opérateur de **croisement**
- Influence l'**exploration** de l'espace de recherche
- Fonction pratique: non valide  $\rightarrow$  valide, qui transforme en un individu **valide** « proche »

# Reproduction

## Mutation

- Opérateur **unaire**
- Transforme un individu en une version **légèrement modifiée**
- Généralement appliquée avec une **faible probabilité**, rarement plus grande que dans 5% des cas

## Recombinaison

- Opérateur  $k$ -aire, pour  $k \geq 2$
- Souvent  $k = 2$
- Parfois, **tout** individu peut être croisé avec n'importe quel individu
- D'autre fois, les croisements sont **contraints**, donc on prépare le croisement en modifiant les « parents »

# Qualité des opérations

## Exploitation

- Exploration de l'environnement **immédiat**
- La qualité de l'opérateur de **croisement** a un énorme impact sur la qualité de l'exploitation
- De façon générale, les résultats sont meilleurs si on applique une **recherche locale** après mutation et après croisement
- Aussi, mieux vaut effectuer les croisements sur les composantes les plus « **prometteuses** » des solutions

## Exploration

- Exploration de l'espace **complet**
- La **mutation** est moins importante et sert surtout à **diversifier** l'espace de recherche exploré

# Exercice: problème du sac à dos

## Rappel

- On a  $n$  objets, indicés de 1 à  $n$
- Chaque objet a une **valeur**  $v(i)$  et un poids  $w(i)$
- Le sac à une capacité maximale  $W$
- On souhaite maximiser la **valeur totale**

## Questions

- Comment représente-t-on un **individu**?
- Proposer un opérateur de **mutation**
- Et un opérateur de **croisement**

# Exercice: problème du voyageur de commerce

## Rappel

- On doit visiter  $n$  villes, **une** seule fois chacune
- La **distance** entre deux villes est  $d(i, j)$
- On souhaite minimiser la **distance totale**

## Questions

- Comment représente-t-on un **individu**?
- Proposer un opérateur de **mutation**
- Et un opérateur de **croisement**

# Exercice: problème de satisfiabilité

- On peut représenter une **formule logique** à l'aide d'un arbre binaire
- On souhaiterait attaquer le problème avec une stratégie **évolutive**

## Questions

- Proposer un opérateur de **mutation**
- Et un opérateur de **croisement**

# Sélection

- **Élitiste:** les meilleurs sont toujours choisis
- **Roulette:** chaque individu a un secteur de roulette de taille variable selon sa *fitness* et est choisi si sa section est tirée
- **Par tournoi:** les individus sont choisis par sous-groupe, après avoir gagné leur tournoi respectif
- **Par rang:** permet d'aplatir la fonction *fitness* et empêche les individus surperformants d'écraser tous les autres
- **Par génération:** on ne conserve aucun individu de l'ancienne génération
- **Contrôlée:** on ne remplace qu'une partie des individus
- **Sélection hiérarchique:** on élimine les individus inintéressants avec une évaluation sommaire, puis on étudie plus extensivement ceux qui restent

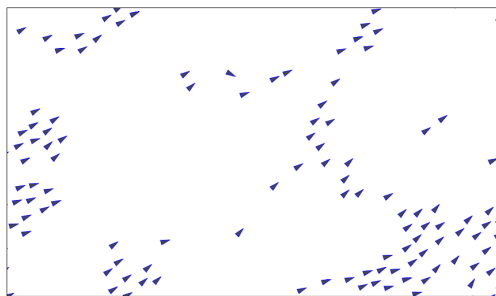
# Optimisation par essaim particulaire

# Inspiration: agrégation (*flocking*)

## 3 règles

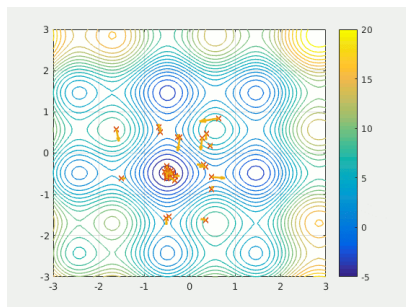
- **Séparation**: répulsion quand il y a une forte proximité
- **Alignement**: direction tend vers celle des voisins
- **Cohésion**: déplacement vers les voisins

Démonstration: Neat Algorithms Flocking



# Idée générale

- En anglais, *particle swarm optimization* (PSO)
- Chaque particule fait une exploration **locale**
- Mais est aussi influencée par les **autres particules**
- L'influence est pondérée par les particules **prometteuses**
- Démonstration: [Wikipedia](#)



# Représentation

- **Fonction objectif:**  $f : \mathbb{R}^d \rightarrow \mathbb{R}$
- **Borne inférieure:**  $x^- \in \mathbb{R}^d$
- **Borne supérieure:**  $x^+ \in \mathbb{R}^d$
- **Particule:**  $p_i$ , pour  $i = 1, 2, \dots, n$
- **Position:**  $x(p_i) \in \mathbb{R}^d$
- **Meilleure position:**  $b(p_i) \in \mathbb{R}^d$
- **Meilleure position globale:**  $b \in \mathbb{R}^d$
- **Vélocité:**  $v(p_i) \in \mathbb{R}^d$

# Pseudocode

```
1: fonction ESSAIMPARTICULAIRE
2:
3:   pour  $i = 1, 2, \dots, n$  faire
4:      $x(p_i) \leftarrow \text{UNIFORME}(x^-, x^+)$ 
5:      $b(p_i) \leftarrow x(p_i)$ 
6:     si  $f(b(p_i)) < f(b)$  alors
7:        $b \leftarrow b(p_i)$ 
8:      $v(p_i) \leftarrow \text{UNIFORME}(-|x^+ - x^-|, |x^+ - x^-|)$ 
9:
10:
11:   tant que condition d'arrêt non vérifiée faire
12:     pour  $i = 1, 2, \dots, n$  faire
13:        $u_\beta \leftarrow \text{UNIFORME}(0, 1)$ 
14:        $u_\gamma \leftarrow \text{UNIFORME}(0, 1)$ 
15:        $v(p_i) \leftarrow \alpha v(p_i) + u_\beta \beta (b(p_i) - x(p_i)) + u_\gamma \gamma (b - x(p_i))$ 
16:        $x(p_i) \leftarrow x(p_i) + v(p_i)$ 
17:       si  $f(x(p_i)) < f(b(p_i))$  alors
18:          $b(p_i) \leftarrow x(p_i)$ 
19:         si  $f(b(p_i)) < f(b)$  alors
20:            $b \leftarrow b(p_i)$ 
21:        $v(p_i) \leftarrow \text{UNIFORME}(-|x^+ - x^-|, |x^+ - x^-|)$ 
```

▷ Initialisation

▷ Simulation

# Observations

## Plusieurs **variantes** possibles

- **Communication limitée** entre les particules
- **Paramètres**  $\alpha$ ,  $\beta$  et  $\gamma$  qui évoluent dans le temps
- Mises à jour différentes selon la **dimension**

## Convergence

- Démontrée sous des **hypothèses** assez fortes
- Pas **démontrée** de façon générale
- Résultats **empiriques** prometteurs

## Généralisation

- Autres espaces **combinatoires**
- En particulier espaces **discrets**
- Ne dépend que d'une distance et des opérateurs arithmétiques  
 $+$ ,  $-$  et  $\times$