Techniques d'optimisation

hepia HES-SO

Guido Bologna

Paul Albuquerque

Michel Vinckenbosch

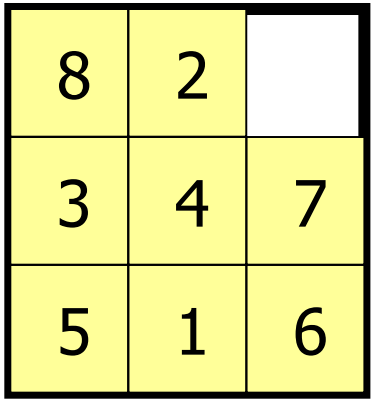
Sujets abordés

- Introduction aux algorithmes de recherche
- Satisfaction de contraintes
- Apprentissage statistique
- Réseaux de neurones artificiels
- Recuit simulé
- Algorithmes génétiques
- Programmation linéaire

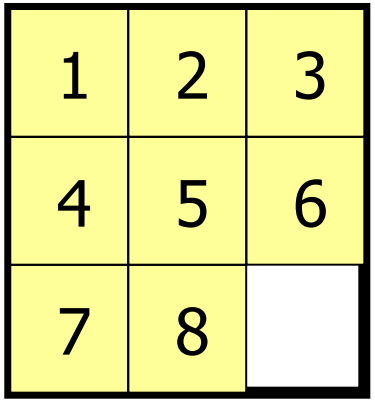
But du cours

- Maîtriser les modèles et algorithmes étudiés de telle sorte à pouvoir déterminer pour un problème d'optimisation donné (défini par exemple par un futur employeur) quel est la technique la plus appropriée à appliquer

Algorithme de recherche : puzzle-8



État initial



État final

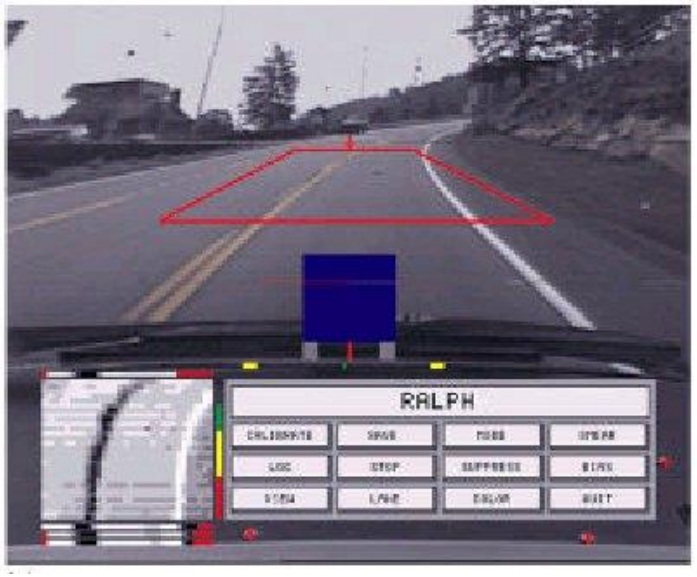
État: n'importe quel arrangement des 8 plaquettes numérotées et de la case vide sur un damier 3x3

Exemple: coloration de carte par propagation des contraintes

- Les solutions sont des affectations satisfaisant toutes les contraintes (couleurs différentes entre les territoires voisins)

Ralph: *un agent de conduite automatique obtenu par apprentissage statistique*

- échantillonne l'image de la route devant le véhicule,
- détermine le rayon de courbure de la route,
- détermine le décalage latéral du véhicule par rapport à la ligne médiane de la route,
- sélectionne l'action de braquage calculée à partir de la courbure de la route et de la position estimée du véhicule sur la route.



fenêtre d'échantillonnage

image basse résolution

Rapidly Adapting Lateral Position Handler

No hands across USA !

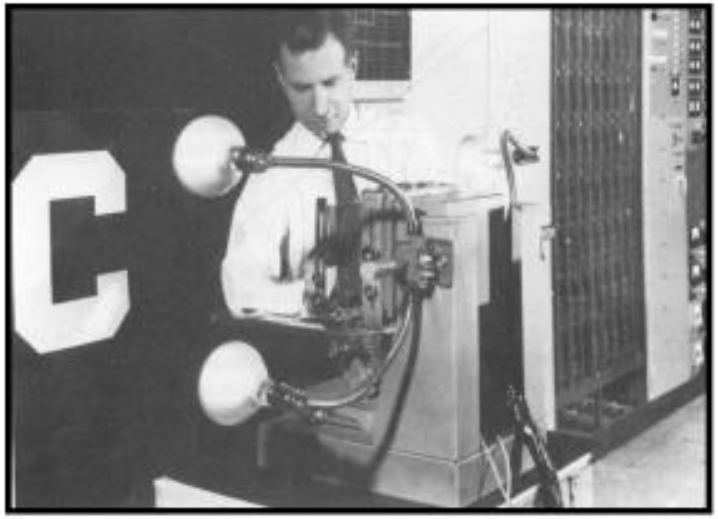
- 2850 miles entre Washington DC et San Diego, sur autoroutes.
- défis du voyage: conduite de nuit, sous la pluie, sur des routes au marquage effacé ou en construction, etc.
- mode d'évaluation: pourcentage de la distance totale parcourue sous le contrôle du système *Ralph*,
- résultat: *Ralph* a conduit le véhicule sur 2796 des 2850 miles du trajet (98.1%):
  - 10 miles de nouvelle route sans marque (ligne centrale, bordures),
  - conduite en ville avec des marquages manquants ou cachés par d'autres véhicules.

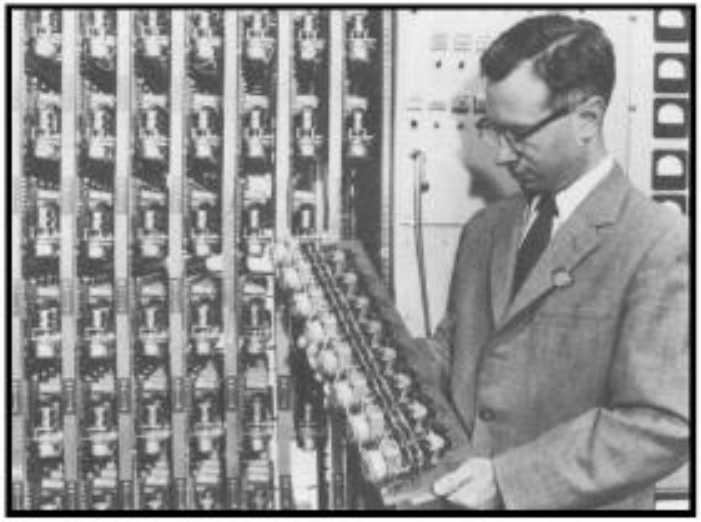
Réseaux de neurones

- Différents types de réseaux de neurones artificiels
  - Perceptron, perceptron multicouche, réseaux de Hopfield, cartes de Kohonen
- Inspirés des neurones biologiques
- Processeurs parallèles et distribués
- Perceptron multicouche
  - apprentissage par rétropropagation des erreurs

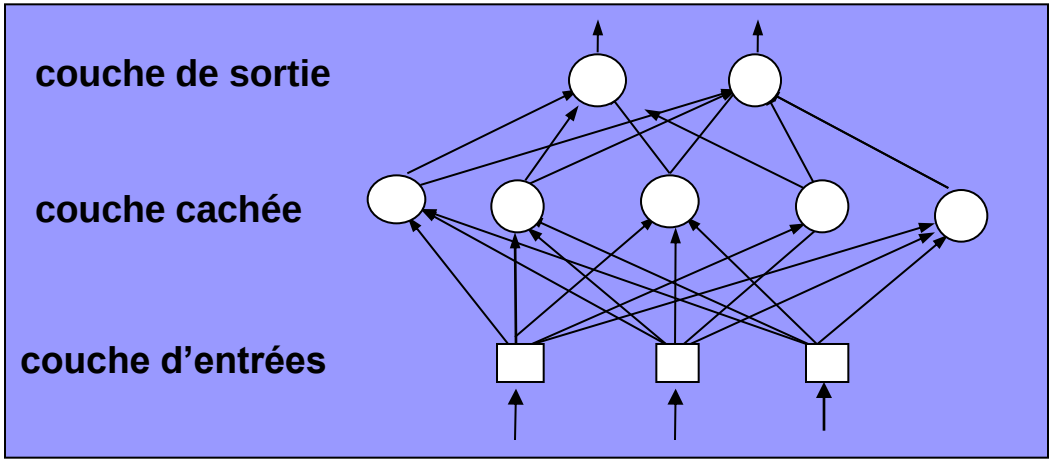
Le Mark I Perceptron

F. Rosenblatt





Réseau multicouche



- Propriétés des réseaux de neurones formels
  - approximation universelle
  - apprentissage
  - généralisation

Réseaux de neurones
Applications

- classification (reconnaissance de caractères)
- approximation de fonction
- traitement du signal, d'images
- commande de processus non linéaires
- prédiction de phénomènes: physiques, économiques
- robotique
- mémoire associative

Formulation et terminologie

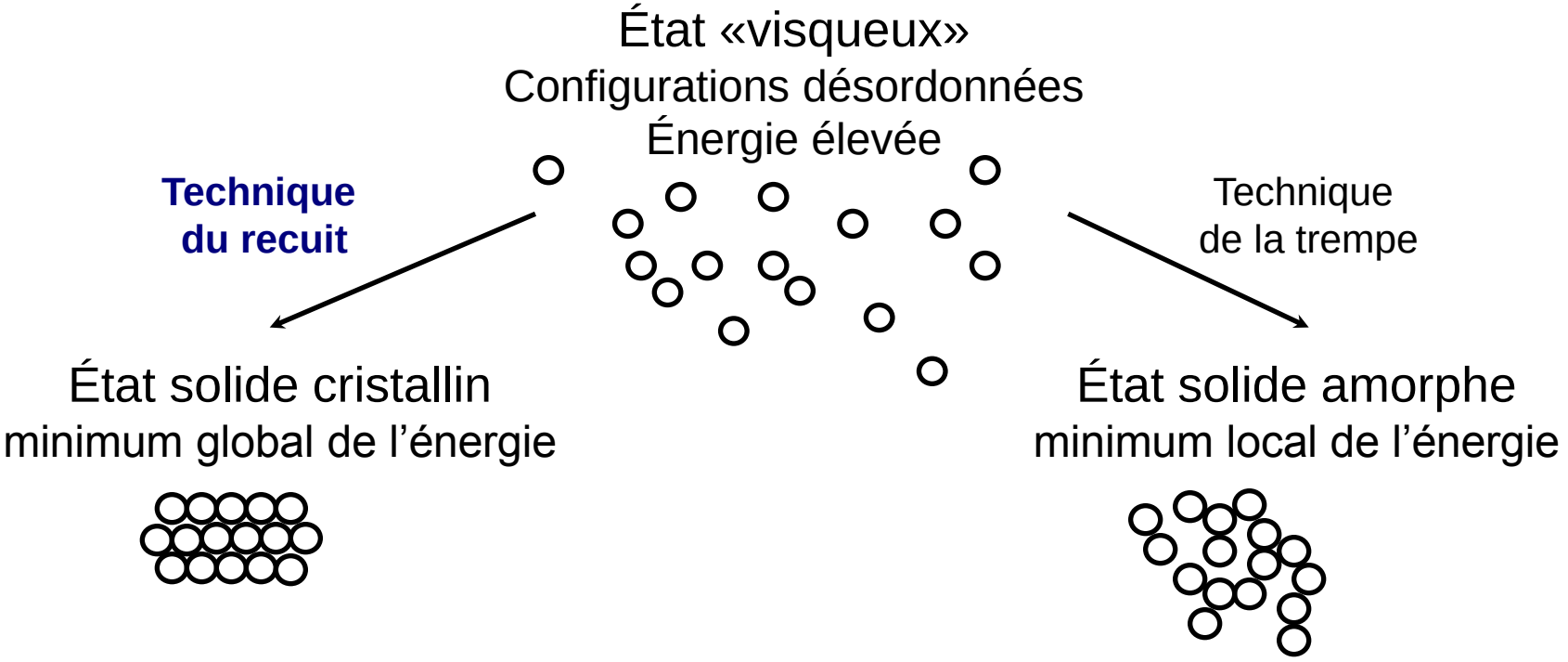
- Description générale d'un problème d'optimisation
  - Ω : **ensemble de configurations** (ou de solutions) du problème
    C : **fonction de coût** (ou objectif)
  - Trouver la configuration $x' \in \Omega$ qui minimise le coût minimal

$$C(x') = \min \{ C(x) \mid x \in \Omega \}$$

  - x' n'est pas obligatoirement unique;
  - la **fonction de coût** est aussi appelée ***fitness function***
  - Ω est aussi appelé l'**espace de recherche**

Le recuit simulé

- Origine du recuit simulé (*simulated annealing*)
  - Recuit métallurgique ou cristallisation d'un liquide



Principe du recuit

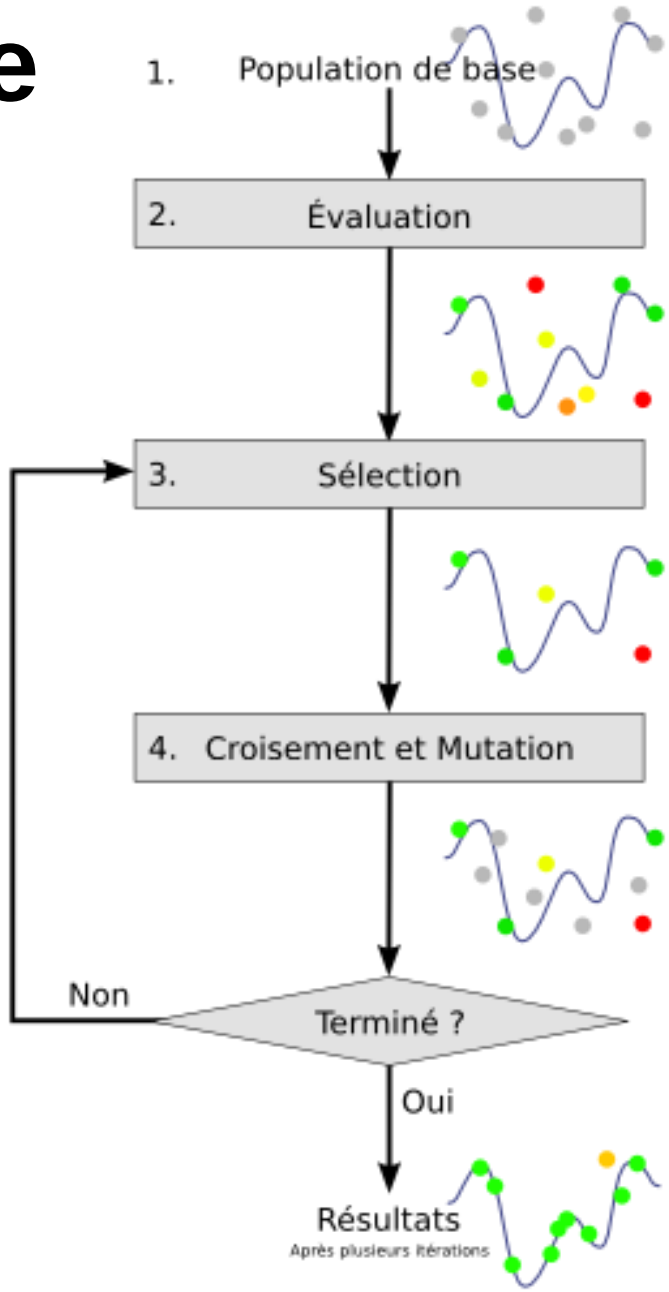
- Initialement le métal est porté à haute température
- Ensuite on le refroidit progressivement:
  - à haute température les atomes sont agités (configurations atomiques équiprobables)
  - à basse température les atomes s'organisent selon une structure atomique parfaite, proche de l'état d'énergie minimale
- Contrainte
  - Le refroidissement doit être lent pour ne pas rester bloqué dans un minimum local
  - un refroidissement trop rapide aboutirait à une configuration sous-optimale

Algorithme génétique (AG)

Principes

- Individu
  - une configuration (solution admissible) du problème à traiter
- Fonction d'évaluation / adaptation (*fitness function*)
  - Fonction objectif à optimiser, permettant d'évaluer l'adaptation d'un individu
- Population
  - ensemble d'individus évoluant simultanément
- Génération
  - itération de la boucle de base: sélection, croisement, mutation

Algorithme génétique

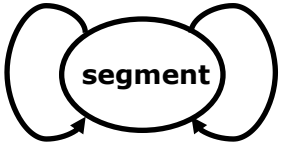


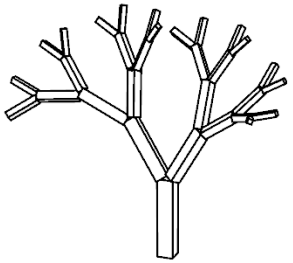
Résoudre le Soduku par AG

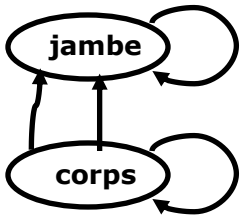
8			4		6			7
						4		
	1					6	5	
5		9		3		7	8	
				7				
	4	8		2		1		3
	5	2					9	
		1						
3			9		2			5

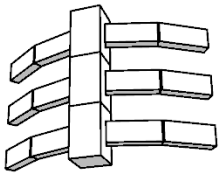
Evolved Virtual Creatures

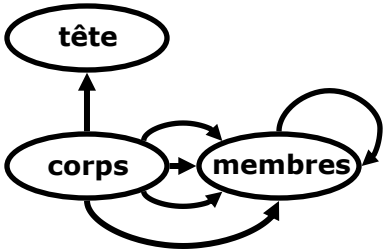
- Chaque « créature » est définie par un graphe
  - Un nœud = un « bloc corporel »
  - Un lien = une articulation
  - Liens récursifs = structures répétitives
- Les nœuds et les liens sont valués
  - Dimensions
  - Limites articulaires
  - position/orientation
  - limites de récursivité
  - contrôle de l ’articulation

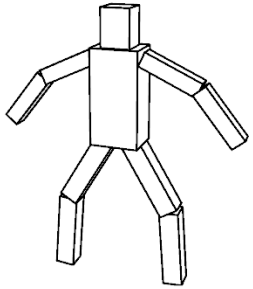








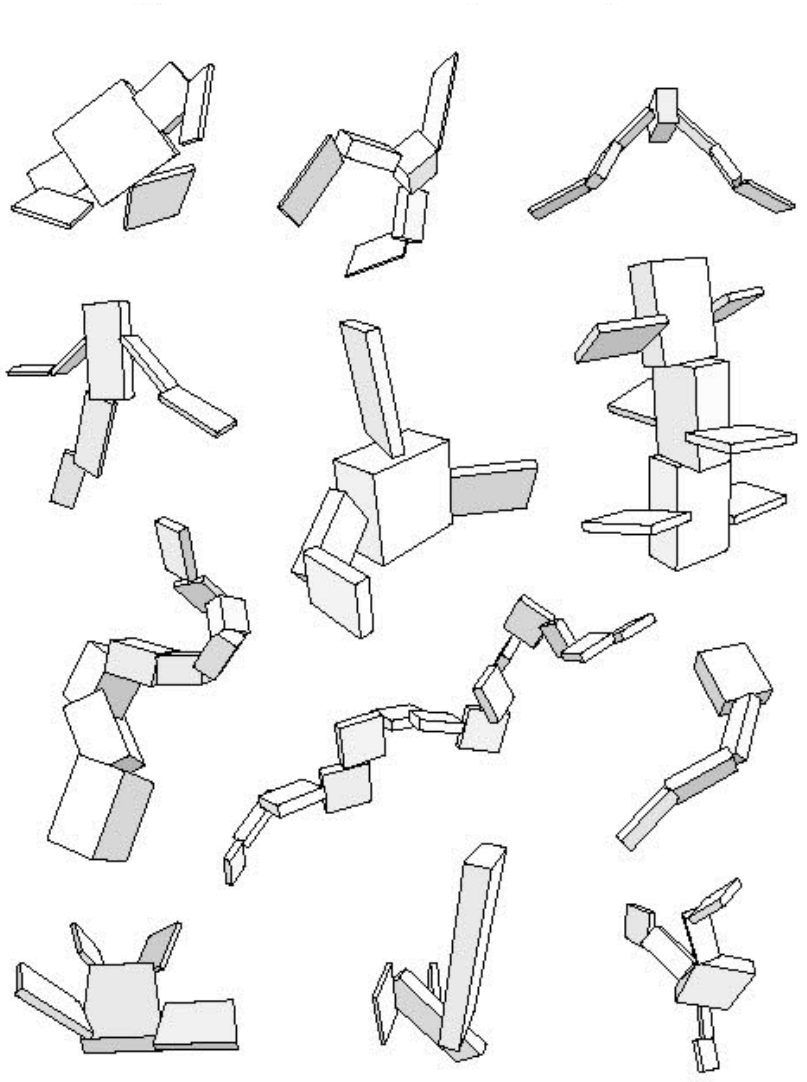




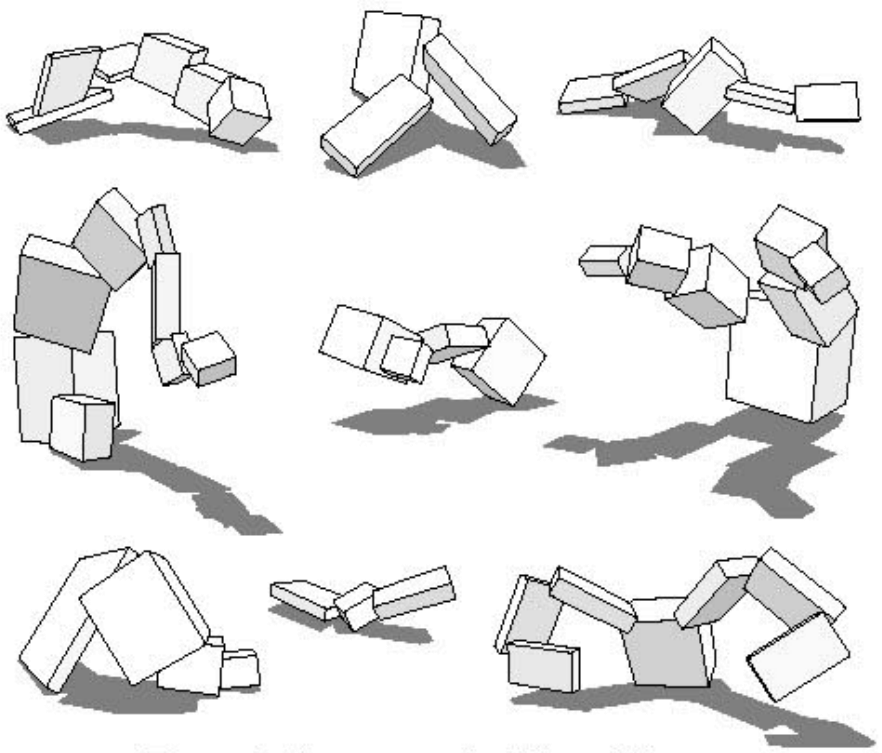
⇘Les régularités structurelles sont « prévues »

Evolving Virtual Swimmers (Karl Sims)

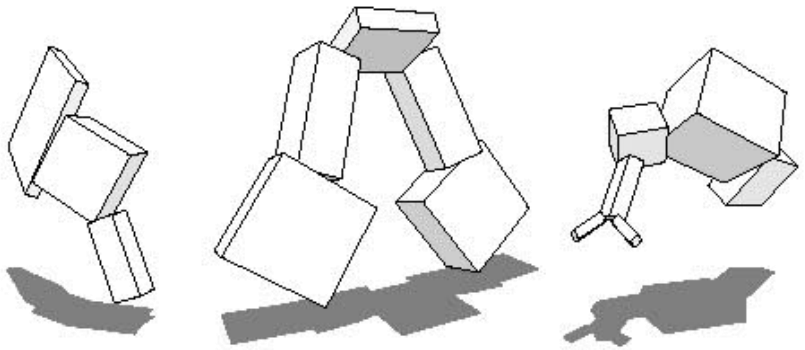
- Environnement :
  - Visqueux sans gravité
- Fitness :
  - Distance parcourue
- Trois types de formes :
  - Rames + gouvernail
  - Nageoires symétriques
  - « serpents marins »



Evolving Virtual Walkers/Jumpers (Karl Sims)



- Environnement :
  - « Grave » sans viscosité
- Fitness :
  - distance parcourue
  - hauteur atteinte
- Formes très variées :
  - déplacement basé sur le saut
  - reptation
  - « début de bipédie »



Programmation linéaire: exemple

- ***H*** hectares de terres, semés de blé et de maïs.
- Quantités totales ***E*** d'engrais et ***I*** d'insecticide.
  - Quantité par hectare: E_1 d'engrais et I_1 d'insecticide pour le blé, E_2 et I_2 pour le maïs
  - prix de vente du blé P_1 ; nombre d'hectares à planter en blé x_1 ; P_2, x_2 pour le maïs
- Nombre optimal d'hectares à planter en blé et en maïs?
  - maximiser $P_1x_1 + P_2x_2$ (maximiser le revenu net)
  - sous contraintes
    - $x_1 + x_2 \leq H$ borne sur le nombre total d'hectares
    - $E_1x_1 + E_2x_2 \leq E$ borne sur la quantité d'engrais
    - $I_1x_1 + I_2x_2 \leq I$ borne sur la quantité d'insecticide
    - $x_1 \geq 0$, $x_2 \geq 0$

Optimisation de systèmes linéaires

- **Programmation linéaire**
  - En mathématiques, les problèmes de **programmation linéaire** (PL) sont des problèmes d'optimisation où la **fonction de coût** et les **contraintes** sont **linéaires**.
  - La PL est un domaine utile de l'optimisation, car les problèmes de PL sont les problèmes d'optimisation les plus faciles.

Optimisation combinatoire
Problème du voyageur de commerce

- Quel est le plus court circuit pour visiter n villes en passant une seule fois par ville ?
- (n-1)!/2 chemins possibles

Nb villes	Nb possibilités	Temps (estimation)
5	12	12 microsecondes
10	181440	0.18 seconde
15	43 milliards	12 heures
20	6 E+15	1928 ans
25	310 E+21	9.8 milliards d'années

Exemple : Optimisation combinatoire ; problème du sac à dos (knapsack)

- Remplir un sac à dos sans excéder sa capacité en maximisant la valeur de son contenu
  - capacité du sac M = 9 (poids max. entier)
  - objets $x_i \in \{0,1\}$ (présent, absent), $i \in [1..n]$
  - poids de x_i : p_i
  - valeur de x_i : v_i

Objet x	v	p
Conserve	15	4
Sucre	10	3
Pain	7	2

Variables : $x_c, x_s, x_p \in \{0,1\}$

$4x_c + 3x_s + 2x_p \leq 9$

$15x_c + 10x_s + 7x_p \geq 30$

Formulation du problème du sac à dos

- Peut-on emporter une valeur totale V supérieure à 30 sans dépasser la capacité
- Trouver une solution (x_1, …, x_n) telle que
  - la fonction V = $\sum_{i=1}^{n} x_i v_i$ est maximale
  - sous contraintes $\sum_{i=1}^{n} x_i p_i \leq 9$

    $\sum_{i=1}^{n} x_i v_i \geq 30$

Heuristiques et métaheuristiques

- Une **heuristique** est une méthode pour résoudre de manière approchée un type de problème particulier, dont la solution optimale exacte ne peut être obtenue (car, par exemple, son obtention nécessiterait un temps de calcul d'ordinateur, de plusieurs milliers d'années)
- Une **métaheuristique** est une méthode, ou plus précisément, un canevas de méthodes, pour résoudre de manière approchée tous les problèmes dont la solution optimale ne peut être obtenue. La méthode ne dépend donc plus du type de problème auquel on est confronté

Références

Métaheuristiques pour l'optimisation difficile
Johann Dréo – Alain Pétrowski
Patrick Siarry – Eric taillard
Eyrolles 2003

Stuart J. Russell, Peter Norvig
Artificial Intelligence : A Modern Approach (3ème édition)
Prentice Hall, 2010

Exigences

Partie pratique : 2 TP notés + 2 TP à rendre (non notés)

Partie théorique : une épreuve écrite

Moyenne = (Note pratique + Note théorique) / 2