Algorithmes de recherche locale (2)

Recherche Opérationnelle et Optimisation
Master 1

Sébastien Verel
verel@lisic.univ-littoral.fr
http://www-lisic.univ-littoral.fr/~verel

Université du Littoral Côte d'Opale
Laboratoire LISIC
Equipe CAMOME

Plan

Optimization

Inputs

- Search space : Set of all feasible solutions,

$$\mathcal{X}$$

- Objective function : Quality criterium

$$f : \mathcal{X} \to \mathbb{R}$$

Goal

Find the best solution according to the criterium

$$x^\star = \operatorname{argmax} f$$

But, sometime, the set of all best solutions, good approximation of the best solution, good 'robust' solution...

Contexte

Black box Scenario

We have only $\{(x_0, f(x_0)), (x_1, f(x_1)), ...\}$ given by an "oracle"
No information is either not available or needed on the definition of objective function

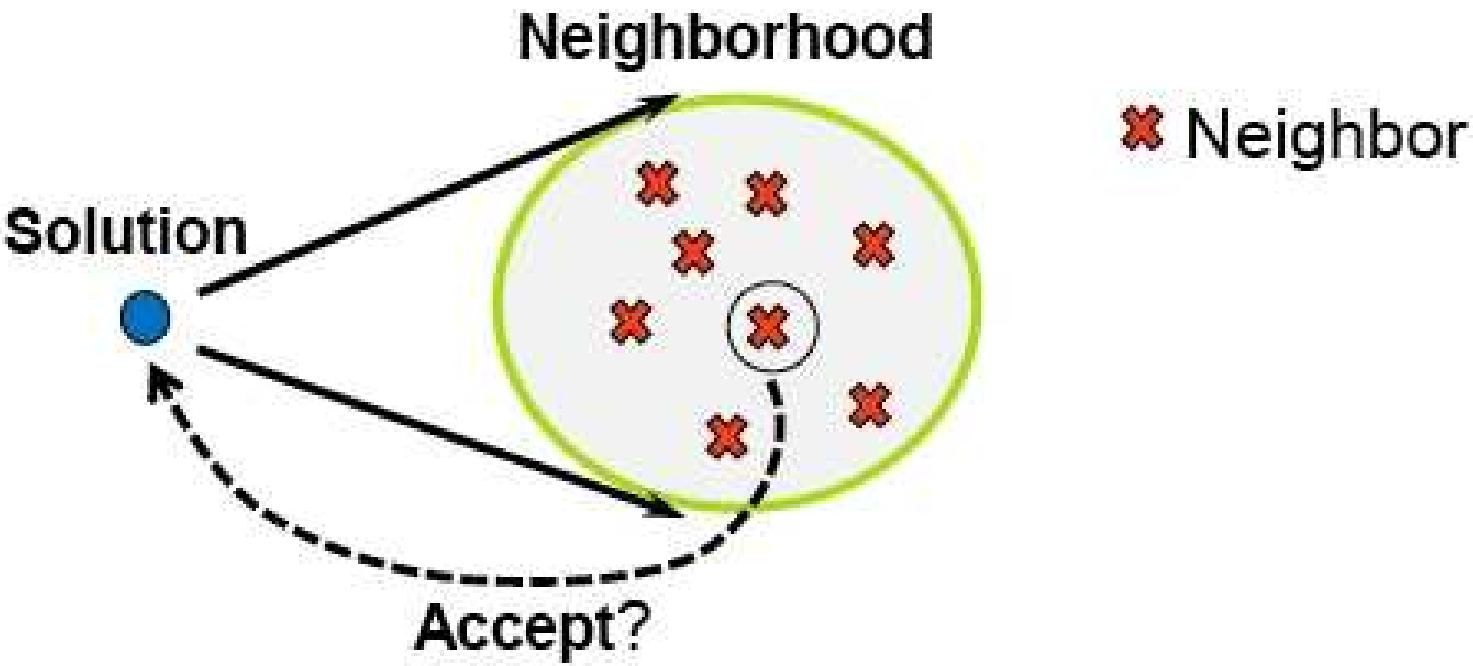
- Objective function given by a computation, or a simulation
- Objective function can be irregular, non differentiable, non continous, etc.

Typologie des problèmes

- Espace de recherche très large dont les variables sont discrètes (cas NP-complet) : optimisation combinatoire
- Espace de recherche dont les variables sont continues : optimisation numérique

Stochatic algorithms with unique solution (Local Search)

- $\mathcal{S}$ set of solutions (search space)
- $f : \mathcal{S} \rightarrow \mathbb{R}$ objective function
- $\mathcal{V}(s)$ set of neighbor's solutions of s

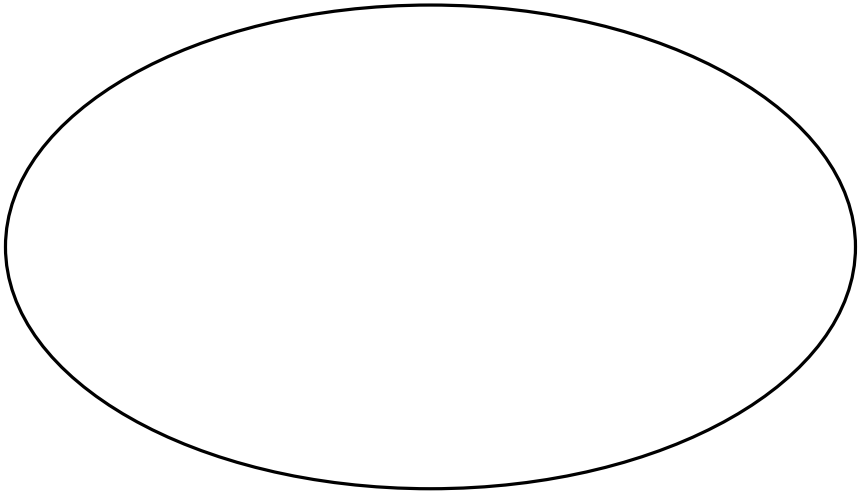


Idée derrière la stratégie locale

Pourquoi une stratégie locale de recherche basé sur un voisinage ?

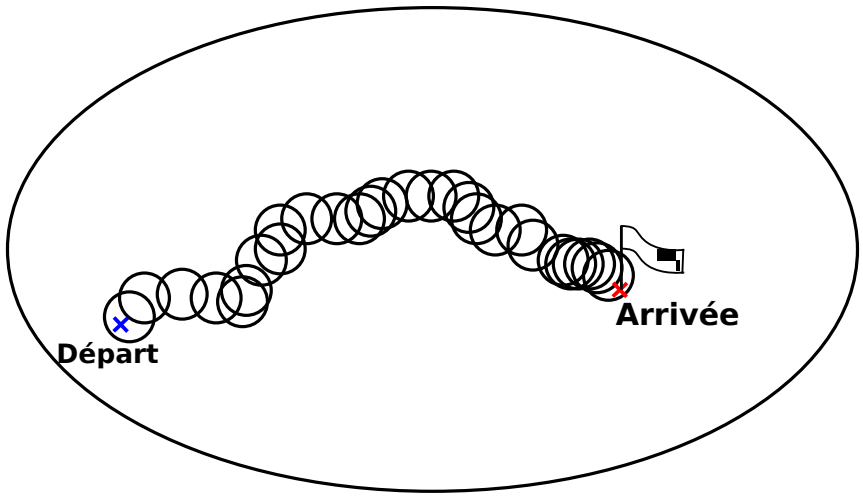
Idée derrière la stratégie locale

Pourquoi une stratégie locale de recherche basé sur un voisinage ?



Idée derrière la stratégie locale

Pourquoi une stratégie locale de recherche basé sur un voisinage ?

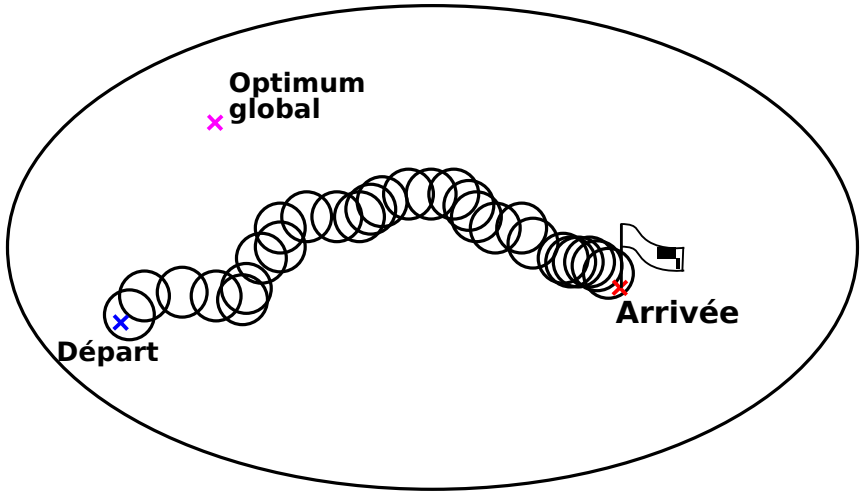


Réduire la résolution du problème global
à une suite de problèmes de petite taille

- Avantage : réduire la complexité

Idée derrière la stratégie locale

Pourquoi une stratégie locale de recherche basé sur un voisinage ?



Réduire la résolution du problème global
à une suite de problèmes de petite taille

- Avantage : réduire la complexité
- Risque : ne pas aboutir à une solution optimale

Concevoir et implémenter une recherche locale

Points critiques dans la conception

- $\mathcal{S}$, codage des solutions
- $f : \mathcal{S} \rightarrow \mathbb{R}$, "bonne" mesure de la qualité des solutions
- $\mathcal{V}$, les voisins doivent être "voisins par rapport à f"
- Sélection du voisin, critère d'acceptation
- Critère d'arrêt (algorithme anytime, non)
- Introduction d'une expertise particulière sur un problème

Points critiques dans l'implémentation

- Choix des structures de données (complexité)
- Evaluation incrémentale (complexité de l'évaluation)
- Générateur aléatoire
- ...

Points critiques

Codage des solutions

- Codage plus ou moins redondant,
- Introduction dans le codage de connaissances au problème,
- Complexité du codage, de l'évaluation

Voisinage

- Taille (nombre de voisins) : $\sharp\mathcal{V}(s)$
- Continuité : Pour tout $s \in \mathcal{S}$ et $s^{'} \in \mathcal{V}(s)$, $Pr(|f(s^{'}) - f(s)| \leq \epsilon)$ est grande
- Probabilité d'amélioration de la solution grande : $Pr(\{s^{'} \in \mathcal{V}(s) : f(s^{'}) > f(s)\})$

Fonction évaluation (fitness)

- Fonction f doit être un guide vers l'optimalité : Plus $f(x)$ est grand, plus x est proche de l'optimum.
- Fonction f ne doit pas être "trompeuse"

Recherche Locale Aléatoire (marche aléatoire)

Heuristique d'**exploration** maximale

Recherche locale aléatoire
Marche aléatoire

Choisir solution initiale $s \in \mathcal{S}$
Evaluer s avec f
repeat
  choisir $s' \in \mathcal{V}(s)$ aléatoirement
  Evaluer s' avec f
  $s \leftarrow s'$
until Nbr d'éval. $\leq$ maxNbEval

- Algorithme inutilisable en pratique
- Algorithme de comparaison
- Opérateur local de base de nombreuses métaheuristiques

Hill-Climber (HC)

Heuristique d'**exploitation** maximale.

Hill Climber (best-improvement)

Choisir solution initiale $s \in \mathcal{S}$
Evaluer s avec f
repeat
  Choisir $s' \in \mathcal{V}(s)$ telle que $f(s')$ est maximale
  if s' strictement meilleur que s **then**
    $s \leftarrow s'$
  end if
until s optimum local

- Algorithme de comparaison
- Opérateur local de base de métaheuristique

Hill-Climber (HC)

Quel est l'inconvénient majeur du Hill-Climbing ?

Optimum local

Optimum local

Etant donné $(\mathcal{S}, f, \mathcal{V})$, f à maximiser.

$x^\star$ est un optimum local ssi pour tout $x \in \mathcal{V}(x^\star)$, $f(x) \leq f(x^\star)$

Optimum local strict

Etant donné $(\mathcal{S}, f, \mathcal{V})$, f à maximiser

$x^\star$ est un optimum local ssi pour tout $x \in \mathcal{V}(x^\star)$, $f(x) < f(x^\star)$

Hill-Climber (HC)

Peut-on imaginer des situations où ce n'est qu'un inconvénient relatif ?

Optimum local

Exercice

- Trouver les maximum locaux (strict et non strit)

Uncontrainst Quadratic Binary Problem (UBQP)

Exercice

Le problème Uncontrainst Quadratic Binary (UBQP) est un problème d'optimisation combinatoire NP-difficile définit par :

$$\forall x \in \{0,1\}^n, \quad f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

- Créer une classe `UBQPEvalFunc` qui dérive de `EvalFunc` qui permet de calculer la fonction f. Les instances définissant la matrice (q_{ij}) sont disponibles sur le site de QAPlib.
- Comparer les performances des recherches locales Hill-Climber best-improvement et first-improvement sur ce problème UBQP.

Metaheuristics

Random search / Hill Climbing

Algorithm 1 Random walk

Choose randomly initial solution $s \in \mathcal{S}$
repeat
  Choose $s^{'} \in \mathcal{V}(s)$ randomly
  $s \leftarrow s^{'}$
until ...

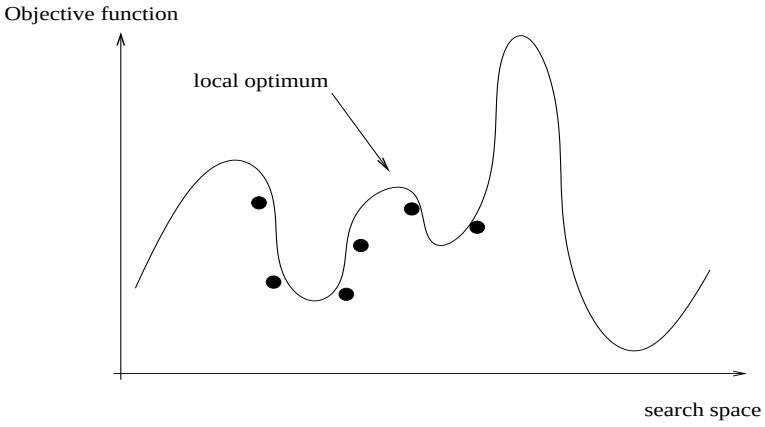
Algorithm 2 Hill-climbing

Choose randomly initial solution $s \in \mathcal{S}$
repeat
  Choose $s^{'} \in \mathcal{V}(s)$ such as $f(s^{'})$ is maximal
  if $f(s^{'})$ is better than $f(s)$ **then**
    $s \leftarrow s^{'}$
  end if
until s local optimum

Metaheuristics

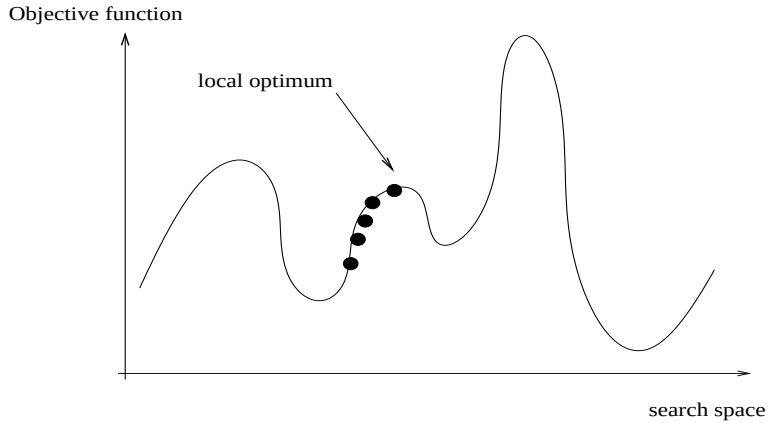
Random search / Hill Climbing

Random walk



maximal exploration ,
diversification

Hill-climbing



maximal exploitation ,
intensification

Compromis Exploration / Exploitation

Escape from local optima, etc.

$\Longrightarrow$ simulated annealing, tabu search, Iterated Local Searc

Recuit Simulé (Simulated Annealing)

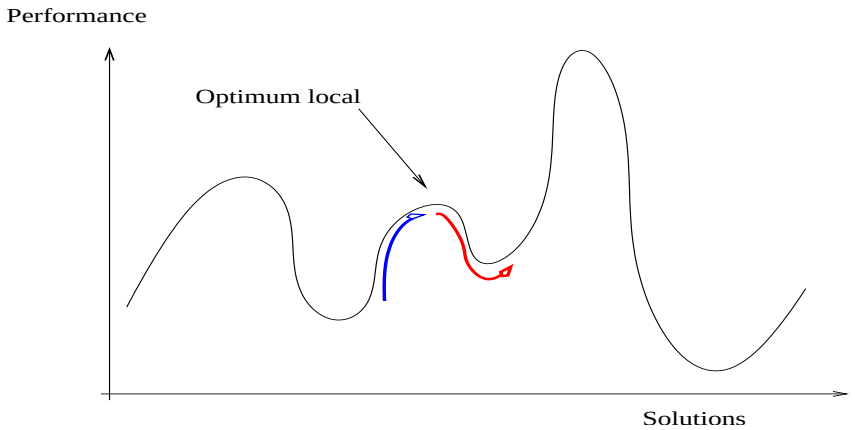
Utilisé depuis les années 80,

- Metropolis (1953) simulation du refroidissement de matériaux (Thermodynamique)
- Kirkpatrick *et al* (IBM 1983) utilisation pour la résolution de problème d'optimisation.

But : échapper aux optima locaux
Principe : probabilité non nulle de sélection d'une solution voisine dégradée

Recuit Simulé : analogie

Système physique	Problème d'optimisation
Energie	fonction objectif
Etats du système	solution
États de basse énergie	bonne solution
Température	paramètre de contrôle



Recuit Simulé

Choisir solution initiale $s \in \mathcal{S}$ et temperature initiale T
repeat
  choisir aléatoirement $s' \in \mathcal{V}(s)$, $\Delta = f(s') - f(s)$
  if $\Delta > 0$ **then**
    $s \leftarrow s'$
  else
    u nombre aléatoire de $[0, 1]$
    if $u < e^{\frac{\Delta}{T}}$ **then**
      $s \leftarrow s'$
    end if
  end if
  update temperature T
until Critère d'arrêt vérifié

Recuit Simulé : remarques

Si $\Delta < 0$ alors la probabilité $\exp(\frac{\Delta}{T})$ est proche de 0 lorsque :

- la différence $|\Delta = f(s') - f(s)|$ est grande
- la temperature est petite

Conséquences :

- lorsque température grande (début de la recherche) :
  $\longrightarrow$ recherche aléatoire
- lorsque température petite (fin de la recherche) :
  $\longrightarrow$ Hill-Climbing

Recuit Simulé : température initiale

Evaluer $\Delta_0 = f(s_0^{'}) - f(s_0)$:

- Choisir n (grand si possible) solutions aléatoires initiales s_0 et une solution voisine $s_0^{'}$
- calculer la moyenne de Δ_0 sur l'échantillon

Température initiale T_0 telle que $\tau_0 = e^{\frac{\Delta_0}{T_0}}$ désiré :
qualité "médiocre" ($\tau_0 = 0.50$) : démarrage à haute température
qualité "bonne" ($\tau_0 = 0.20$) : démarrage à basse température

Recuit Simulé : décroissance de “température”

décroissance suivant une loi géométrique $T_{k+1} = \alpha T_k$
souvent $0.8 \leq \alpha < 1.0$

Changement par pallier de température suivant l'une des deux conditions :

- $12.N$ perturbations acceptées (mouvements de solution)
- $100.N$ perturbations tentées (mouvement ou non mouvement)

où N est un paramètre qui décrit la taille du problème (nombre de villes, de variables...)

Recuit Simulé : Critère d'arrêt

Arrêt aprés 3 palliers successifs sans aucune acceptation.

Recuit Simulé : Remarques

- Toutes ces indications ne sont pas universelles :
  L'analyse du problème et l'expérience de concepteur permettent de les adapter
- Vérifier votre générateur aléatoire
- La qualité du résultat doit dépendre "peu" de l'exécution de l'algorithme

Premières Applications : dans le placement de circuits électroniques

Recuit Simulé : Bibliographie

E. Aarts, J. Korst : "Simulated Annealing and Boltzmann machine" John Wiley, New-York 1989
P. Siarry : "La méthode du recuit simulé : théorie et application" ESPCI - IDSET , 10 rue Vauquelin, Paris 1989

Recherche Tabou (Tabu Search)

Introduite par Glover en 1986 :
"Future paths for Integer Programming and Links to Artificial Intelligence", Computers and Operations Research, 5 :533-549, 1986.

But : échapper aux optima locaux
Principe : Introduction d'une notion de mémoire dans la stratégie d'exploration

Interdiction de reprendre des solutions déjà (ou récemment) rencontrées

Recherche Tabou (Tabu Search)

Choisir solution initiale $s \in \mathcal{S}$
Initialiser Tabou M
repeat
  choisir $s' \in \mathcal{V}(s)$ telle que :
  ($f(s')$ meilleure solution de $\mathcal{V}(s)$ ET Critère d'aspiration vérifié)
  OU $f(s')$ meilleure solution de $\mathcal{V}(s)$ non taboue
  $s \leftarrow s'$
  update Tabou M
until Critère d'arrêt vérifié

Recherche Tabou : mémoire des tabous

Les tabous sont souvent des mouvements tabous pendant une durée

exemple : problème maxsat avec $n = 6$
$M = (0, 3, 0, 0, 0, 0)$
le deuxième bit ne peut être modifié pendant 3 itérations.
$M = (1, 2, 0, 0, 2, 5)$
seuls bits non tabou 3 et 4

Lorsqu'un mouvement est effectué :
interdiction pendant n itérations

Exercice Tabou

Exercice

Exécuter un Tabou sur un problème MAX-SAT.

Recherche Tabou : mémoire des tabous

Lorsqu'un mouvement est effectué :
interdiction pendant n itérations
Si n trop faible, tabou peu efficace
Si n trop grand, les solutions sont "à flanc de coteau".

$\longrightarrow$ Stratégie de diversification

Recherche Tabou : Mémoire à long terme

Statistique sur les mouvements :
Repérer les mouvements trop utilisés (difficulté de recherche, optimum local...)

Fréquence $freq(m)$ d'utilisation d'un mouvement m :
pénalisation du mouvement m par ajout d'interdiction en fonction de $freq(m)$.

Recherche Tabou : Critère d'aspiration

Enlever le caractère tabou d'une solution :

Lorsque la solution est la meilleure jamais rencontrée

Recherche Tabou : Bibliographie

Glover *et al* : "Tabu Search" Kluwer Academic Publishers, 1997

Iterated Local Search

Principe

Une fois la solution courante dans un optimum local,
Perturbation (grande modification) de la solution courante
pour initier une nouvelle recherche locale à partir de celle-ci.

Iterated Local Search (ILS)

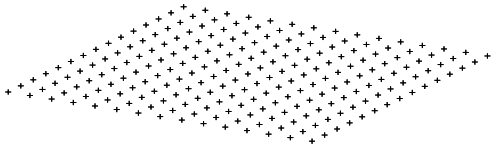
Algorithme

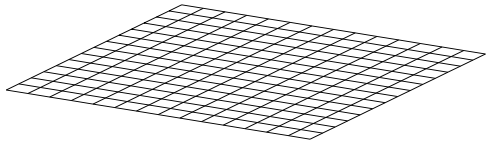
Choisir solution initiale $s \in \mathcal{S}$
$s \leftarrow$ localSearch(s)
repeat
  $s' \leftarrow perturbation(s)$
  $s' \leftarrow$ localSearch(s')
  Si accept(s, s') Alors
    $s \leftarrow s'$
  FinSi
until Critère d'arrêt vérifié

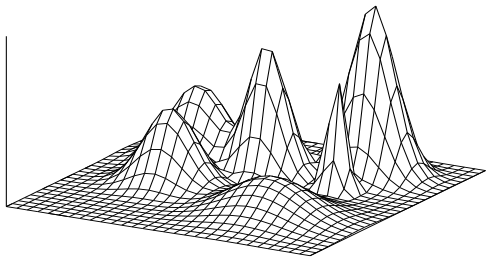
Travaux pratiques

Comparer les performances du recuit simulé et l'Iterated local search sur le problème knapsack

Paysage de fitness



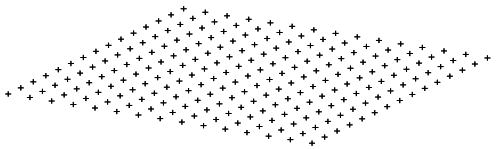


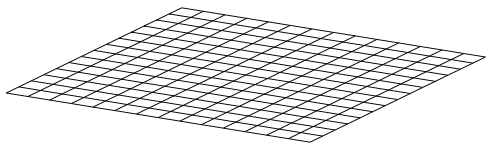


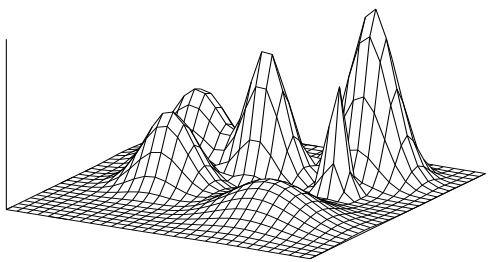
Origine Biologique (Wright 1930) : Modélisation évolution des espèces

Utiliser pour modéliser des systèmes dynamiques : physique statistique, évolution moléculaire, écologie, etc

Optimisation combinatoire







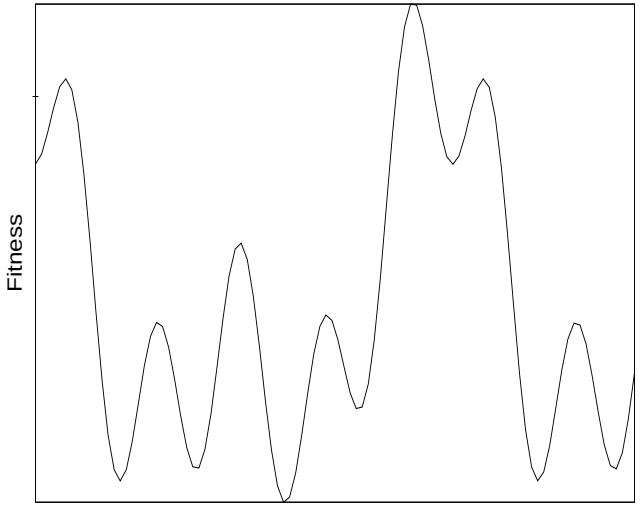
Paysage de fitness $(\mathcal{S}, \mathcal{V}, f)$:

- $\mathcal{S}$: ensemble de solutions potentielles,
- $\mathcal{V} : \mathcal{S} \rightarrow 2^{\mathcal{S}}$: relation de voisinage,
  - $\mathcal{V}(x) = \{y \mid y = op(x)\}$
  - $\mathcal{V}(x) = \{y \mid d(y, x) \leq 1\}$
- $f : \mathcal{S} \rightarrow \mathbb{R}$: fonction à optimiser.

Intérêts du concept

- Relation entre description géométrique d'un problème et dynamique de recherche
- Pertinence du choix de l'opérateur
- Connaissance de la géométrie du problème
  $\Rightarrow$ conception de métaheuristiques adaptées

Paysage Multimodal



paysage multimodal

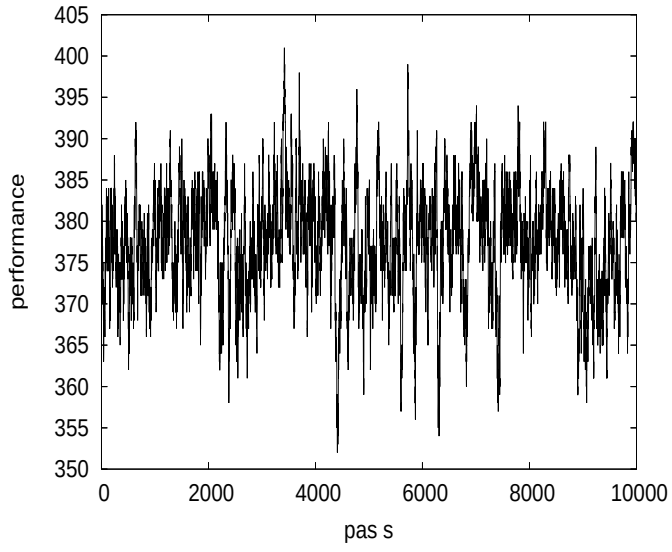
Optimum local : aucune solution voisine de meilleure performance.

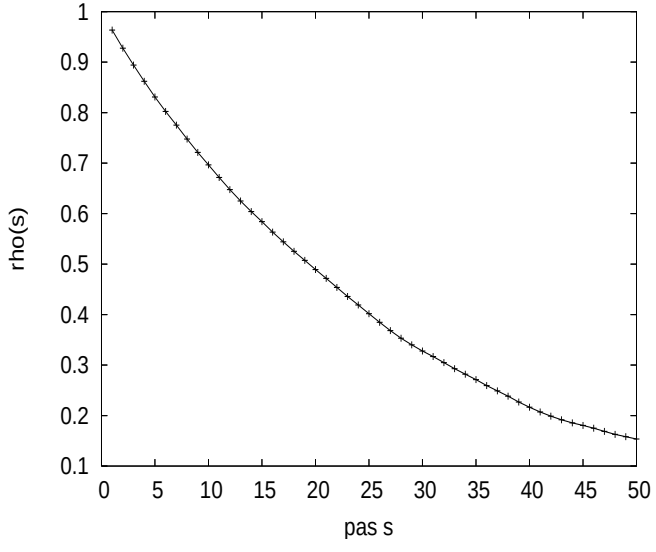
- Difficulté liée au nombre
- Taille des bassins d'attraction

Estimation : Marche adaptative $(s_0, s_1, \ldots)$ où $s_{i+1} \in \mathcal{V}(s_i)$ $f(s_i) < f(s_{i+1})$

- Terminaison sur optimum local
- Longueur : indice de distance inter-optima

Paysage Rugueux

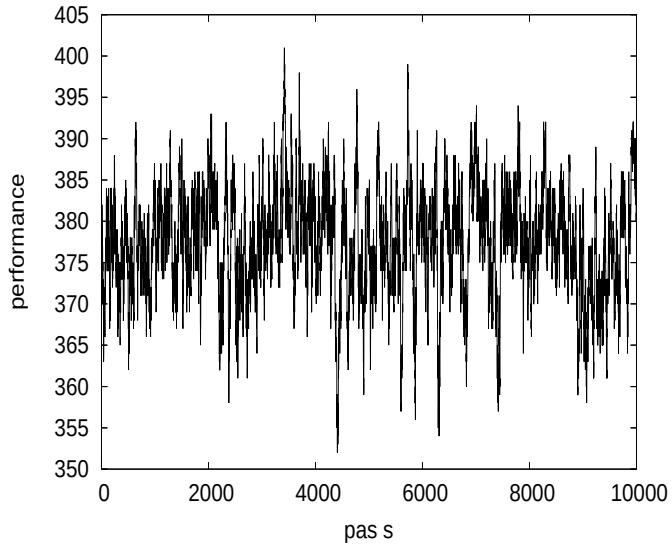


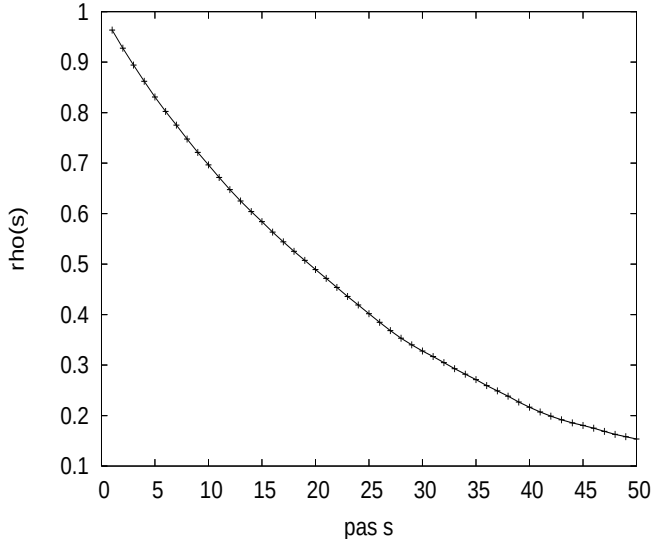


Autocorrélation lors d'une marche aléatoire (Weinberger 1996)
Longueur de corrélation $\tau = \frac{1}{\rho(1)}$

- τ petit : paysage rugueux
- τ grand : paysage lisse

Paysage Rugueux





Autocorrélation lors d'une marche aléatoire (Weinberger 1996)
Longueur de corrélation $\tau = \frac{1}{\rho(1)}$

- τ petit : paysage rugueux
- τ grand : paysage lisse

conjecture
(Stadler 92, Garcia 97) :

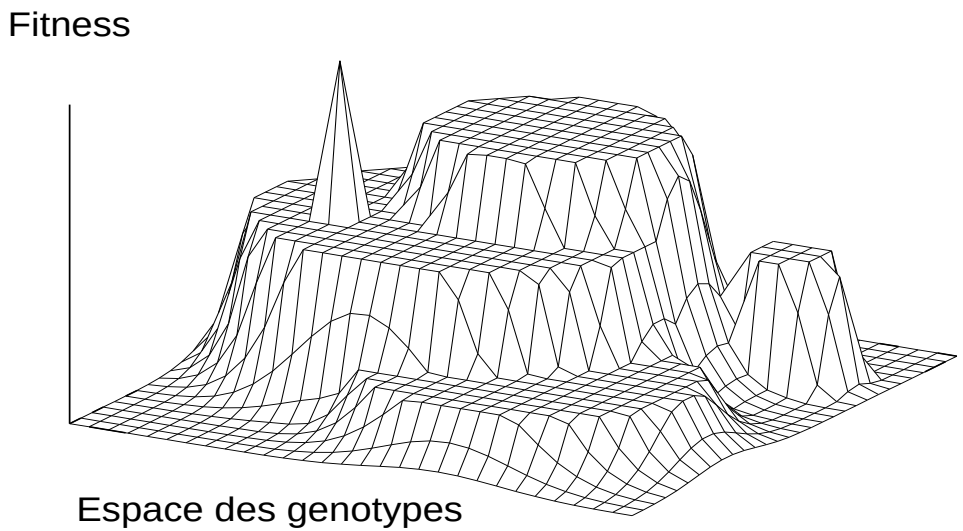
$$M \approx |\mathcal{S}|/|\mathcal{B}(x, \tau)|$$

Paysage Neutre

Théorie de la neutralité (Kimura $\approx$ 1960)

Théorie de la mutation et de la dérive aléatoire

Rôle prépondérant des mutations sans influence sur la performance

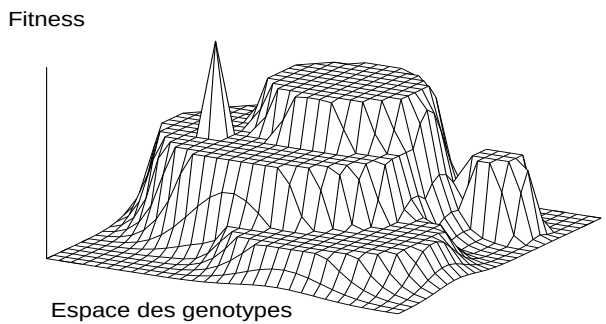


- Géométrie de plateaux
- Degré de neutralité
- Réseaux de neutralité (Schuster 1994, structure secondaire de l'ARN)

Paysage Neutre

Évolution artificielle

Prise en compte depuis les années 80 en évolution artificielle : redondance (Goldberg 87)



Présence dans :

- Programmation génétique
- Contrôleur de robot
- Conception de circuit (Cartesian GP)
- Étiquetage de graphe (MinLA)

Paysage neutre
Optimisation combinatoire

Plusieurs possibilités :

- Diminuer la neutralité :
  conjecture : redondance nuit aux performances
- Utiliser une métaheuristique adaptée :
  conjecture : neutralité est intrisèque
- Augmenter la neutralité par un choix de codage redondant :
  conjecture : éviter les optima locaux

Paysage neutre
Optimisation combinatoire

Plusieurs possibilités :

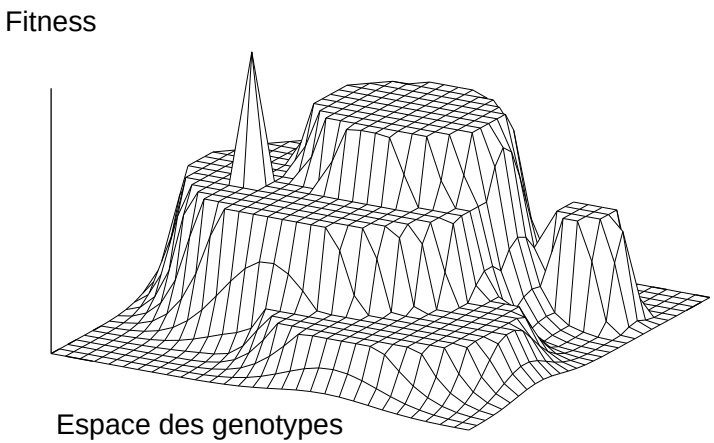
- Diminuer la neutralité :
  conjecture : redondance nuit aux performances
- Utiliser une métaheuristique adaptée :
  conjecture : neutralité est intrisèque
- Augmenter la neutralité par un choix de codage redondant :
  conjecture : éviter les optima locaux

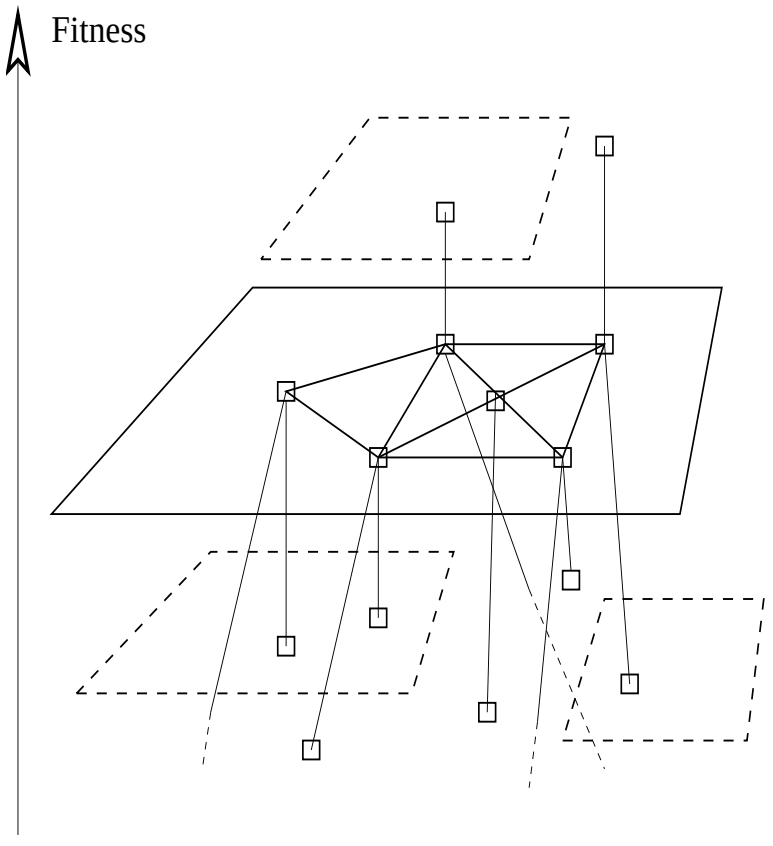
Meilleure description et connaissance des paysages neutres

$\longrightarrow$ Concevoir de nouvelles métaheuristiques

$\longrightarrow$ Évaluer la pertinence d'un codage

Réseaux de neutralité (Schuster 1994)

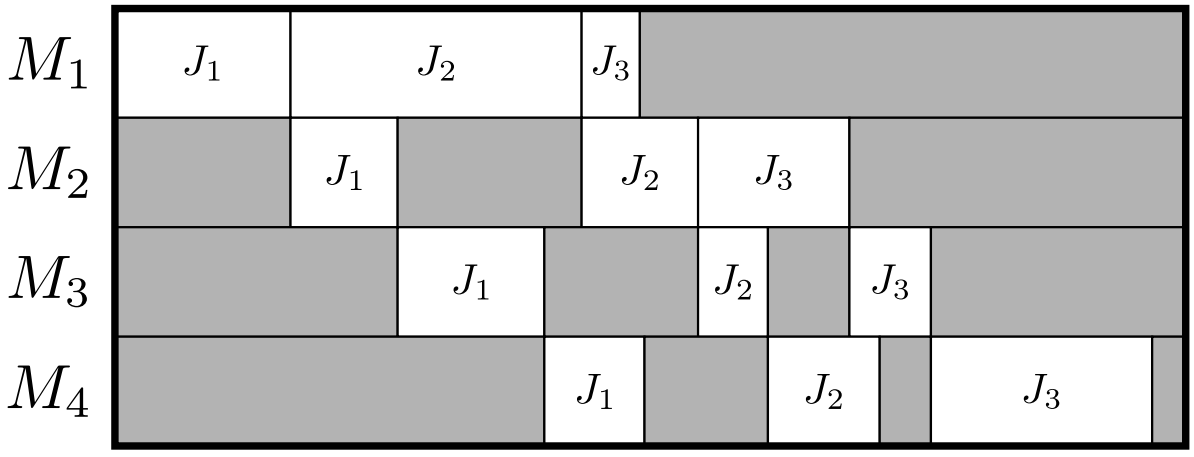




From fitness landscapes to design

Example of Flow Shop Scheduling problem.

M.-E. Marmion, L. Jourdan, C. Daehnens, A. Liefooghe, S. Verel

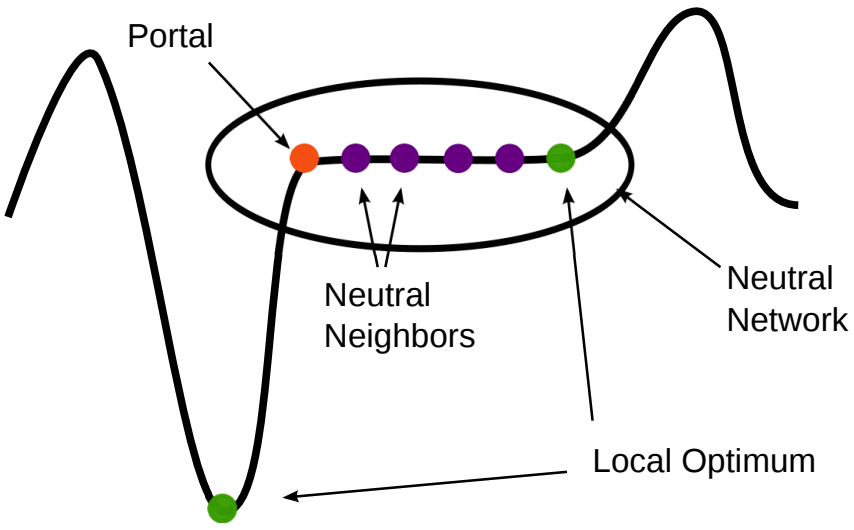


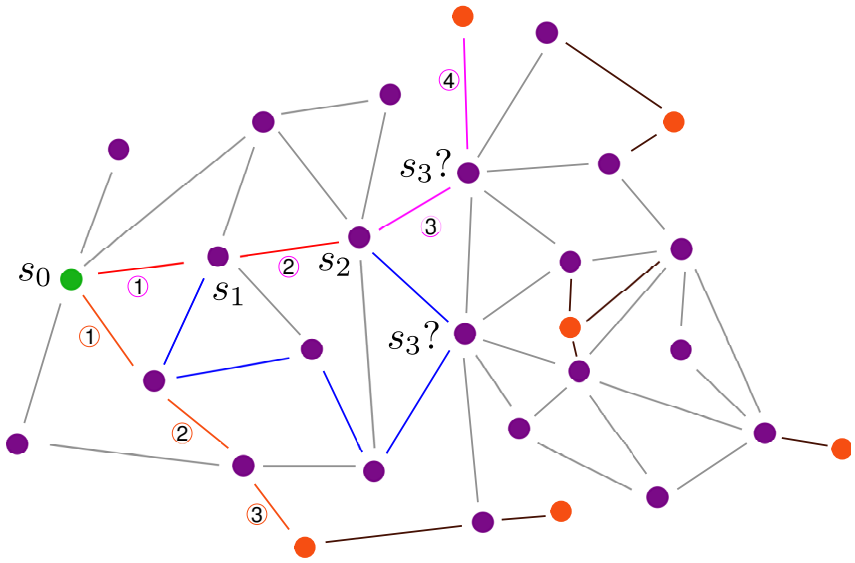
- N jobs, M machines
- Processing time can be different on each machine
- Solution representation = Permutation
- Minimization of the makespan

From fitness landscapes to design
[LION'11]

Analysis / Questions

- Is there some neutrality and plateaus ?
- Is it large plateaus ?
- Can we escape from plateaus ?





From fitness landscapes ...

Analysis / Questions

- Is there some neutrality and plateaus ?

Average neutral degree :

M / N	20	50	100	200
5	87 (24%)	720 (30%)	3038 (31%)	
10	32 (9%)	336 (14%)	1666 (17%)	7920 (20%)
20	14 (4%)	168 (7%)	882 (9%)	3960 (10%)

From fitness landscapes ...

Analysis / Questions

- Is there some neutrality and plateaus ?

Average neutral degree :

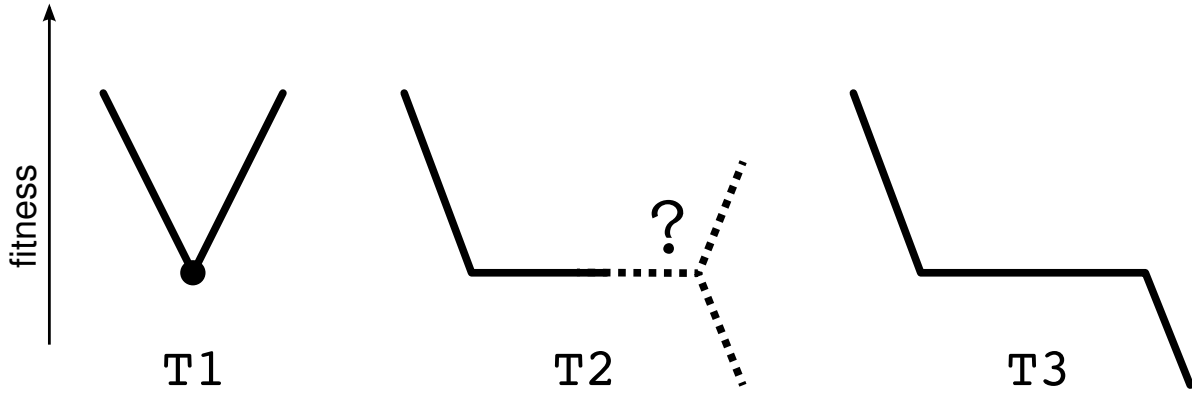
M / N	20	50	100	200
5	87 (24%)	720 (30%)	3038 (31%)	
10	32 (9%)	336 (14%)	1666 (17%)	7920 (20%)
20	14 (4%)	168 (7%)	882 (9%)	3960 (10%)

YES

From fitness landscapes ...

Analysis / Questions

- Is it large plateaus ?

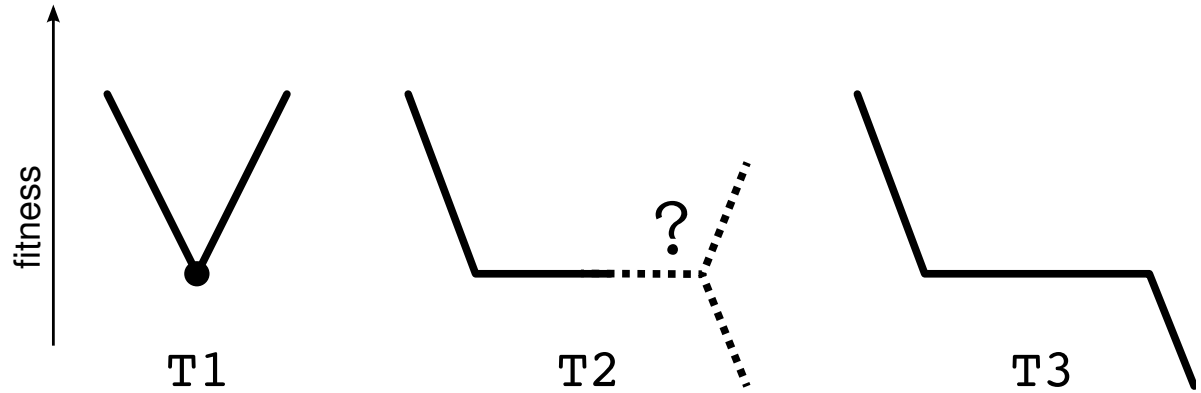


- No T_1 for $N = 50, 100, 200$
- 0 - 20% T_1 T_2 for $N = 20$
- $> 97\%$ of T_3

From fitness landscapes ...

Analysis / Questions

- Is it large plateaus ?



- No T_1 for $N = 50, 100, 200$
- 0 - 20% T_1 T_2 for $N = 20$
- $> 97\%$ of T_3

YES

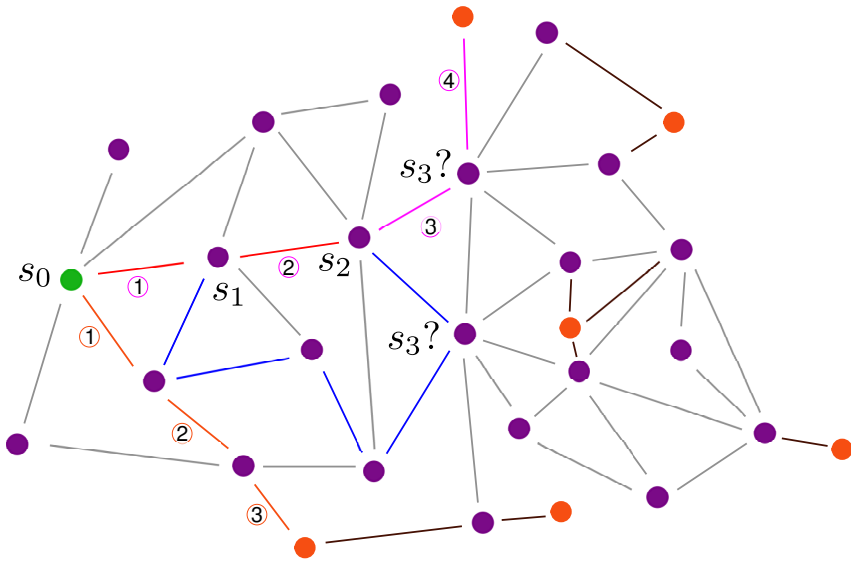
From fitness landscapes ...

Analysis / Questions

- Can we escape from plateaus ?

Average number of steps to find a portal :

M / N	20	50	100	200
5	17	33	34	
10	10	14	17	30
20	6	6	6	6



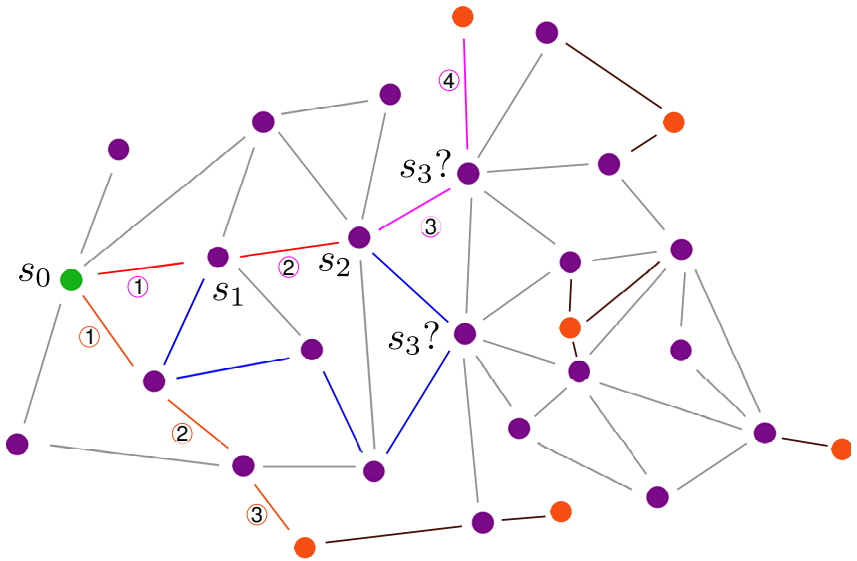
From fitness landscapes ...

Analysis / Questions

- Can we escape from plateaus ?

Average number of steps to find a portal :

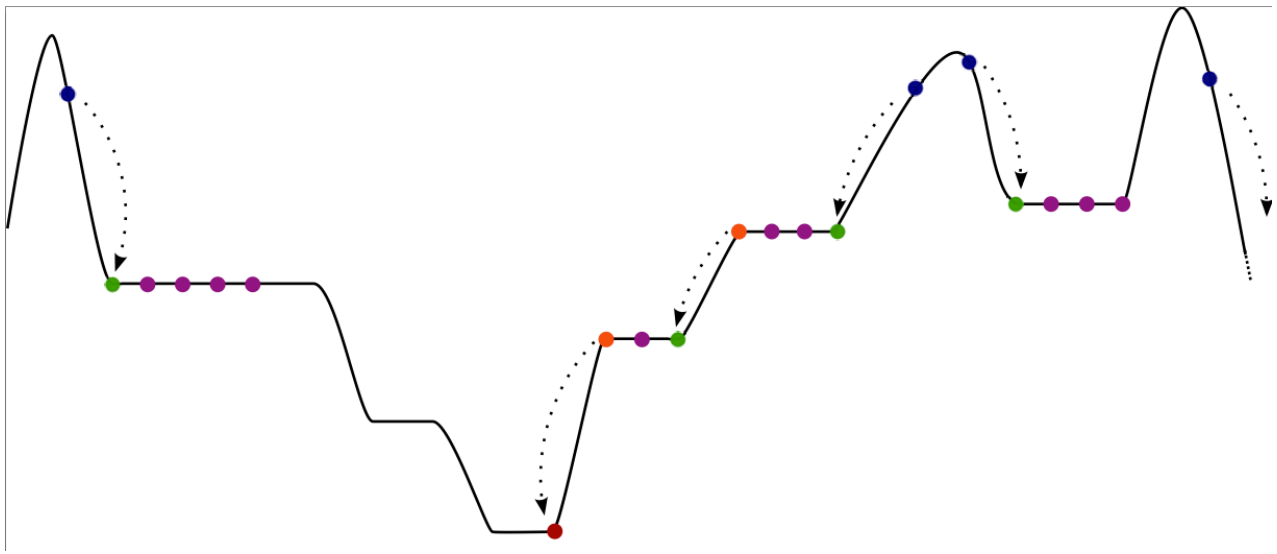
M / N	20	50	100	200
5	17	33	34	
10	10	14	17	30
20	6	6	6	6



YES

From fitness landscapes to design of LS

Neutrality based Iterated Local Search (NILS) [EVOCOP'11]



NILS principle

- Local Search :
  - First-improvement Hill-Climbing
- Perturbation :
  - Neutral moves until portal or maximum number of steps
  - Kick move when no improvement

From fitness landscapes to design of LS

Neutrality based Iterated Local Search (NILS)

- Efficient local search compare to previous ones
- Much simpler one, and performances are well understood
- One new best known solution on structured a real-like instance
- The methodology can be applied to others combinatorial problems

Conclusion

- Problèmes d'optimisation combinatoire fréquents dans l'industrie (et fondamentaux en informatique théorique)
- Métaheuristiques de recherche locale :
  $\longrightarrow$ Recherche aléatoire
  $\longrightarrow$ Hill-climbing Best-improvment ou first-ascent
  $\longrightarrow$ Recuit simulé
  $\longrightarrow$ Recherche tabou
  $\longrightarrow$ Iterated Local Search
- Paysage de Fitness : métaphore qui permet
  $\longrightarrow$ l'analyse du lien entre métaheurisque et problème
  $\longrightarrow$ imaginer de nouvelles metaheurisques (par ex. NILS)
- Propriété principale :
  $\longrightarrow$ rugosité : optima locaux, structure de corrélation
  $\longrightarrow$ neutralité : réseau de neutralité

Travaux pratiques

Comparer la longueur des marches adaptatives sur le problème knapsack selon le type de fonction de pénalité (linéaire et quadratique par exemple)