Optimisation Stochastique

Monte-Carlo, Recuit Simulé, Tabou, Optimisation par Essaims Particulaires, Fourmis

Pr Pierre Collet

Laboratoire ICUBE

Campus Numérique des Systèmes Complexes

Pierre.Collet@unistra.fr

Méthode de Monte-Carlo

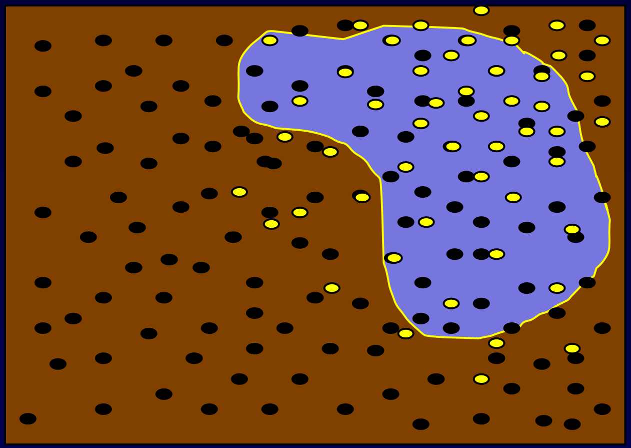
- Méthode née avec les ordinateurs, inventée par Fermi, Ulam, von Neumann, … Metropolis et utilisée sur ENIAC, MANIAC à partir de 1947 pour prédire (sans les résoudre) le résultat d'équations différentielles.
- Naissance des mathématiques expérimentales grâce aux ordinateurs.
- L'idée est d'utiliser un grand nombre d'essais aléatoires pour trouver une solution approchée à un problème donné.
- Le degré zéro d'une approche Monte-Carlo est la recherche aléatoire (on essaie des millions de solutions au hasard et on garde la meilleure).

N. Metropolis and S. Ulam, « The Monte Carlo method », Journal of the American Statistical Association 44:335-341, 1949.

Surface d'un étang par Monte-Carlo

- Utilisation plus subtile :

Sur 100 cailloux envoyés,
34 « ploufs » !

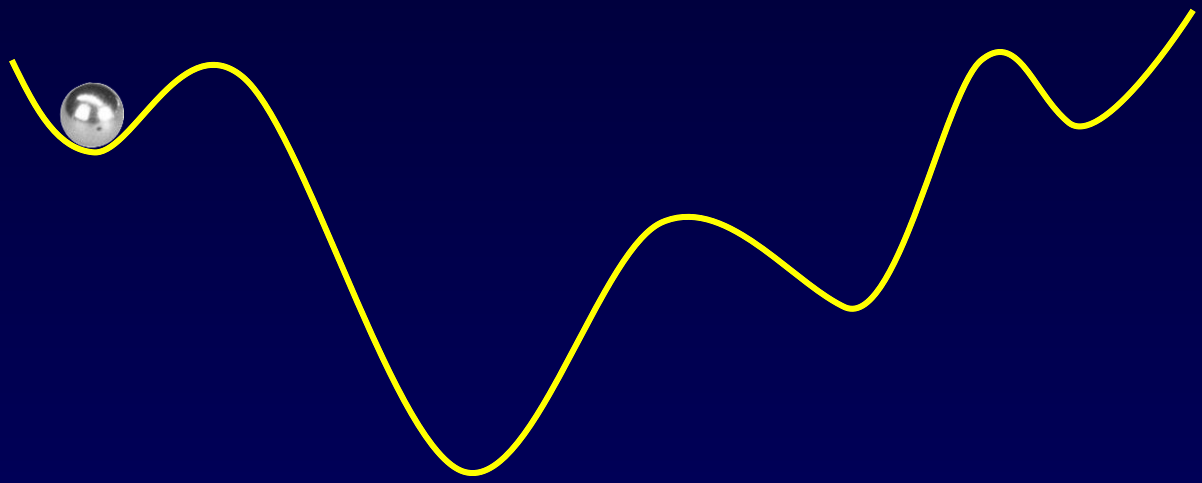


- Pour augmenter la précision : envoyer plus de cailloux !
- Qualité du résultat dépendant de la qualité du générateur de nombres pseudo-aléatoires !
  - Mersenne twister (2è version)

Recuit simulé (simulated annealing)

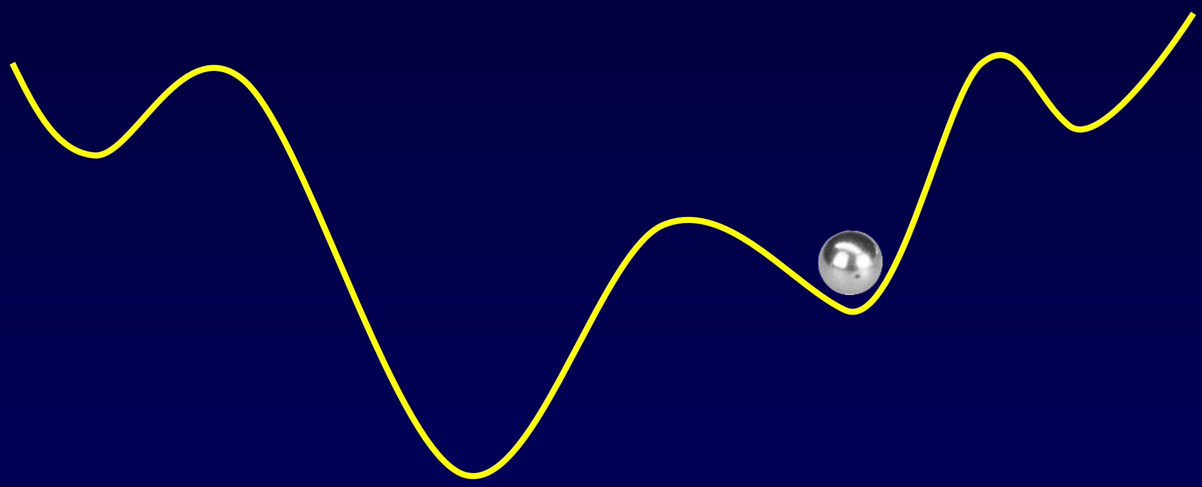
- Kirkpatrick, Gelatt, Vecchi, 1983.
- Analogie avec procédés métallurgiques pour un « Hill-Climbing avancé »
- Au début, x_0 pour lequel on calcul l' « énergie » E_0 (fonction d'évaluation qu'on cherche à minimiser) et on choisit arbitrairement une « température » de départ élevée T_0.
- On se déplace la solution dans Ω dépendant de T.
- Si la solution est meilleure, on la garde. Sinon, on prend la nouvelle solution avec la probabilité $e^{-(\Delta E)/T}$ (algo de Metropolis/Hastings).
- On fait baisser la température, (linéairement ou par paliers).
  - Variation linéaire : T_{i+1}=0,99 T_i
- Sous certaines conditions, on finit par trouver l'optimum global après une série de mutations de plus en plus faibles (température qui diminue) en un temps infini.

Fonctionnement visuel



On commence par secouer fort

Fonctionnement visuel



Puis on secoue moins fort

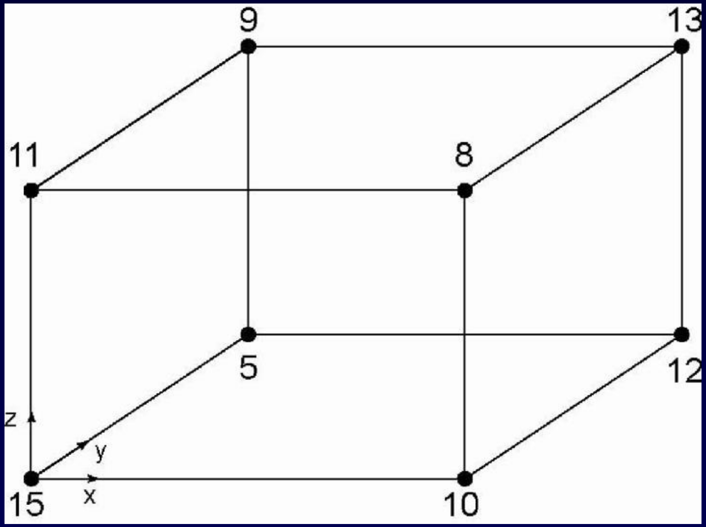
Garantie de trouver l'optimum global

- Recuit simulé : sous certaines conditions, on peut garantir qu'on trouvera avec certitude l'optimum global dans un temps infini.
- Dans la pratique, la convergence requise est si lente que les utilisateurs du recuit simulé font converger l'algo trop vite pour conserver cette garantie ➔ Possibilité de convergence prématurée vers un optimum local, même avec un recuit simulé !!!
- Les algorithmes évolutionnaires ne garantissent pas l'obtention de l'optimum global, sauf à paramétrer pou

Comment choisir quel algorithme ?

- Critère de choix des méthodes de recherche :
  - Complexité du problème (unimodal, multimodal)
  - Type d'espace de recherche (continu, discontinu, discret, mixte)
  - Régularité de la fonction objectif (forme de l'espace de recherche)
- Algorithme stochastique : la qualité des résultats dépend de la qualité du générateur de nombres pseudo-aléatoires

Exemple de recherche Tabou



- On cherche à trouver le nœud de coût minimal, en commençant au nœud 10.
- On mémorise les déplacements effectués, et on s'interdit de revenir en arrière…

1) Parmi les voisins, on choisit 8, suivant z+ (et on met z– dans la liste tabou).
1) Maintenant qu'on est en 8, les voisins sont 11, 13, 10. Le meilleur mouvement (vers 10) est tabou à cause de z–. On va donc vers 11, et on met x+ sur la liste Tabou.
1) Les voisins sont maintenant 9, 8 et 15. Le meilleur mouvement est x+ mais c'est tabou. Du coup, on suit y+, qui sélectionne 9 (et on met y– dans la liste Tabou).
2) Les voisins sont 11, 13 et … 5, mais z– est tabou ! Comme ça améliore néanmoins le résultat, on accepte exceptionnellement, et 5 est l'optimum global.

Parallélisme au niveau des données

Vers la fin des années 80, apparition du parallélisme au niveau des données (*data-level parallelism*) :

- S. Wolfram, « Universality and Complexity in Cellular Automata, » Physica D, 10:1-35, 1984.
- D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing*, MIT Press, 1986.
- M. Minsky, *The Society of Mind*, Basic Books, 1986.
- W. D. Hillis and G. L. Steele, « Data Parallel Algorithms, » Communications of the ACM, Vol 29 (12), 1986.
- Thinking Machines Corporation, « Introduction to Data Level Parallelism », Tech Rep 86.14, Cambridge MA, 1986.
- J. Ferber, « Eco Problem Solving by Interactions, » Proc. of the 9th Workshop on Distributed A.I., pp271-335, Seattle, 1989.
- …

Data-Level Parallelism

- Développement des approches « objet », mais surtout :
- Conclusions importantes :
  - Il est possible de reformuler de nombreux algorithmes de manière parallèle.
  - La notion d'explosion combinatoire ne s'applique pas aux éco-systèmes.
  - Les résultats sont des effets de bord découlant de l'activité des agents des éco-systèmes.

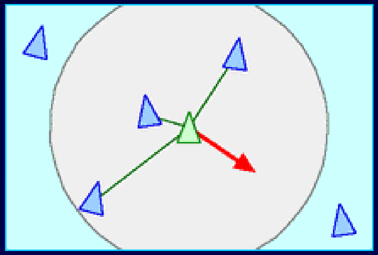
La notion d'émergence devient palpable

Machines hautement parallèles

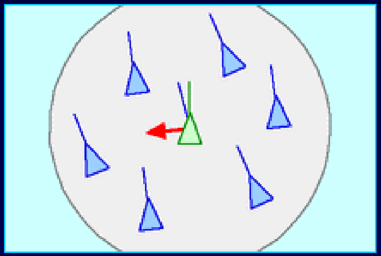
- Tentatives de machines hautement parallèles MIMD :
  - CM-1 (D. Hillis, Thinking Machines, 64K proc. en 1986 !).
- Applications de la CM-1 : implémentation de « particules » simulant des billes qui pouvaient bouger à une vitesse fixe dans 6 directions, ne « connaissant » que leurs voisins directs.
- A grande échelle, le flux des particules est similaire à un fluide.
- Lorsque toutes les particules sont mises en mouvement ensemble, on observe le comportement complexe d'un fluide turbulent, qu'une modélisation avec des équations de Navier-Stokes aurait mis des heures de calcul à simuler.
- La notion de « turbulence » n'a pas été implémentée dans le comportement de chaque particule : il a « émergé » lorsque les particules ont été mises ensemble.

En // algos reproduisant la nature…

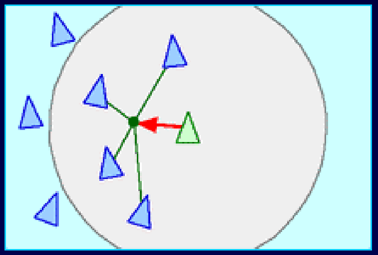
- 1986 : Craig Reynolds programme les Boids suivant 3 règles antagonistes (comportement dynamique) :



Evitement
Éviter les voisins



Alignement
Même direction / vitesse que la moyenne des voisins



Cohésion
Essayer d'aller vers le CG des voisins

Résultat : comportement émergent d'un vol d'oiseaux, d'un banc de poissons, … étrangement réaliste :

/home/collet/Presentation… (film original de 89)

/home/collet/Presentation… (Système interactif 00)

http://www.red3d.com/cwr/boids/

Optimisation par Essaim Particulaire

- Des particules collaborent pour trouver la meilleure solution à un problème.
- Chaque particule (une solution potentielle au problème, sous la forme d'un vecteur de réels) connaît la position de la meilleure particule trouvée (*gbest*) et la meilleure position qu'elle a « personnellement » trouvée (*pbest*).
- Cependant, il ne sert à rien d'aller directement à « *pbest* » ou « *gbest* » car ces solutions ont déjà été évaluées.
- L'idée est donc d'emmener les particules vers « *pbest* » et « *gbest* », mais avec de l'« inertie » en modifiant leur vecteur vitesse plutôt que leur position.

Algorithme PSO

- Chaque particule p connaît :
  - sa position courante (p.pos[i]),
  - la meilleure solution qu'elle a personnellement trouvé (p.pBest[i] pour chaque dimension i),
  - une vélocité pour chaque dimension (p.Veloc[i]),
  - la meilleure solution trouvée par l'essaim (gBest.Pos[i]).
- Ensuite, l'algorithme est le suivant :
  1. Initialiser toutes les particules au hasard.
  2. Clôner la meilleure particule, et la sauvegarder dans gBest, et pour toutes les particules, sauvegarder leur position dans pBestVal.
  3. Calculer une nouvelle vélocité pour toutes les particules et toutes les dimensions suivant l'équation :

     pVeloc[i] ← pVeloc[i]+pInc*rand()*(p.pBest[i]-pPos[i])

     +gInc*rand()*(gBest.pos[i]-pPos[i])

     (Avec rand() renvoyant une valeur entre 0 et 1.)

Algo PSO : suite

1. Déplacer toutes les particules en utilisant les vélocités calculées (i.e. pour toutes les dimensions de toutes les particules : p.pos[i]=p.pos[i]+p.Veloc[i]).
2. Evaluer les particules, et retour en « 2 » jusqu'à ce qu'un des critères d'arrêt soit satisfait.

Une particule est attirée vers 2 endroits : l'endroit où elle a trouvé son meilleur résultat (pBest), et l'endroit où l'essaim a trouvé le meilleur résultat (gBest). Les 2 incréments (pInc, gInc) ajustent la vélocité vers les différents endroits.

Une grande valeur de pInc par rapport à gInc résulte en des individus qui vadrouillent aléatoirement, alors que gInc>>pInc amène à une convergence prématurée.

Le meilleur résultat semble provenir de 2 valeurs égales, valant 2 (pour donner à rand() une moyenne de 1).

Pour Kennedy et Eberhart, le succès de l'OEP consiste à la tendance des particules à louper leur cible.

PSO version 2

- Ajout d'un « Local Best »

Oiseaux

- http://www.youtube.com/watch?v=eakKfY5aHmY&feature=related
- http://www.youtube.com/watch?v=V71hz9wNsgs&feature=related

2000 : modèles émergents connus et utilisés.

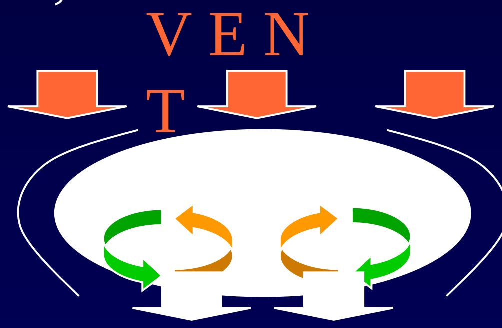
- Maintenant qu'on sait qu'ils existent, on en voit partout et on peut les utiliser !
  - ➢ Manchots empereurs. En dessous de –10°, formation en « tortue.»

  Des mouvements de convection

  apparaissent : par T° ext de –40°

  et vent de 300km/h,

  T° au centre de la tortue : >30° !



  - ➢ Application industrielle : trouver la forme optimale minimisant le rapport surface/volume autour d'un objet 3D complexe (applications en stéréolithographie).
- Evolution : ***concevoir*** des comportements émergents lorsque les conditions s'y prêtent.

Accès à des comportements naturels de groupe

Comportement émergent à l’insu des individus !

- L’impression de contrôle centralisé, intentionnel des bancs de poissons / vol d’oiseaux est très forte mais ***rien de tel*** !!!
- ***Le comportement émergent n’existe que dans les yeux du spectateur.***
- Le poisson/oiseau/… est inconscient du résultat visible à l’échelle supérieure.
- Il ne connaît que ses voisins immédiats !

Et pourtant…

Franchissement d’un niveau d’abstraction.

Fourmis

- 10^17 fourmis, 10¨18 insectes !
- 950K espèces sur 8M espèces d'insectes => peu d'espèces en regard du nombre de fourmis.
- 1 colonie de fourmis africaines (dorylus) peut compter 20M habitants
- Super-colonies : Plus de 300M ouvrières, 1M de reines, 45K noeuds sur 2.7km2 seulement.
- Fourmi d'argentine : 6000 km de côte médirerrannéenne, 200M ouvrières, 120K reines.

- hadi nobahari shortest path

- Antoine dutot ants.html

- ant demo Mark Wodrich

Insectes sociaux (fourmis, termites, …)

- Article fondateur de 1983 (Deneubourg *et al.*) décrivant le comportement de fourmis ayant trouvé un point de nourriture[1].
- 1988, Manderick et Moyson implémentent le comportement d'une fourmi sur ordinateur, retrouvent le comportement d'une fourmilière[2]. Le papier décrit complètement une optimisation par colonie de fourmis.

[1] J.-L. Deneubourg, J.-M. Pasteels et J.-C. Verhaeghe, « Probabilistic behaviour in ants: a strategy of error ?, » Journal of Theoretical Biology, 105, 1983.

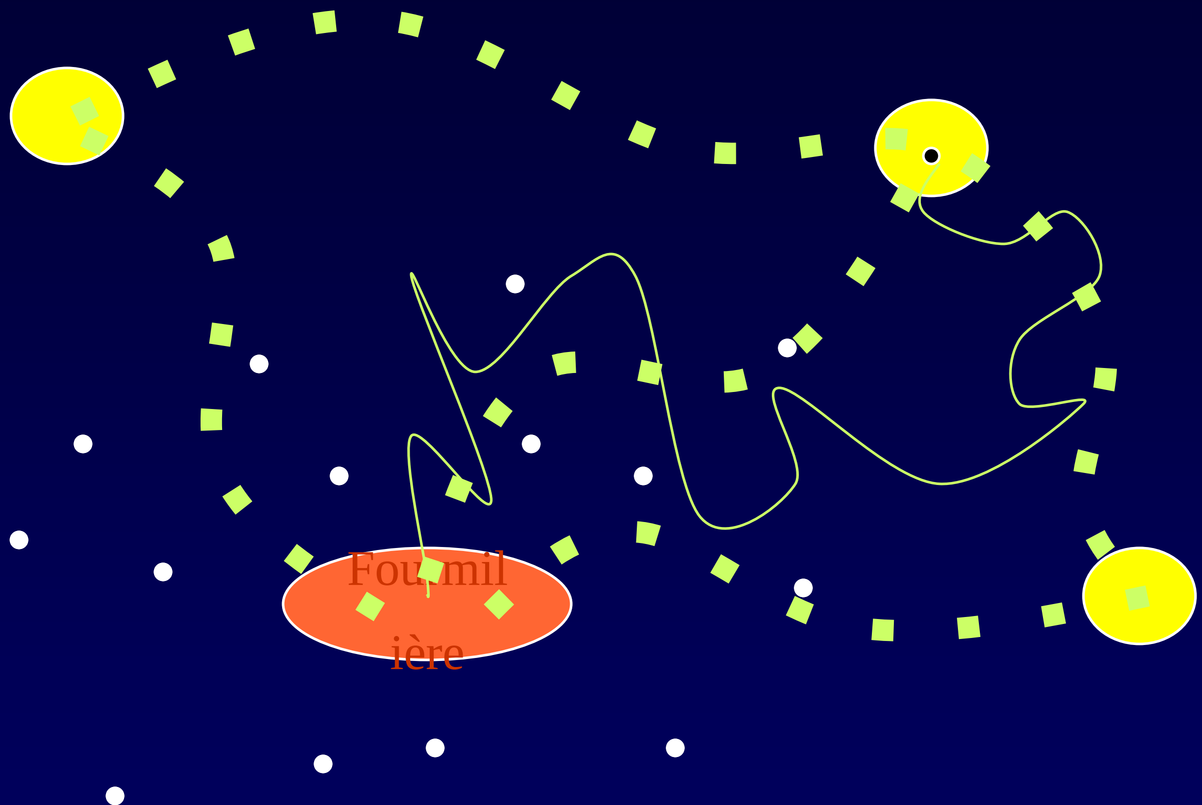
2 F. Moyson et B. Manderick, « The collective behaviour of ants : an example of self-organisation in massive parallelism, » in Proc. of the AAAI Spring Symposium on Parallel Models of Intelligence, Stanford CA, 1988.

Utilisation des paradigmes naturels

- Début des années 90, les ordinateurs deviennent suffisamment puissants pour obtenir des résultats intéressants unissant le parallélisme massif au niveau des données et les techniques trouvées dans la nature :
  - 1992 Dorigo développe l'optimisation par colonies de fourmis et l'applique au TSP (thèse de la Politecnico di Milano).
  - 1995 Kennedy et Eberhart développent l'Optimisation par Essaim Particulaire, où un essaim de particules parcourt un espace de recherche pour trouver un optimum global.

J. Kennedy, R. C. Eberhart, « Particle Swarm Optimisation, » in Proc. of ICNN IV, pp1942-1948, 1995

Optimisation par colonie de fourmis



Phéromones + stigmergie + erreurs =
émergence de chemin optimal

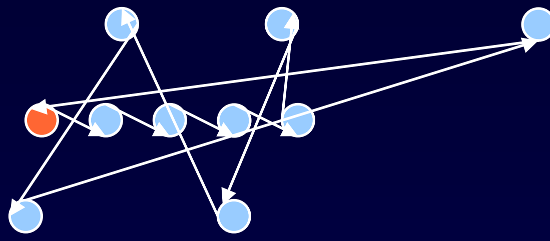
Simulation

- http://litis.univ-lehavre.fr/~dutot/MiscResearch.html

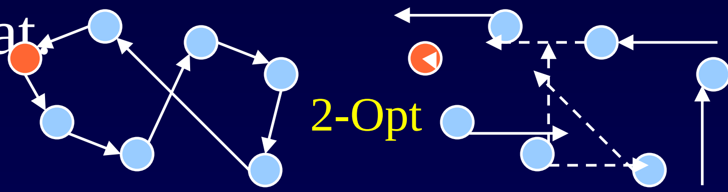
Résolution du TSP

- Méthode du plus proche voisin…

  (aussi appelé algorithme « glouton »)



- *n*-Opt : à chaque itération, on cherche une permutation de *n* arcs qui améliorerait le résultat.



- Combien de tests nécessaires pour un 2-Opt sur n villes ?

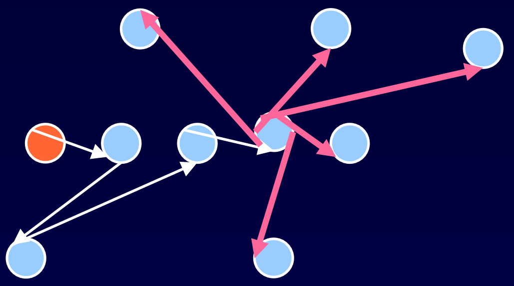
$$\sum_{i=1}^{n-1} i$$

- C’est pas très cher.

  Mais si la bonne solution venait d’un 3-Opt ? Ou d’un 25-Opt ?

Résolution du TSP par ACO (OCF)

- Probabilité pour une fourmi de choisir la ville suivante :



$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}$$

pour $j \in \mathcal{N}_i^k$ le voisinage atteignable de la fourmi k située en i, $\eta_{ij} = 1/d_{ij}$, une information heuristique disponible (ici, la distance), $\tau_{ij}(t)$ le taux de phéromone sur l'arc (i,j).

- Les deux variables α et β déterminent l'influence à donner au chemin de phéromone, et à l'heuristique.
- Si $\alpha = 0$, la probabilité de sélection de l'arc ne dépend que de l'heuristique, et on est en présence d'un algorithme glouton stochastique (avec des points de départ multiples, car les fourmis sont distribuées au hasard sur les nœuds).
- Si $\beta = 0$, il n'y a plus rien à « optimiser, » et les fourmis convergent rapidement sur un chemin quelconque.

Évaporation

- Après que chaque fourmi a fait un tour (une fois qu'elle a construit une séquence de n villes), on évapore les traces de phéromones sur chaque arc, puis on renforce l'arc avec la solution de la fourmi avec l'équation:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

où $0 < \rho < 1$ est le « taux » d'évaporation, m le nombre de fourmis, et $\Delta\tau^{k}_{ij}$ est la dose de phéromone laissée sur l'arc, défini de la manière suivante :

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} 1/L^{k}(t) & \text{si l'arc } (i,j) \text{ est utilisé par la fourmi } k \\ 0 & \end{cases}$$

et où $L^{k}(t)$ est la longueur du tour de la fourmi k.

Les arcs parcourus par beaucoup de fourmis, et qui font partie de « petits » tours, auront plus de choix d'être choisis dans le futur.

Ajustements pour augmenter l'exploitation

- Pour rendre l'OCF compétitive avec l'état de l'art en TSP, il faut ajouter une pincée de poudre de perlimpinpin…
- Stratégie élitiste : On donne au meilleur tour trouvé au temps *t* un poids plus grand. Si l'arc (*i*,*j*) fait partie du meilleur tour :

$$\Delta\tau_{ij}^{gb}(t) = \begin{cases} e/L^{gb}(t) \\ 0 \end{cases}$$

  avec $e > 0$.
- Autre modif : seules les *n* meilleures fourmis ont le droit de déposer leur phéromones.
- …
- (En fait, l'OCF explore trop, et n'exploite pas assez…)

min max ant system (Stützle)

- Seule la meilleure fourmi dépose des phéromones
- on borne les valeurs de phéromones dans un intervalle [min,max] pour éviter que les fourmis ne divergent de trop.