Les métaheuristiques et leur adaptation pour l'optimisation continue

Patrick SIARRY, Univ. de Paris-Est Créteil (LiSSi, E.A. 3956)

Principe des principales métaheuristiques

Cadre général de l'optimisation continue « difficile »

Notre adaptation au cas continu de 4 métaheuristiques

Exemple d'application continue en génie biomédical

Principe des principales métaheuristiques

Les métaheuristiques sont des « méthodes approchées »

Grandes familles de méthodes approchées :

- Méthodes par construction
- Méthodes par décomposition
- Méthodes de voisinage (<u>une</u> solution courante) :
 - * heuristiques classiques
 - * métaheuristiques de voisinage :
 - recuit simulé
 - recherche tabou, etc
- Métaheuristiques à base de population :
 - algorithmes évolutionnaires
 - colonies de fourmis
 - estimation de distribution
 - essaims particulaires, etc

Méthodes par construction

A chaque itération, les décisions sont prises <u>définitivement</u> pour un sous-ensemble de variables.

Décisions pour X_o

Décisions pour X₁

Décisions pour X₂

•

Décisions pour X_q

on cherche un optimum local à chaque prise de décision

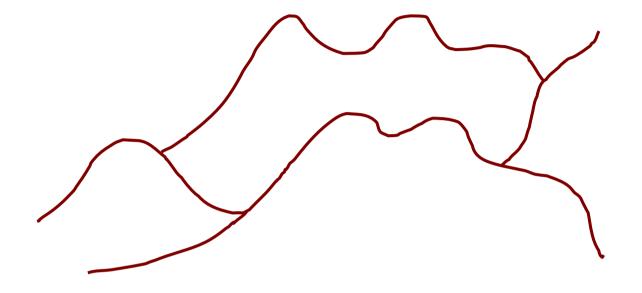
algorithmes *gloutons* ou gourmands

Méthodes par décomposition

- de <u>l'ensemble des solutions</u> du problème : procédure par séparation et évaluation
 - méthode arborescente
 - trouve une solution optimale, si on lui en laisse le temps
- de <u>l'ensemble des décisions</u> à prendre : programmation dynamique
 - démonstration d'optimalité sur la récurrence des calculs
 - selon les problèmes, algo. polynomial ou exponentiel

Méthodes de voisinage

(manipulent une seule « solution courante » à la fois)



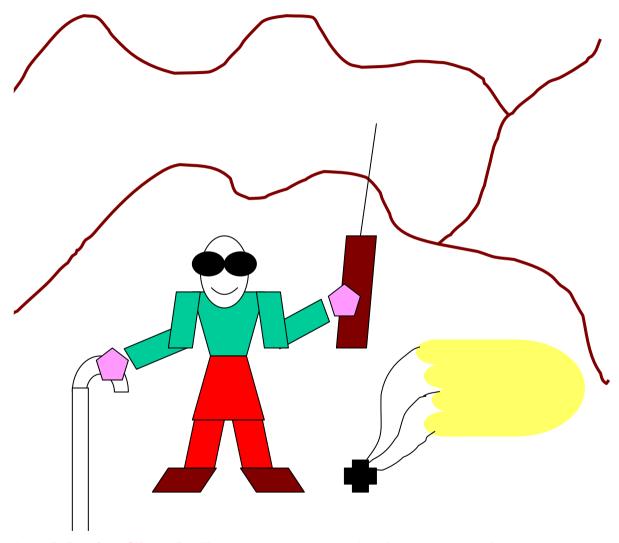
montagnes représentant les variations du niveau de la fonction à minimiser

(en supposant que l'on peut étaler les solutions du problème sur une surface)

L'objectif est de trouver le fond de la vallée la plus profonde

Aveugle parachuté en un point de départ quelconque, muni :

- d'une canne,
- d'un appareil de mesure de niveau,
- d'un appareilde mesure de position,
- d'un appareil pour communiquer.



(quelques transparents empruntés à Marie-Claude Portmann, Ecole des Mines de Nancy)

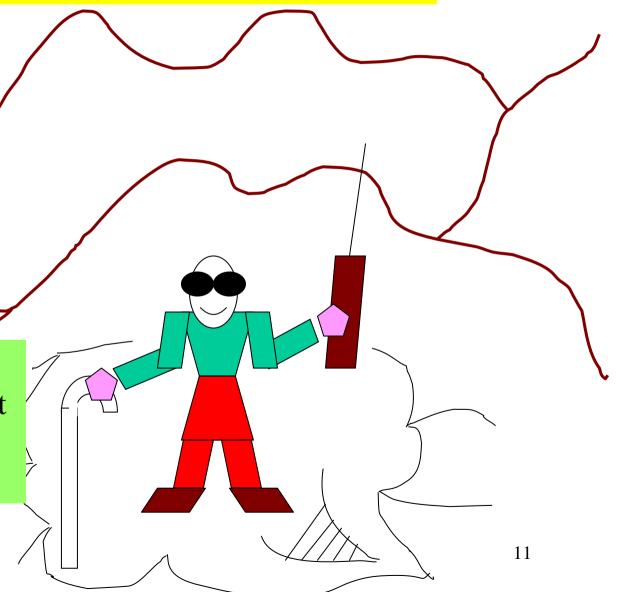


Heuristiques « classiques »

Une approche « classique » : la plus forte pente

L'aveugle choisit toujours le point le plus bas de son voisinage et s'y rend.

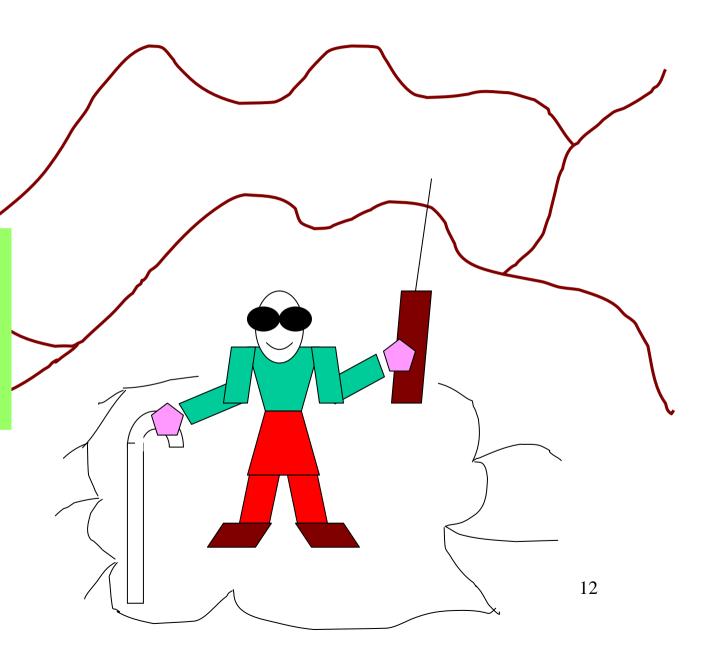
Arrêt dans un optimum local, lorsqu'aucun point du voisinage n'est strictement meilleur.



Seconde méthode Méthode de descente.

L'aveugle choisit le premier point qu'il découvre et qui lui permet de descendre.

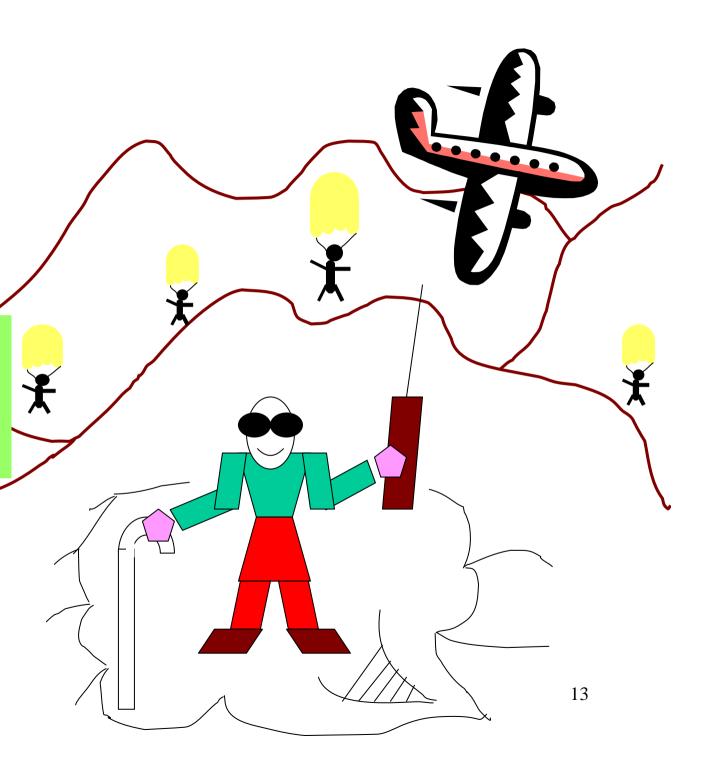
Divers critères d'arrêt.



Amélioration de toute méthode qui se termine sur un minimum local:

appliquer la même méthode avec plusieurs points de départ aléatoires.

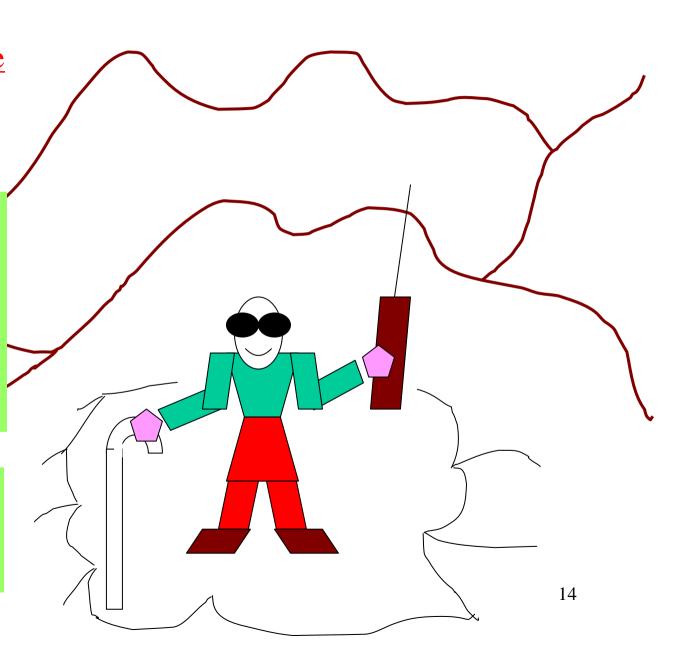
parachutage d'un escadron d'aveugles parachutistes.



Troisième méthode *Marche stochastique*.

L'aveugle erre de manière totalement aléatoire, on retient la meilleure position visitée.

Arrêt par fatigue (après avoir visité N points).



Métaheuristiques de voisinage

Le recuit simulé

Au départ l'aveugle est plein d'énergie : il accepte aléatoirement de remonter, si la remontée n'est pas trop forte et si son énergie est suffisante.

Il perd progressivement son énergie : il termine par une méthode de descente, lorsqu'il n'a plus d'énergie.



Recuit simulé inspiré de la thermodynamique

- Pour que la structuration finale d'un matériau soit harmonieuse, il est nécessaire qu'il passe par des phases intermédiaires moins favorables.
- A haute température, le corps passe par de nombreux stades de transformation, qu'il atteint facilement grâce à l'énergie liée à sa température.
- Plus la température baisse et plus les transformations sont régulières et orientées vers l'état final.

Recuit simulé

Problème d'optimisation

Fonction objectif

Paramètres du problème

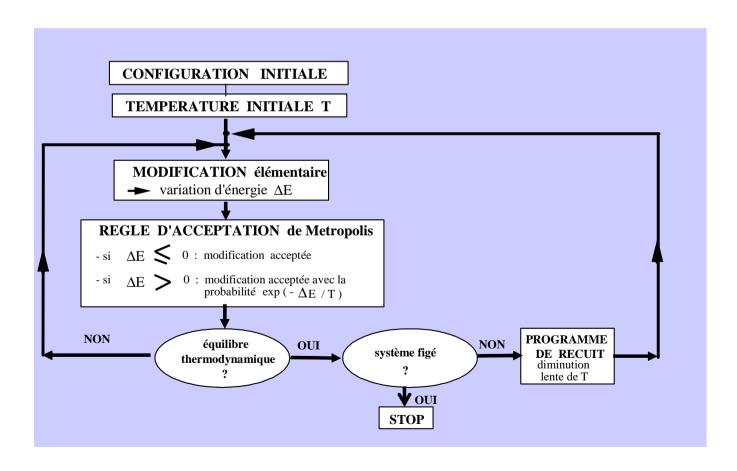
Trouver une bonne configuration

Système physique

énergie libre

coordonnées des particules

trouver les états de basse énergie

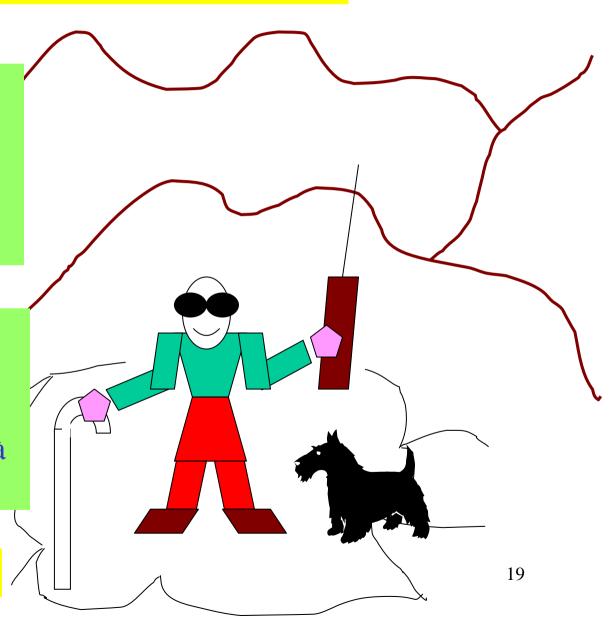


La recherche tabou

On accepte de remonter, pour pouvoir échapper aux optima locaux (en pratique, on va vers le meilleur « voisin »).

Pour ne pas y retomber, on place des « tabous », qui empêchent de retourner aux points déjà visités.

mémoire à court terme.



Problème

On sature la mémoire en y conservant la description de tous les points visités.

Solution

On ne conserve qu'un nombre limité de points visités, les L derniers, dans une liste Tabou.

Problème

On peut avoir tout de même un blocage ou un bouclage.

Variante très utilisée

On ne met pas dans la liste Tabou les derniers points visités, mais les derniers mouvements faits et on interdit les mouvements inverses.

Problème

On risque d'interdire ainsi l'accès à des solutions jamais visitées.

Solution

On lève le statut tabou, si la solution concernée est « suffisamment » intéressante : niveau d'aspiration.

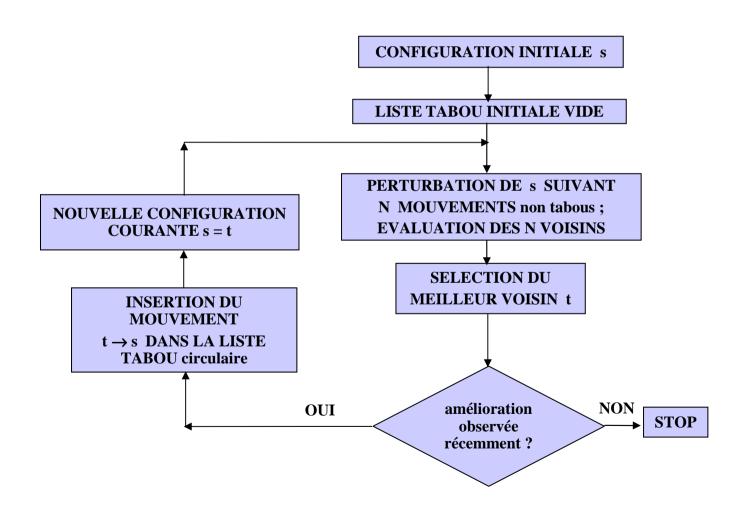
Mécanismes de mémoire à long terme

- diversification : on force la visite de régions non encore explorées.
- intensification : on retourne visiter plus en détail des régions jugées prometteuses.

Un slogan des fans de Tabou : TABOU BAT TOUT

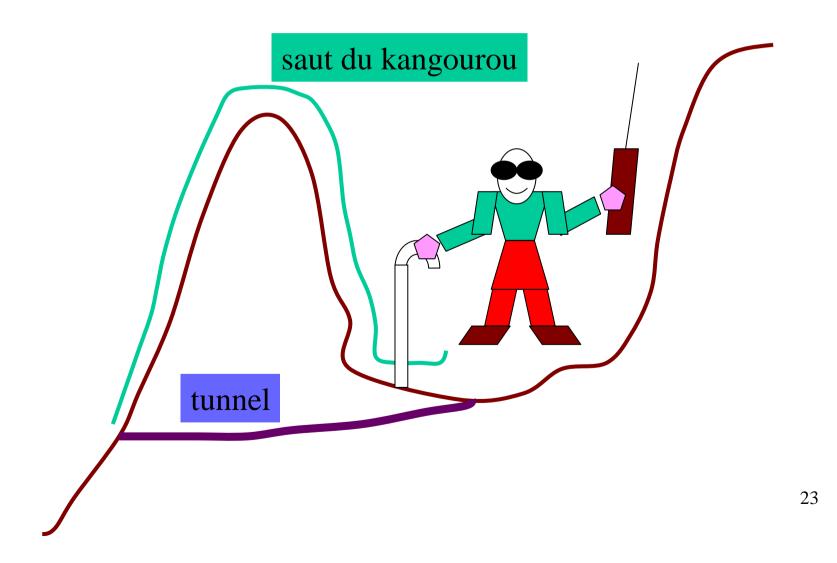
Recherche tabou

Idée de base : modélisation de la mémoire



Sixième méthode Le kangourou ou le tunnel.

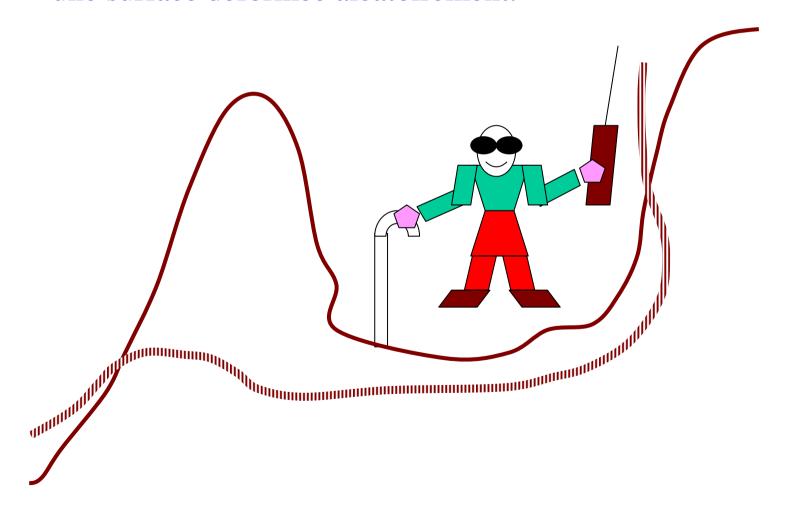
On passe immédiatement à un point distant meilleur, lorsqu'on est coincé dans un optimum local.



Septième méthode

La méthode du bruitage ou le tremblement de terre.

On secoue la montagne, i.e. on travaille temporairement sur une surface déformée aléatoirement.



Métaheuristiques à base de population

Algorithmes évolutionnaires

Famille d'algorithmes s'inspirant de l'évolution des espèces vivantes.

L'évolution d'une espèce est marquée par une suite de transformations, permettant d'améliorer l'adaptation de l'espèce à son milieu.

Cette adaptation est réalisée grâce à la sélection naturelle (*survival of the fittest*) et aux mécanismes de la reproduction.

Algorithme génétique (AG) type

Une population est un ensemble d'individus ; chaque individu représente une solution du problème d'optimisation considéré.

Le plus souvent, un individu, ou chromosome, est représenté par une chaîne de bits, ou d'entiers, ou - plus récemment - de réels, de longueur fixée.

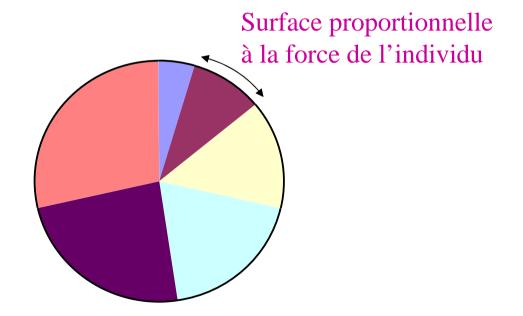
Ces constituants sont appelés les gènes de l'individu.

La qualité de cette solution est évaluée à l'aide de la fonction d'adaptation (*fitness function*), qui est bâtie à partir de la fonction objectif.

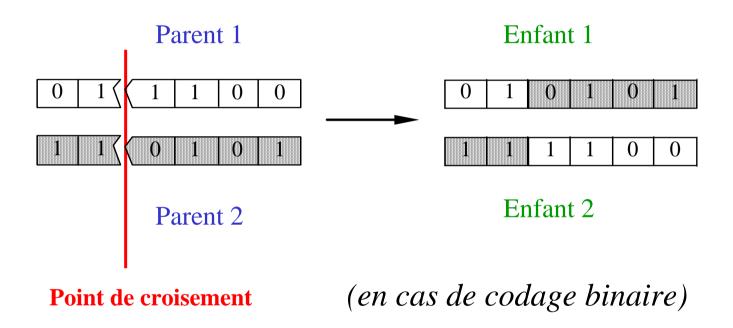
La population évolue grâce à 3 opérations principales :

la <u>sélection</u>, qui favorise les "bons" individus, au détriment des "mauvais" :

Exemple
la sélection par
"roulette biaisée"

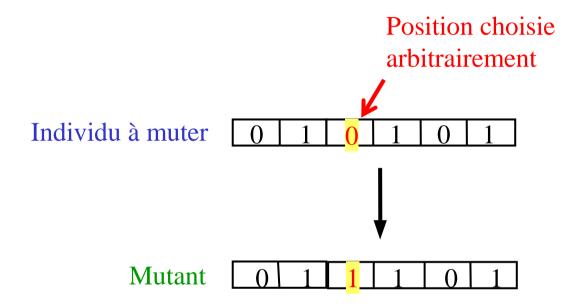


le <u>croisement</u>, qui engendre 2 enfants, héritant des caractéristiques de chacun de leurs 2 parents :



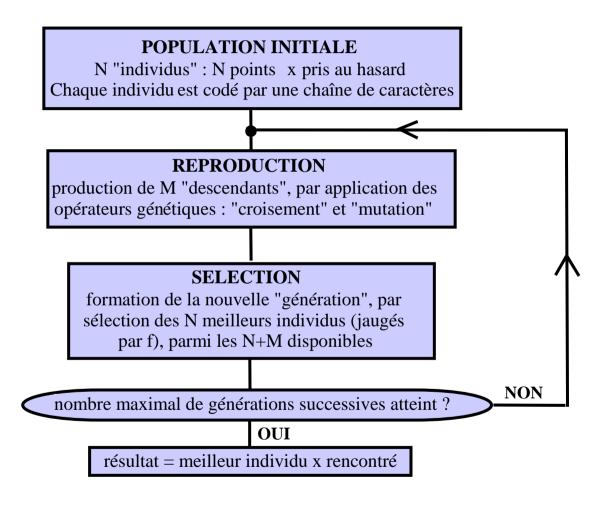
Les croisements cherchent à conserver les meilleures caractéristiques des parents.

la mutation, qui altère un seul individu:



Les mutations servent à créer de la diversité, pour éviter la dégénérescence.

Algorithme génétique (AG)



En général, la taille de la population est gardée constante.



Théorie des schémas

(sous les strictes hypothèses du SGA de Goldberg)

On démontre que les algorithmes génétiques convergent vers de très bonnes solutions.



"No free lunch theorem"

Il n'existe pas de paramètres optimaux, qui garantiraient la meilleure efficacité possible, quel que soit le problème traité.

Colonies de fourmis





Idée-force:

la **coopération** prend le dessus sur la **compétition** des AGs

Quoique dotées de facultés individuelles très limitées, les fourmis ont bâti une vie en société élaborée.

En particulier, elles collaborent entre elles pour aller chercher de la nourriture à l'extérieur de la fourmilière.

On a observé qu'elles suivent toujours le même chemin, et que ce chemin est le plus court possible.

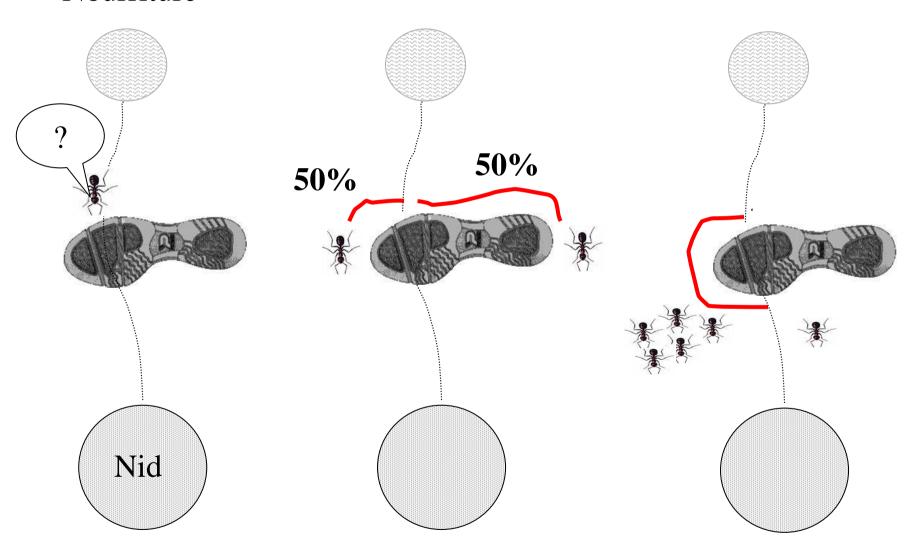
Ce comportement s'explique par un mode de communication indirecte, via l'environnement (la "stigmergie") :

chaque fourmi dépose, le long de son chemin, une substance chimique, la "phéromone".

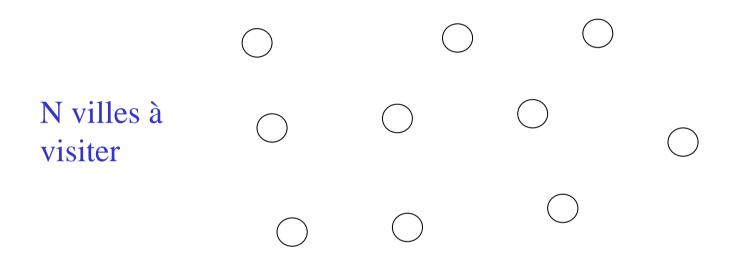
Tous les membres de la colonie perçoivent cette substance, et orientent préférentiellement leur marche vers les régions les plus "odorantes" :

⇒ (par exemple) faculté collective de retrouver le plus court chemin, si celui-ci se trouve obstrué par un obstacle :

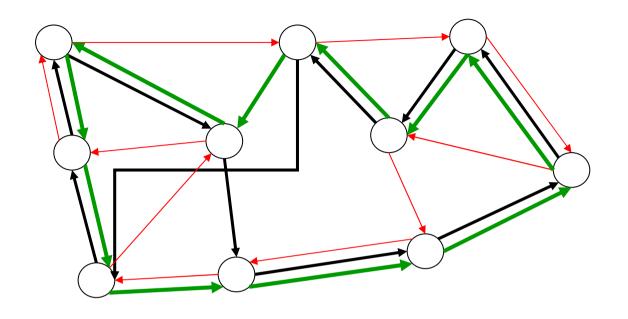
Nourriture



Application au problème du voyageur de commerce

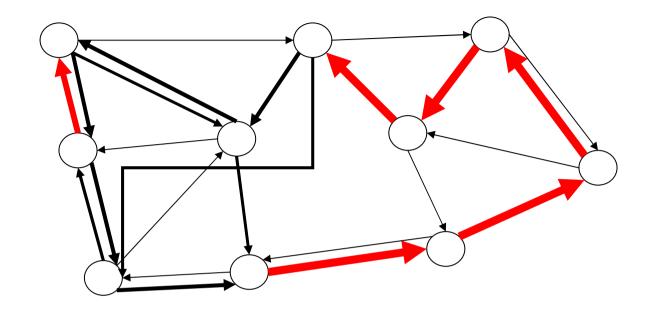


au début de la recherche : pas de trace de phéromone sur les arcs du graphe.



M fourmis (ici 3) partent en exploration aléatoire et effectuent une tournée complète.

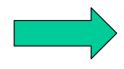
Les traces de phéromone sont mises à jour à la fin, en fonction de l'intérêt des tournées effectuées.



M nouvelles fourmis partent en exploration aléatoire.

A partir de chaque ville, la probabilité de choisir la ville suivante est fonction de la densité de phéromone présente sur l'arête y menant.

Les traces de phéromone sont mises à jour à la fin, en fonction de l'intérêt des tournées effectuées, etc



Caractéristiques des algo. de colonies de fourmis

Parallélisme intrinsèque élevé.

Flexibilité: une colonie de fourmis est capable de s'adapter à des modifications de l'environnement.

Robustesse: une colonie de fourmis est apte à maintenir son activité, si quelques individus sont défaillants.

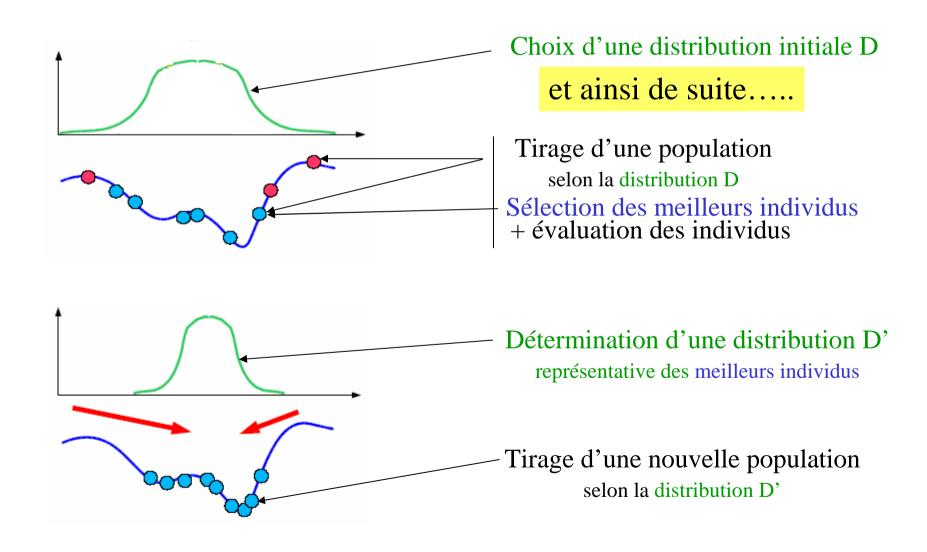
Décentralisation : une colonie de fourmis n'obéit pas à une autorité centralisée.

Auto-organisation: une colonie de fourmis trouve d'ellemême une solution, qui n'est pas connue à l'avance. Les algorithmes de fourmis paraissent indiqués pour les problèmes :

- distribués par nature
- susceptibles d'évolution dynamique
- qui requièrent une forte tolérance aux pannes

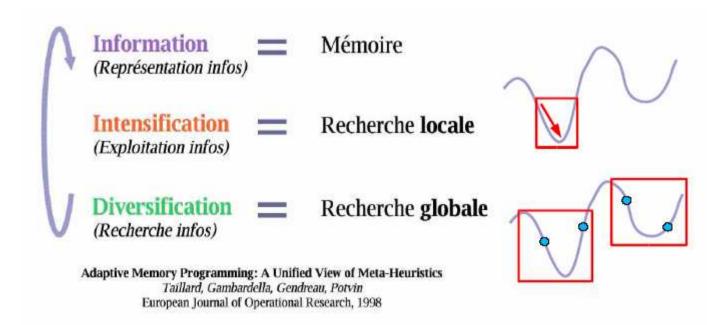
La transposition à chaque problème doit, cependant, faire l'objet d'un traitement spécifique, plus ou moins ardu....

Algorithmes à estimation de distribution (EDA)



Les EDA entrent bien dans le cadre général de la

« programmation à mémoire adaptative »



mémoire = distribution associée aux meilleurs individus

intensification = nouvelle population selon cette distribution

diversification = utilisation d'une population

L'optimisation par essaim particulaire (OEP)

- L'OEP est aussi une technique fondée sur la **coopération** entre des agents : les « particules ».
- Ces agents peuvent être vus comme des « animaux » aux capacités individuelles limitées : peu de mémoire et de facultés de raisonnement.
- L'<u>échange direct</u> d'information entre ces agents rudimentaires permet néanmoins de résoudre des problèmes difficiles.

- C'est le cas, par exemple, chez les abeilles vivant en essaim : exploitation de sources de nourriture, construction de rayons, etc...
- Simple à comprendre et à programmer, l'OEP se révèle efficace pour les problèmes d'optimisation non linéaire, à <u>variables continues</u> ou mixtes.
- Elle a fait ses preuves dans des domaines variés :
 - la régulation de systèmes électriques
 - la conception d'ailes d'avions
 - l'analyse d'images

• • • • •

Schéma de principe du déplacement d'une particule

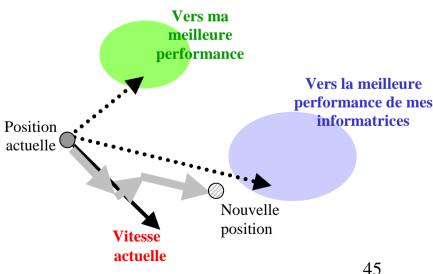
Pour réaliser son prochain mouvement, chaque particule combine trois tendances:

suivre sa propre vitesse

revenir vers sa meilleure performance

aller vers la meilleure performance de ses informatrices

compromis psycho-social, entre confiance en soi et influence de ses relations sociales...



algorithme OEP standard

Pour chaque particule:

$$v_{k+1} = a \otimes v_k + b_1 \otimes r_1 \otimes (p_1 - x_k) + b_2 \otimes r_2 \otimes (p_2 - x_k)$$

$$x_{k+1} = c \otimes x_k + d \otimes v_{k+1}$$

k = numéro de l'itération

x =position de la particule (solution candidate)

v = vitesse de la particule

 p_1 = meilleure position de la particule en question

 p_2 = meilleure position dans le voisinage (essaim)

a = coefficient d'inertie

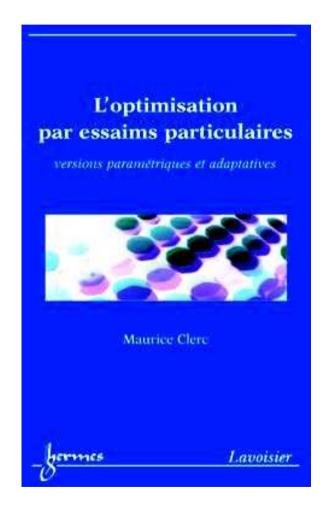
 b_1 , b_2 = coefficients d'attraction

 r_1 , r_2 = nombres aléatoires, de distribution uniforme en [0,1]

Ø = produit vectoriel, élément par élément

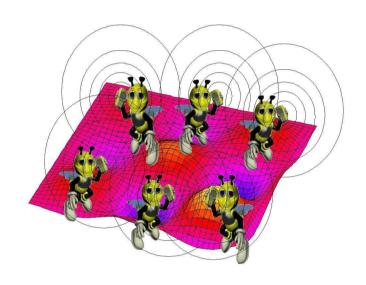


Livre de Maurice Clerc paru en janvier 2005



Maurice Clerc a créé et met à jour régulièrement ce site dédié à l'OEP:

http://www.particleswarm.info/



conférence OEP'2007:

résumés et transparents en ligne :

http://www.particleswarm.info/ oep_2007

Et il y a encore:

Recuit microcanonique

Méthode du seuil

Méthode du grand déluge

Méthode du voyage de record en record

Méthode de recherche distribuée

Méthode GRASP

Algorithme de « scatter search »

Algorithme de « path relinking »

Méthode Aliénor

Méthode de cross-entropie

Méthode de l'évolution différentielle

Stratégies d'évolution

Algorithme des mouches

Systèmes immunitaires artificiels

Algorithmes mémétiques

etc.....

pour ne citer que les espèces les plus connues...

Conclusion sur cet aperçu des métaheuristiques

- Une grande variété de méthodes existantes.
- Beaucoup de choix à faire pour essayer de les rendre les plus efficaces possible.
- Plus les problèmes sont difficiles, plus il devient intéressant d'utiliser des méthodes hybrides, associant plusieurs approches.

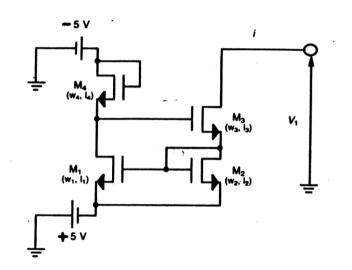
Cadre général de l'optimisation continue « difficile »

Problèmes « difficiles » en variables continues :

(1) Optimisation

Exemple : optimisation des performances d'un circuit électronique

Générateur de courant constant (CEA)

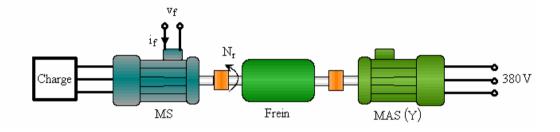


objectif: $i = C^{te}, \forall V_1$

8 variables: dimensions (w_i, l_i) des canaux des 4 transistors MOS

(2) <u>Caractérisation</u>, ou identification:

Exemple : caractérisation d'un modèle de moteur électrique synchrone (CEGELY, Lyon)

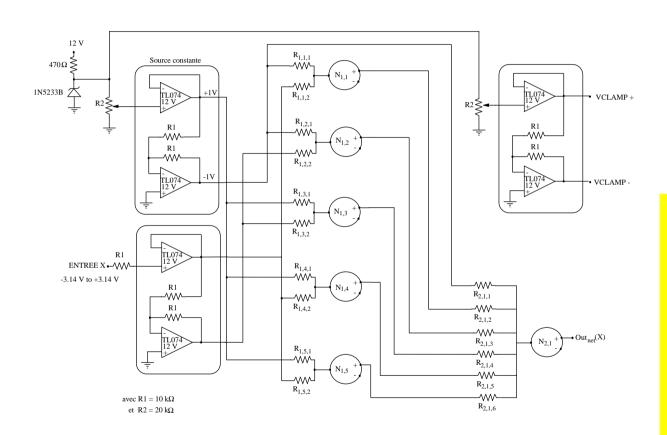


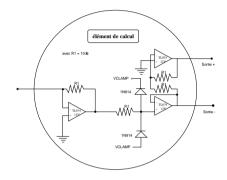
<u>objectif</u>: faire coïncider au mieux les courbes calculées (sorties du modèle) et les courbes expérimentales (mesures relevées sur le moteur)

19 variables : tous les « paramètres » du modèle numérique

(3) Apprentissage (réseau de neurones, base de règles floues)

Exemple : apprentissage d'un réseau de neurones analogique (CEA)





objectif : reproduire au mieux la fonction « sinus » sous forme câblée

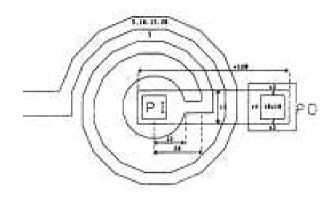
16 variables : les résistances (coefficients synaptiques) du réseau de neurones

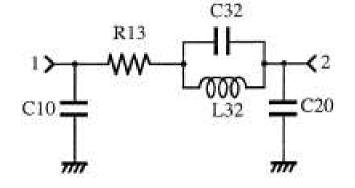
Nombreuses applications continues, dans tous les domaines de l'ingénierie (<u>électronique</u>, mécanique, ...)

Existence de <u>problèmes mixtes</u> discrets / continus :

Exemple:

(J.P. Courat, G. Raynaud, I. Mrad & P. Siarry: Electronic component model minimization based on Log Simulated Annealing, IEEE Trans. on Circuits and Systems <u>41</u> (1994) 790-795)





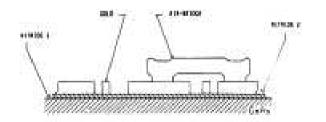


schéma équivalent initial

$$C10 = 60 \text{ fF}$$
; $C20 = 40 \text{ fF}$;
 $R13 = 4 \Omega$;
 $L32 = 3.5 \text{ nH}$; $C32 = 5 \text{ fF}$

inductance MMIC Thomson

Détermination automatique de la topologie <u>et</u> des valeurs des paramètres : « recuit simulé logarithmique »

Hypothèses

- problème mono-objectif
- fonction objectif $f(\mathbf{x})$, à minimiser
- seules contraintes : « contraintes de boîte » :

$$x_i^{MIN} < x_i < x_i^{MAX}$$

Difficultés spécifiques aux problèmes continus :

(on parle aussi de « problèmes difficiles », mais sans référence ici à la théorie de la complexité...)

- pas d'expression analytique de f
- f « bruitée » :
 - bruit expérimental (exploitation de mesures)
 - « bruit de calcul » numérique
 (utilisation d'un simulateur de circuits électroniques)

⇒ gradients de f non accessibles

- f comporte des non-linéarités
- existence de corrélations (non précisément localisées) entre variables

⇒ nombreux minimums locaux

⇒ besoin d'une méthode d'optimisation :

- « directe » (i.e. sans gradients) : interdit l'emploi de méthodes classiques puissantes, de « type Newton »
- « globale » (dans le sens suivant : méthode non piégée, en principe, dans un minimum local)

N.B.: les termes « global » / « local » ne caractérisent pas ici le type de mouvement utilisé (⇒ par exemple, recuit simulé : à la fois local, par son mécanisme, et global, par sa finalité…)

⇒ recours aux métaheuristiques, qui sont toutes à la fois « directes » et « globales »



- l'aspect « direct » des métaheuristiques, lié à leur origine combinatoire, n'a pas d'attrait particulier dans le cas discret
- c'est, au contraire, un avantage très important, dans le cas continu « difficile »



sauf exception (*Essaims particulaires*), métaheuristiques conçues / cadre discret



nécessité d'une « adaptation » aux problèmes continus

Quelques écueils de cette adaptation

- d'ordre « culturel » :
 - applications continues le plus souvent du ressort des sections
 61 & 63 du CNU
 - savoir-faire / métaheuristiques : plutôt section 27
 - informaticiens moins intéressés par les problèmes continus, peu standardisés ⇒ peu de résultats théoriques
- d'ordre « technique » :
 - hétérogénéité des domaines de définition des différentes variables
 - variables de gammes très étendues (plus de 10 décades)

Confrontation avec les études empiriques des concurrents périlleuse :

- codes concurrents rarement publics
- délicat de programmer soi-même les méthodes concurrentes
- résultats publiés difficilement comparables :
 - jeux ≠ | de fonctions de test de domaines d'évolution des variables
 - choix ≠ du mode de sélection du point initial du nb d'exécutions moyennées / résultat
 - définitions ≠ du « succès » (approche de x*)
 de l'« erreur finale » (en f ? en x ?)
 du temps de calcul



grand intérêt pour les travaux d'E. Taillard:

« A statistical test for comparing success rates » : http://www.optimization-online.org/DB_HTML/2003/11/784.html confrontation facilitée, si les concurrents jouent le jeu! La plupart des techniques développées pour adapter une métaheuristique ne sont pas applicables aux autres métaheuristiques...

En continu, comme en discret, aucune métaheuristique ne semble surpasser ses concurrentes...

Expérience / 4 métaheuristiques :

- recuit simulé
- recherche tabou
- algorithmes génétiques
- colonies de fourmis

seules, ou hybridées avec une méthode « directe » et « locale » : l'algo. du polytope de Nelder & Mead

Notre adaptation au cas continu de 4 métaheuristiques

Recuit simulé : cas continu

- Il faut gérer la « discrétisation » de l'espace de recherche
- <u>Procédure (intuitive) possible</u>:
 - variable x_i \langle \Rightarrow \rangle « pas » de discrétisation $STEP_i$:

STEP_i: variation <u>maximale</u> de x_i en un seul « mouvement »

STEP_i est différent pour <u>chaque variable</u> x_i

• pour STEP_i donné, « mouvement » de x_i :

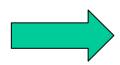
$$x_i' = F(x_i, STEP_i) \rightarrow loi F?$$

- STEP_i doit être ajusté périodiquement, pour « maintenir constante l'efficacité » de l'optimisation, lorsque T diminue :
- → efficacité : peut être évaluée à l'aide :
 - du taux d'acceptation des m^{vts} tentés ?
 - de l'amplitude moyenne des m^{vts} déjà effectués ?
- → fréquence d'ajustement du pas ?
- → loi d'ajustement du pas ?

• Choix des variables x_i concernées par un m^{vt} donné ?

Pas de résultats théoriques :

- pour valider cette procédure intuitive
- pour préciser ses diverses options



approche <u>empirique</u> systématique, au moyen d'une batterie de fonctions de test publiées :

Michalewicz (MZ) (n variables):

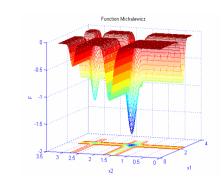
$$MZ(x) = \sum_{i=1}^{n} \sin(x_i) \left[\sin\left(\frac{i \cdot x_i^2}{\Pi}\right) \right]^{20}$$

domaine de recherche : $0 \le x_i \le \pi$, i= 1, n

n = 2, 1 minimum global: $MZ(x^*) = -1.80$

n = 5, 1 minimum global : $MZ(x^*) = -4.687$

n = 10, 1 minimum global : $MZ(x^*) = -9.68$



Goldstein-Price (**GP**) (2 variables) :

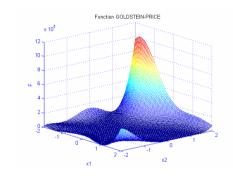
$$GP(x) = [1 + (x_1 + x_2 + 1)^2 * (19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)].$$

$$[30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)]$$

domaine de recherche : $-2 \le x_i \le 2$, i = 1,2;

4 minimums locaux

1 minimum global : $x^* = (-1, 0)$; $GP(x^*) = 3$.



De Jong F6 (DJ F6) (2 variables):

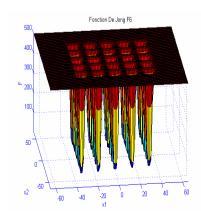
$$F6(x) = \frac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{1 + i + (x_1 - X_i)^6 + (x_2 - Y_i)^6}}$$

 $X_i = [16(i \text{ mod } 5)-2] \& Y_i = 16 [integer(i/5)-2]$

domaine de recherché : $-65 \le x_i \le 65$, i = 1,2 ;

25 minimums : (32j, 32k), j, k = -1, -0.5, 0, 0.5, 1

minimum global : $x^* = (-32, -32)$; $F6(x^*) \cong 0.9980$



<u>Réglage retenu</u>: \rightarrow code disponible

(P. Siarry, G. Berthiau, F. Durbin, J. Haussy, ACM Trans. on Math. Softw. <u>23</u> (1997) 209-228)

- STEP_i initial : ¹/₄ du domaine de variation de x_i
- STEP_i modifié à l'issue de chaque palier de T
- On exploite le taux d'acceptation A_i des m^{vts} tentés, au cours du palier, pour x_i:
 - si $A_i > 20\%$, STEP_i est doublé
 - si A_i < 5%, STEP_i est divisé par 2

• Si le domaine de variation de x_i est « peu étendu » :

$$x_i' = x_i \pm y$$
. $STEP_i$ (y: nb. réel aléatoire $\in [0,1]$)

Sinon (cas de plusieurs décades) : loi logarithmique

 Les n variables du problème sont modifiées par groupes de p, formés aléatoirement : typiquement, p ≅ n/3

Les <u>fréquences</u> de m^{vt} des différentes variables sont <u>égalisées</u>

• 4 critères complémentaires pour l'arrêt du programme

Variante, non « directe »

Succès, tout particulièrement en électronique (*Exemple : IRCOM, Limoges*) d'une variante du recuit simulé continu :

la « diffusion simulée »

(« Simulated Diffusion »: Aluffi-Pentini et al., J. Optimiz. Theory Appl. 47 (1985) 1-16)

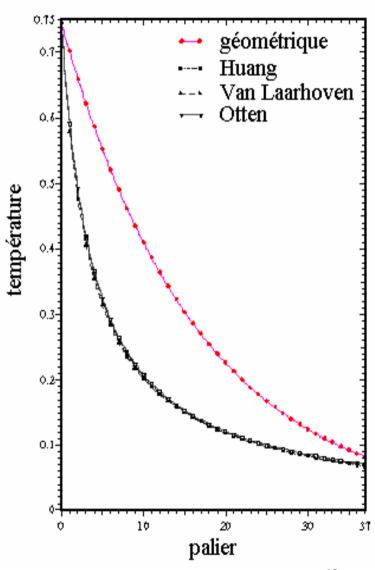
Cette méthode a recours, toutefois, aux gradients de f

Décroissance adaptative de la température

- Etude récente, théorique et expérimentale

(E. Triki, Y. Collette & P. Siarry, EJOR <u>166</u> (2005) 77-92)

- Principaux résultats :
- plusieurs lois adaptatives classiques, ayant des origines et des expressions mathématiques bien différentes, sont, en pratique, équivalentes :



• on montre que ce sont des cas particuliers de la loi générique :

$$T_{k+1} = T_k \cdot \left[1 - T_k \cdot \Delta(T_k) / \sigma^2(T_k) \right]$$

où :
$$\sigma^2(T_k) = < f_{Tk}^2 > - < f_{Tk} >^2$$

 $\Delta(T_k)$: dépend de la loi adaptative choisie

- le choix le plus simple : $\Delta(T_k) = C^{te}$ ne correspond à aucune loi classique
- nous avons montré que ce choix peut être formateur, face à un problème nouveau

Recherche tabou : cas continu

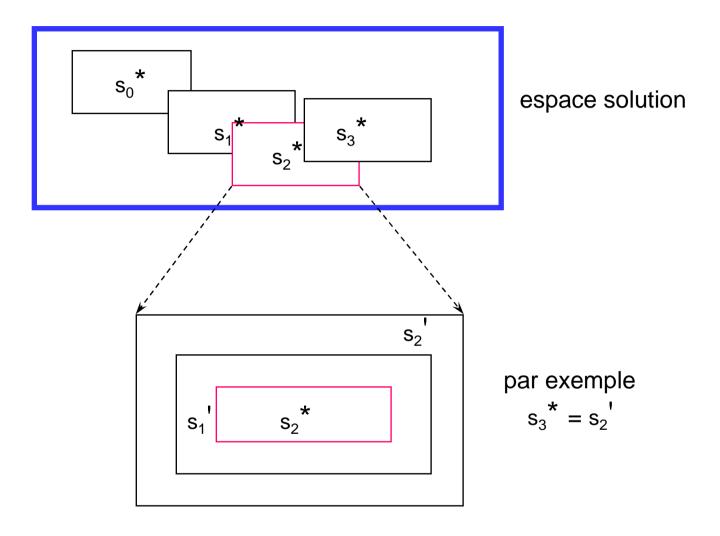
- Elaboration d'un algorithme : \rightarrow code disponible

ECTS (« Enhanced Continuous Tabu Search »): R. Chelouah & P. Siarry, EJOR 123 (2000) 256-270

- ECTS se veut une transposition directe de la méthode combinatoire

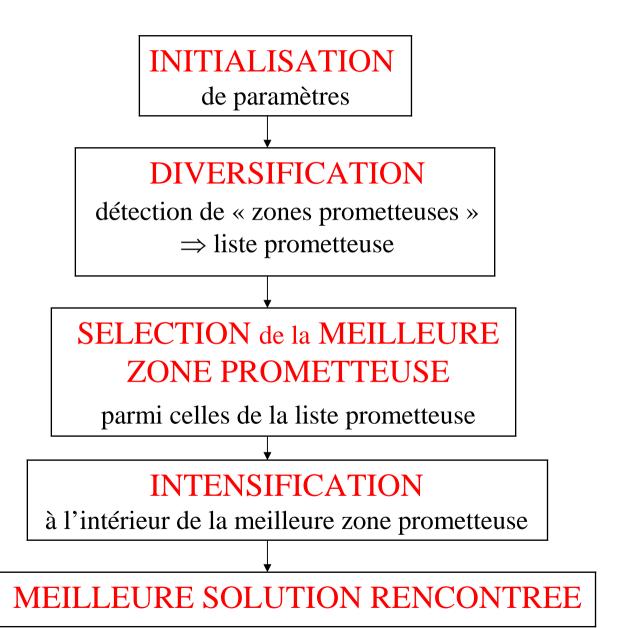
- Liste tabou : formée de « boules » centrées sur les dernières solutions retenues

- Notion de voisinage d'une solution courante :



Sélection aléatoire d'un point dans chaque « couronne » hyper-rectangulaire ⇒ liste de voisins

Structure générale : (cœur de Tabou non décrit)



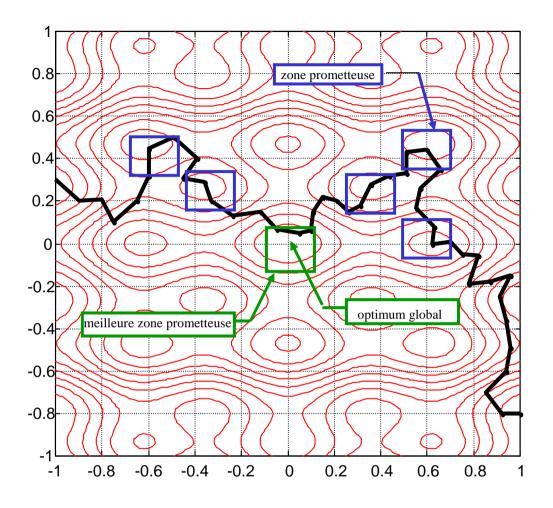
Quelques détails sur l'algorithme ECTS

- diversification :

- « zones prometteuses » → minimum global de f ?
- examen de x : comparaison à tous ses voisins (cœur de Tabou)
- x « prometteur » s'il surpasse « nettement » ses voisins (au sens d'un seuil fixé dynamiquement par ECTS)
- liste prometteuse : « boules » centrées sur les solutions prometteuses

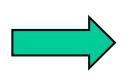
- intensification:

- une nouvelle recherche tabou est lancée à l'intérieur de la meilleure zone prometteuse
- elle opère plus finement (réductions de la dimension de l'espace de recherche, et de la taille de la liste tabou)



Exemple de la *fonction B2* : trajet suivi par ECTS (diversification)

- Principal écueil de cette approche : le nombre élevé de paramètres de réglage :
 - tailles des listes tabou et prometteuse
 - rayons des boules tabou et prometteuse
 - étendue du voisinage
 - nb de voisins du point courant etc
- La plupart des paramètres d'ECTS sont calculés automatiquement, ou ont été figés « au mieux » empiriquement
- Validation expérimentale détaillée d'ECTS

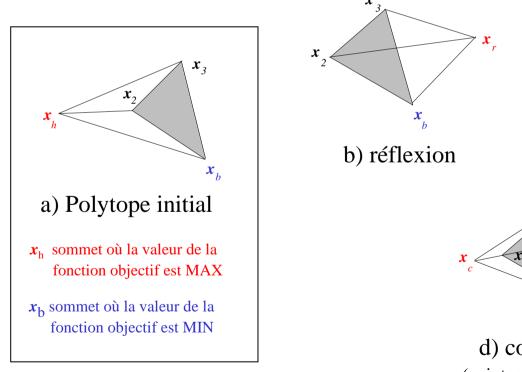


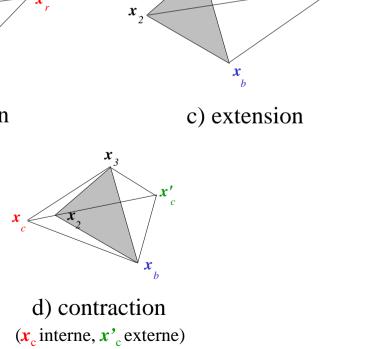
le temps de calcul croît lentement
(≈ linéairement)
en fonction du nb de variables de f

Variante hybride

(R. Chelouah & P. Siarry, EJOR <u>161</u> (2005) 636-654)

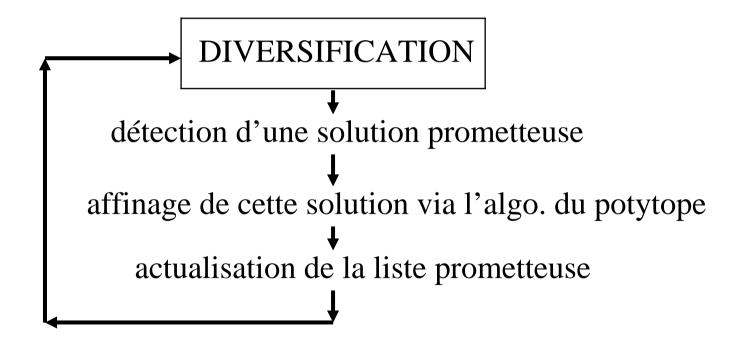
- La convergence peut être accélérée, en ayant recours à l'algo. du polytope de Nelder & Mead :





77

- Les phases de diversification et d'intensification sont alternées :



Algorithmes génétiques: cas continu

- Elaboration d'un 1er algorithme :

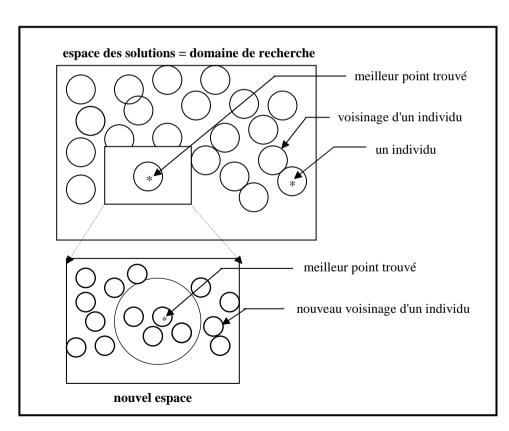
```
CGA (« Continuous Genetic Algorithm ») \rightarrow code disponible R. Chelouah & P. Siarry, J. of Heuristics <u>6</u> (2000) 191-213
```

- Particularités de CGA:
 - alternance de phases de « diversification » et d'« intensification »
 - réductions dynamiques de la taille de la population, et de l'espace de recherche
 - codage réel des individus \Rightarrow affiliation parfois débattue :

```
RCGA (« Real-Coded Genetic Algorithm ») ou EA (« Evolutionary Algorithm ») ?
```

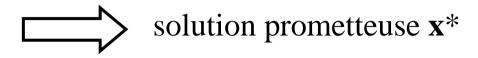
- opérateurs de croisement et de mutation spécifiques
- réduction dynamique de la probabilité de mutation
- alternance « diversification » / « intensification » dans CGA :

effet de « zooms » successifs :



Phase de « diversification » :

AG classique, manipulant des individus « opulents »



Phase d'« intensification »:

- nouvel espace de recherche restreint, autour de \mathbf{x}^*
- nouvel AG, avec une population réduite, formée d'individus
 « amaigris »

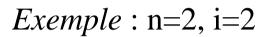
Opérateur de croisement

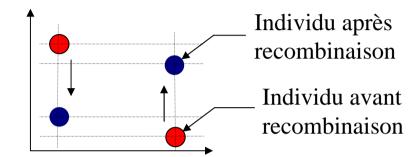
(dénommé « recombinaison » en codage réel)

S'inspire du croisement mono-point en binaire :

- individu = vecteur \mathbf{x} de dimension n \rightarrow ième composante de $\mathbf{x} =$ ième variable d'optimisation
- position i du croisement : sélectionnée au hasard
- les composantes d'indice > i sont échangées entre les 2 parents

• les composantes d'indice i subissent des variations opposées :

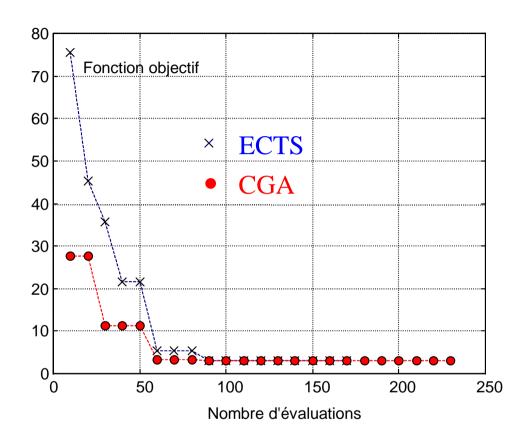




Opérateur de mutation

- affecte la valeur d'une seule variable
- l'amplitude maximale de la perturbation, et la probabilité de mutation sont progressivement diminuées (2ème point <u>délicat</u>)

Exemple typique de résultat



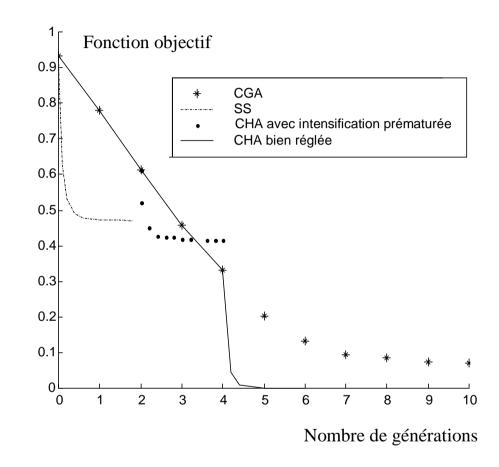
Optimisation de la fonction Goldstein-Price

CGA est gagnant au début, mais s'essouffle ensuite...



On peut effectuer l'intensification via l'algorithme du polytope :

Optimisation de la fonction B2



(R. Chelouah & P. Siarry, EJOR <u>148</u> (2003) 335-348)

→ Principale difficulté : le passage de relais automatique entre les 2 approches

Application, au CEA, de la variante hybride : conception optimale d'un capteur à courants de Foucault (contrôle non destructif)

Atouts majeurs de l'optimisation par AG

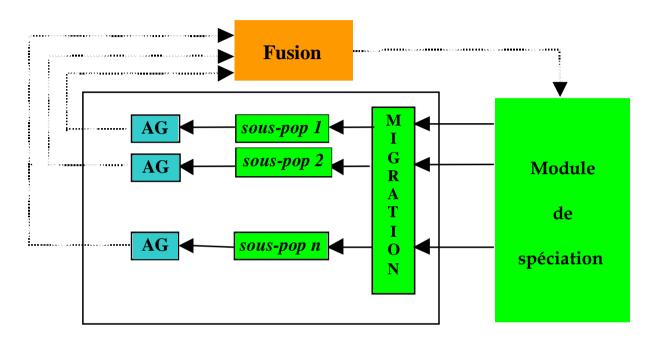
(1) Optimisation « multimodale »

On cherche à repérer tous les minimums locaux de f

Schéma possible: AG multipopulation

(P. Siarry, A. Pétrowski & M. Bessaou, Adv. in Eng. Softw. 33 (2002) 207-213)

Algorithme génétique multipopulation



- (a) Plusieurs AG, manipulant des sous-populations distinctes
 - (b) Fusion, en une population unique, des sous-populations
 - (c) « Spéciation » ⇒ nouvelle répartition des individus en sous-populations, en fonction de leurs ressemblances
 - (d) Retour en (a) ou arrêt

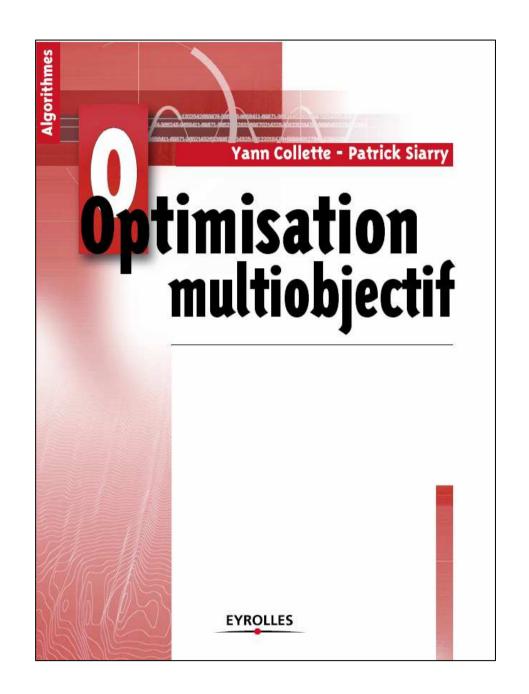
(2) Optimisation « multiobjectif »

Thèse de Yann Collette (Univ. Paris 12, déc. 2002):

AG et recuit simulé appliqués à un problème combinatoire « difficile », à EDF :

l'optimisation des plans de rechargement de combustible nucléaire

Problème ouvert, et <u>non</u> confidentiel



Colonies de fourmis : cas continu

```
2 premiers algorithmes en variables continues : CACO (1995, Bilchev & Parmee)
API (2000, Monmarché & al.)
```

-Principe de CACO (« Continuous Ant Colony Optimization »):

diversification:

- fourmis « globales », chargées de délimiter de grandes « régions »
- création et évaluation des régions : utilisent les mêmes opérateurs qu'un AG

intensification:

- fourmis « locales », affectées aux diverses « régions »
- dépôt de phéromone : proportionné à l'amélioration de f

Inconvénients de CACO :

- complexe : AG + fourmis
- trop de paramètres

-Principe de API:

- S'inspire du comportement d'une fourmi primitive : « Pachycondyla apicalis »
- Simule une forme de communication directe entre 2 fourmis :
 le « tandem-running » → amène à rassembler les individus dans les meilleurs sites de chasse

Inconvénient de API :

N'exploite pas de pistes de phéromone ⇒ pas de « stigmergie » (communication indirecte, via l'environnement)

Nouvel algorithme : CIAC

« Continuous Interacting Ant Colony » (J. Dréo & P. Siarry, Future Generation Computer Systems <u>20</u> (2004) 841-856)

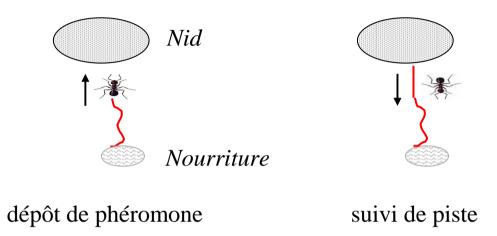
Thèse de Johann Dréo (Univ. Paris 12, déc. 2004)

- → notre idée est :
 - de préserver le cadre des colonies de fourmis
 - d'exploiter plusieurs formes de communication

<u>Insectes sociaux et communication</u>: (Hölldobler & Wilson: « The Ants ». Springer, 1990)

• « Recrutement » : forme de communication , qui entraîne les individus à se regrouper en un endroit où un travail est nécessaire

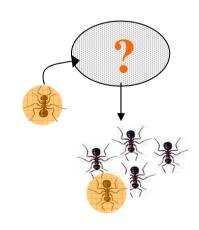
- recrutement indirect par pistes:



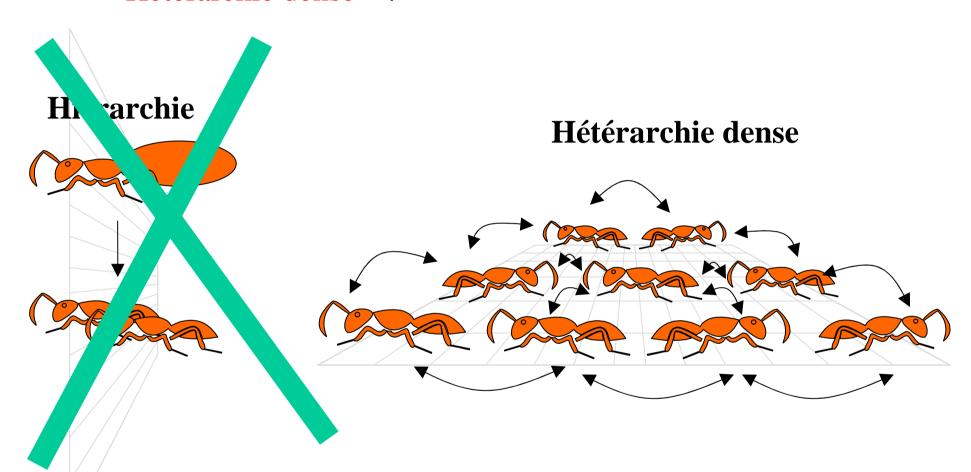
- recrutement direct interindividuel:

exemple : mobilisation

Procède par « trophallaxies » : échanges de nourriture liquide entre 2 individus



• « Hétérarchie dense » :



Wilson & Hölldobler, "Dense Heterarchy and mass communication as the basis of organization in ant colonies". Trends in Ecology and Evolution, <u>3</u> (1988) 65-68

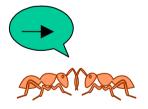
Principale particularité de CIAC :

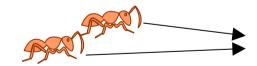
2 canaux de communication

1) spots de phéromone



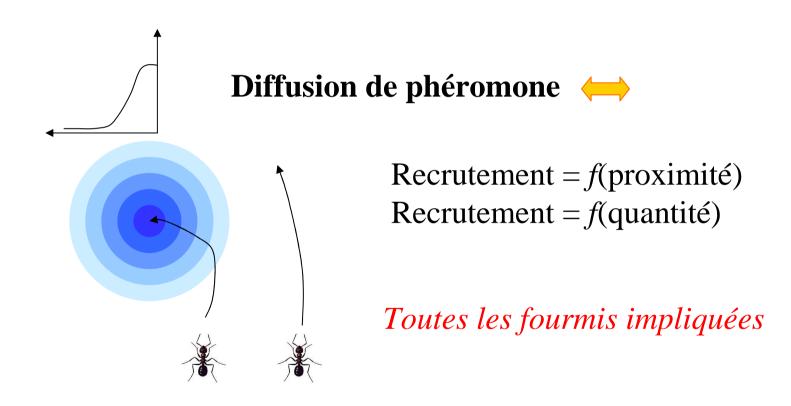
- ⇒ mémoire externe / collective = NIVEAU GLOBAL
- 2) messages entre2 individus





⇒ mémoire interne / individuelle = NIVEAU LOCAL

Niveau global



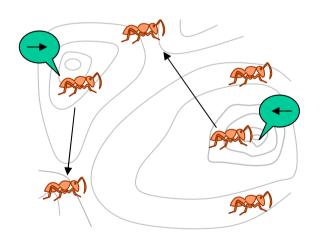
Niveau local

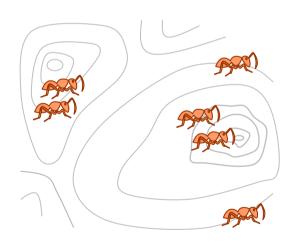
Envoi de messages



Informations:

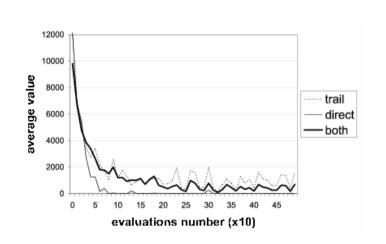
Amélioration fonction objectif Position zone d'intérêt

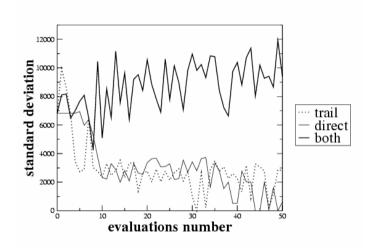




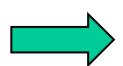
Résultats

• La <u>synergie</u> entre les deux canaux de communication a été observée empiriquement :



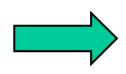


En présence des 2 canaux, l'écart-type des valeurs de f est plus grand, et présente de plus amples oscillations



- « auto-organisation », avec alternance :
- de phases de diversification (σ grand)
- de phases d'intensification (σ petit)

- <u>Validation</u> sur un jeu de 13 fonctions de test « statiques »
 - → performances généralement inférieures / autres métaheuristiques

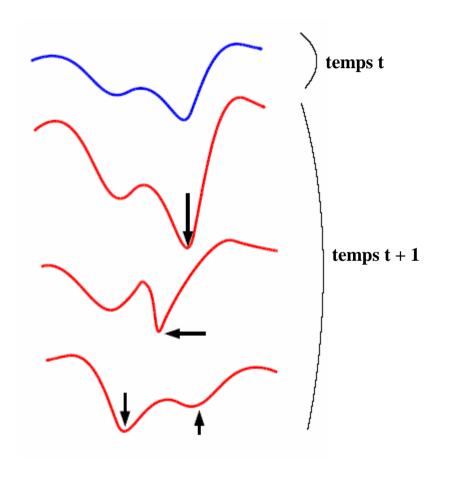


2 orientations actuelles:

- (classique) hybridation application en optimisation continue « dynamique »

En optimisation dynamique, on attend une meilleure exploitation de la flexibilité et de l'auto-organisation inhérentes à un algorithme de colonie de fourmis

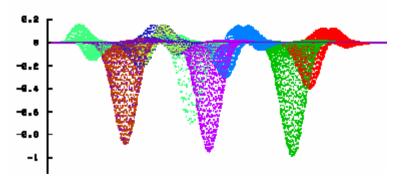
Optimisation dynamique



Formalisation:

- Classes de problèmes de base
- Benchmark

OPLF OPLS APLF APLS OVPF OVPS AVPF AVPS ADLS O(P+V)PS



Exemple d'application de l'optimisation continue en *Génie biomédical*

Recalage d'angiographies rétiniennes via un algorithme de colonie de fourmis

Maladies de la rétine

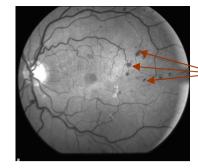
- > cause majeure de <u>cécité</u> à tout âge
- Dégénérescence Maculaire Liée à l'Age (DMLA)
- Rétinopathie Diabétique (RD)
- Rétinite à Cyto-Megalo Virus (RCMV)

- Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi Université Paris 12, Créteil)
- > Service Ophtalmologique Universitaire de Créteil (SOUC Hôpital intercommunal, Université Paris 12, Créteil)

Imagerie de la rétine

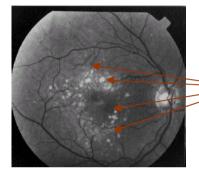
- Angiographie rétinienne
 - Technique de diagnostic dynamique
 - Acquisition des images
 Angiographe classique ou Ophtalmoscope à laser
 - Colorant fluorescent Fluorescéine sodium ou Vert d'Indo-Cyanine
- Tomographie à cohérence optique
- Analyseur d'épaisseur rétinienne

Filtre monochromatique



Tâches d'hémorragies

Filtre monochromatique

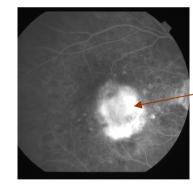


« Drusens »

Techniques de Diagnostic

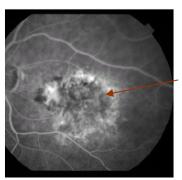
Pathologies

Angiographie fluorescéine



Vaisseau choroïdien classique ou bien défini

Angiographie fluorescéine



Vaisseau choroïdien occulté ou mal défini

10

Méthode nécessaire pour le recalage d'angiographie

Recalage des angiographies rétiniennes

- Mouvements oculaires
- Fond non uniforme
- Sphéricité de la rétine
- Variations de fluorescence

Travaux précédents

=> Recalage par « flot optique »

- J.C. Nunes, "Analyse multiéchelle d'image. Application à l'angiographie rétinienne et à la DMLA", Thèse de doctorat, Université Paris 12, Créteil. Décembre 2003.
- J.C. Nunes, Y. Bouaoune, E. Deléchelle, and P. Bunel, "A Multiscale Registration Scheme of Retinal Angiograms". *Computer Vision and Image Understanding Journal* 95, 2004, pp. 129-149.

B. Recalage des angiographies rétiniennes

Méthode proposée

Pré-traitement non-linéaire

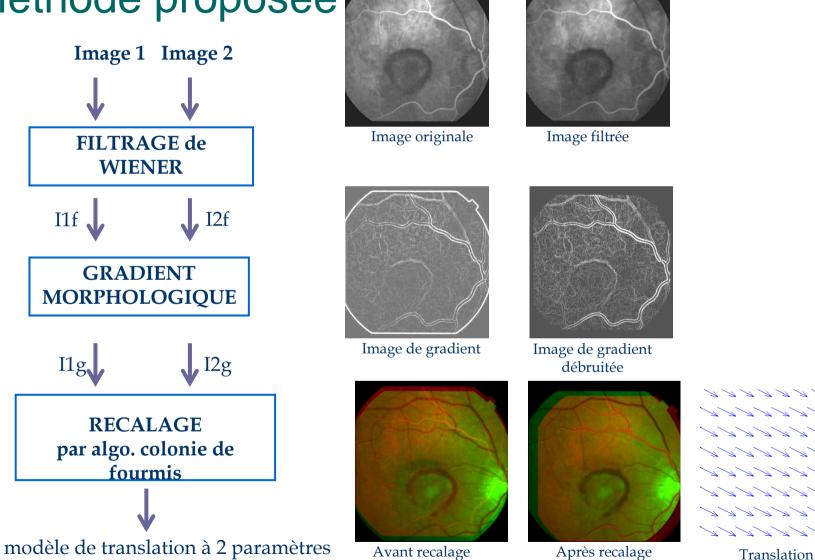
- Traitement d'image
 - débruitage (filtrage de Wiener)
- Calcul de gradient
 - > gradient morphologique

Recalage

- Critère de similarité
 - somme des carrés des différences d'intensité entre images de gradient
- Algorithme de colonie de fourmis
 - > modèle de translation à 2 paramètres

B. Recalage des angiographies rétiniennes

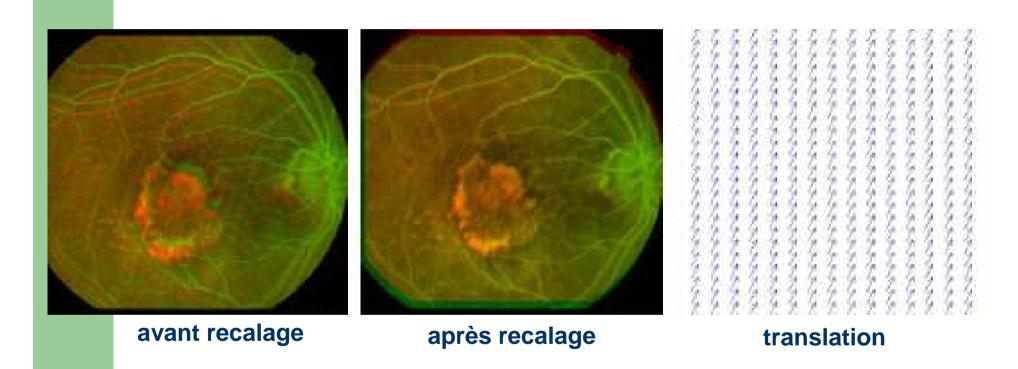
Méthode proposée



B. Recalage des angiographies rétiniennes Colonie de fourmis Fonction de similarité 12 10 30 critère de similarité 10 somme des carrés des différences d'intensité entre images de gradient

C. Résultats

Recalage par translation



Angiographie Fluorescéine

C. Résultats 1) **Angiographie** après recalage avant recalage Fluorescéine 2) avant recalage après recalage

D. Conclusion

Conclusion

- robuste face aux variations d'éclairement, bruits, contrastes locaux faibles, mouvements oculaires
- totalement automatique
- pas d'extraction des caractéristiques rétiniennes
- faible temps de calcul
- temps de calcul peu dépendant de la résolution
 - ⇒ recalage d'images à haute résolution

Quelques pages de publicité pour finir...

IEEE Congress on Evolutionary Computation CEC'2005 Edinburgh, UK, September 2-5, 2005

Special Session on *Real-Parameter Optimization*Prof. Kalyanmoy Deb and A/Prof. P. N. Suganthan

epnsugan@ntu.edu.sg

Chosen Benchmark functions, evaluation criteria and program files are available from :

http://www.ntu.edu.sg/home/EPNSugan

Sessions organisées dans le cadre des conférences CEC successives

| CEC'05 | Single Objective Optimization |
|--------|---|
| CEC'06 | Constrained Single Objective Optimization |
| CEC'07 | Multi-Objective Optimization |
| CEC'08 | Large-Scale Global Optimization |
| CEC'09 | Dynamic and Uncertain Environments |
| CEC'09 | Constrained Multi-Objective Optimization |

European Journal of Operational Research

Feature Issue on

Adaptation of Discrete Metaheuristics to Continuous Optimization

Guest editors: Zbigniew Michalewicz and Patrick Siarry

Published in volume 185

RAIRO Operations Research

Feature Issue on

Cooperative Methods for Multiobjective Optimization

Guest editors: Clarisse Dhaenens, Patrick Siarry and El-Ghazali Talbi

Published in volume 42

Int. J. of Computational Intelligence Research

Feature Issue on

Particle Swarm Optimization

Guest editors: Maurice Clerc, James Kennedy and Patrick Siarry

Published in volume 4

European Journal of Operational Research

Feature Issue on

Metaheuristics in Transportation and Logistics

Guest editors: Patrick Siarry and Eric Taillard

Published in volume 195

Engineering Applications of Artificial Intelligence

Feature Issue on

Advances in Metaheuristics for Hard Optimization: New Trends and Case Studies

Guest editor: Patrick Siarry

Published in volume 23

Engineering Applications of Artificial Intelligence

Feature Issue on

Local Search Algorithms for Real-World Scheduling and Planning Applications

Guest editors: Raymond Chiong and Patrick Siarry

To appear

Ouvrage collectif sur les métaheuristiques (sept. 2003)

- Recuit simulé
- Recherche tabou
- Algorithmes évolutionnaires
- Colonies de fourmis
- Programmation à mémoire adaptative

.

• 3 « études de cas » détaillées



Ouvrage collectif sur un domaine d'application

(mars 2007)

Métaheuristiques présentes dans 8 des 13 chapitres

mais aussi autres techniques diverses:

- ondelettes
- critères d'information
- programmation quadratique
- inférence bayésienne
- approches markoviennes
- modèles de Markov cachés



Information - Commande - Communication

Optimisation en traitement du signal et de l'image

sous la direction de Patrick Siarry



Lavoisier

Ouvrage collectif sur les colonies de fourmis

(déc. 2009)



Traité Informatique et Systèmes d'Information

Fourmis artificielles 1

des bases de l'optimisation aux applications industrielles

> sous la direction de Nicolas Monmarché Frédéric Guinand Patrick Siarry



Lavoisier



Traité Informatique et Systèmes d'Information

Fourmis artificielles 2

nouvelles directions pour une intelligence collective

> sous la direction de Nicolas Monmarché Frédéric Guinand Patrick Siarry



Lavoisier

Ouvrage collectif sur un domaine d'application (juillet 2010)



Traité Informatique et Systèmes d'Information

Partitionnement de graphe

optimisation et applications

sous la direction de Charles-Edmond Bichot Patrick Siarry



Lavoisier

Groupe META

Collaborations nouvelles, via le groupe META, commun aux GdR MACS et RO du CNRS :

Animateurs:

P. Siarry siarry@univ-paris12.fr

E.G. Talbi talbi@lifl.fr

http://www.univ-valenciennes.fr/GDR-MACS/groupes_details.php?gt=META

International conference on swarm intelligence

Theoretical advances and real world applications
June 14-15, 2011, EISTI, Cergy, France

Conference chairs: Rachid Chelouah and Patrick Siarry

Invited guests

James KENNEDY
and
Maurice CLERC

http://icsi11.eisti.fr/