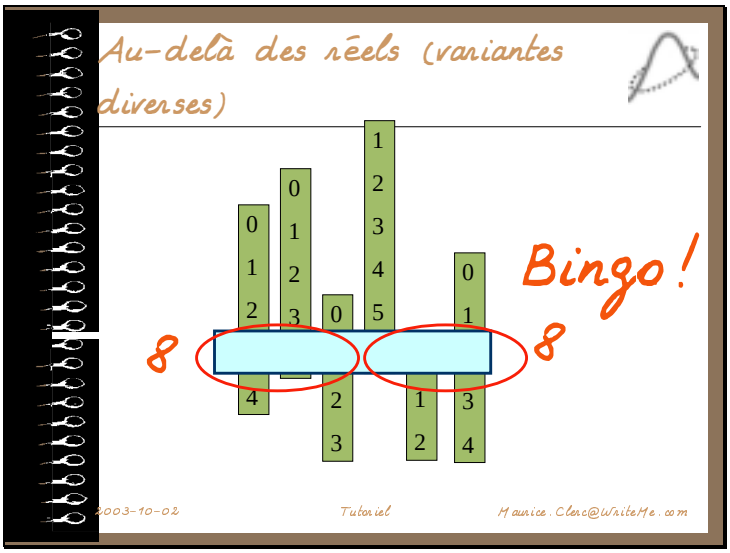
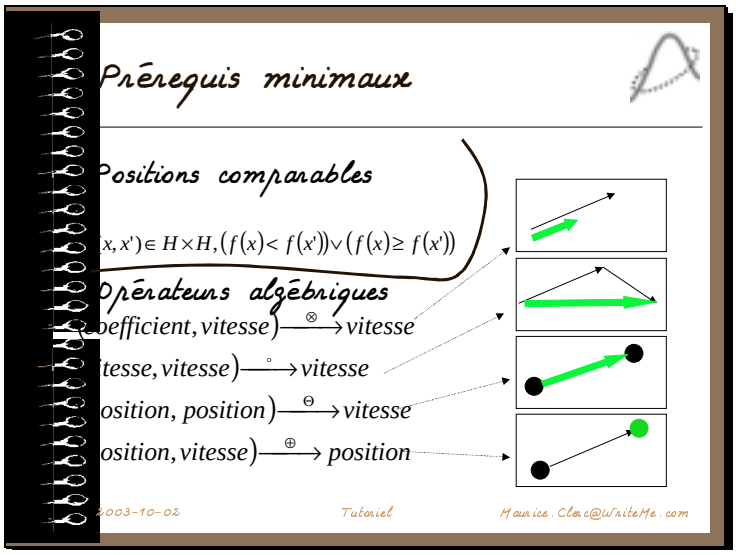
Partie II : variantes

Diapositive 1



Si l'on examine les principes de base de l'algorithme, on s'aperçoit que les structures mathématiques entrant en jeu sont relativement faibles et, donc, que son domaine d'application est assez général. En particulier, ni l'espace de recherche ni la fonction à optimiser n'ont besoin d'être continus.

Diapositive 2



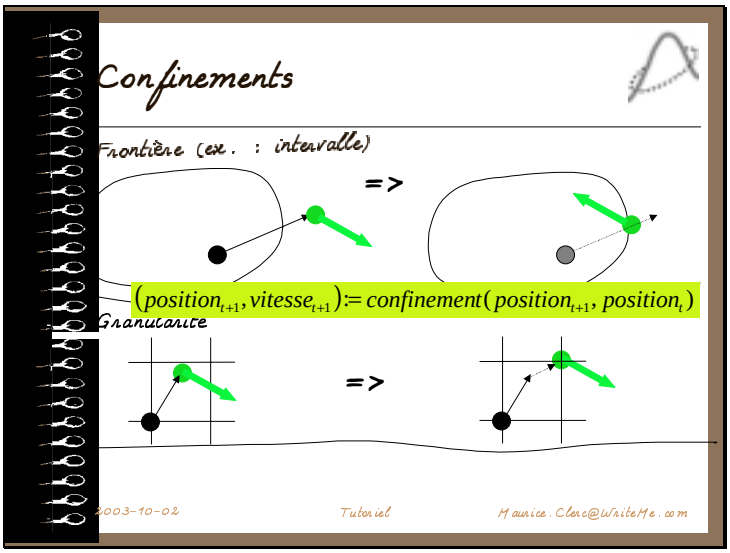
Pour appliquer la méthode, il suffit en effet que les conditions suivantes soient remplies :

- deux positions quelconques doivent être comparables. C'est évidemment un minimum.

- quatre opérateurs algébriques doivent être définis, permettant d'écrire les équations de mouvement.

Ces conditions sont beaucoup moins restrictives que l'on peut parfois croire. Il est, par exemple, parfaitement possible de définir de tels opérateurs pour des problèmes combinatoires type « voyageur de commerce » ou « coloriage de graphe ».

Diapositive 3

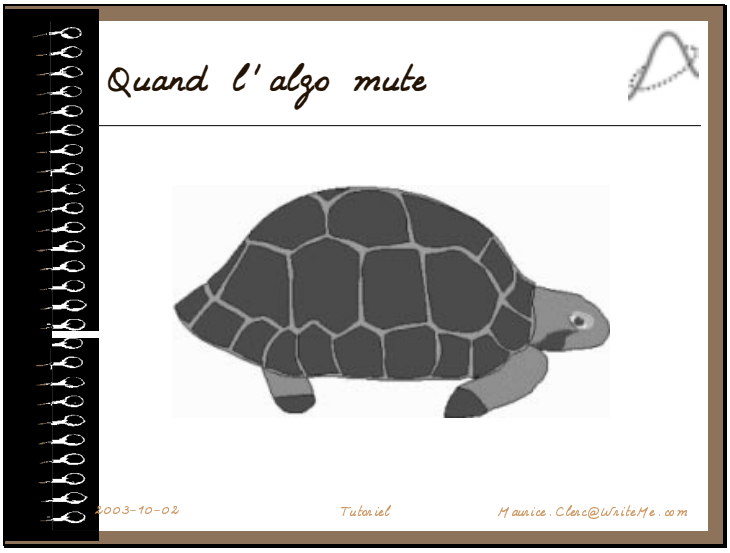


Par ailleurs, même si cela reste parfois implicite, il est toujours nécessaire de définir un ou plusieurs opérateurs de confinement. Le plus classique est le *confinement d'intervalle* : si une composante d'une position tend à sortir de l'intervalle des valeurs admissibles, on l'y maintient, quitte à modifier la vitesse pour rester cohérent.

Un autre également très utile est le *confinement de granularité*. Si l'espace de recherche est discret et que l'on garde des équations de mouvement à valeurs continues, la position trouvée n'est alors pas toujours acceptable. On la déplace donc aussi vers la plus proche position appartenant à l'espace de recherche.

Un troisième opérateur de confinement est parfois utilisé dans des problèmes combinatoires. Faute de mieux, on l'appellera « *tous différents* ». Il assure, au moyen d'un petit sous-algorithme que je ne détaillerai pas ici, que toute position a toutes ses composantes (généralement entières) différentes.

Diapositive 4



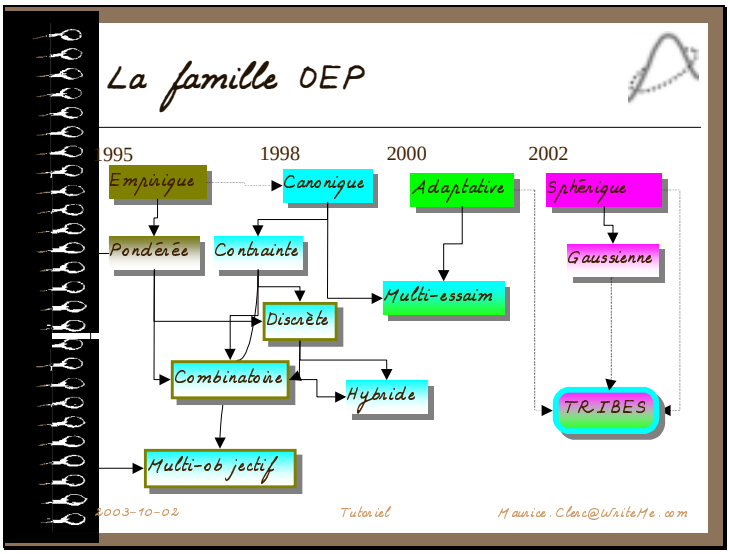
Outre des conditions d'utilisation très générales, on a vu qu'il y avait largement la place pour des améliorations possibles, en particulier concernant l'évaluation des paramètres (adaptation automatique) et le choix des distributions statistiques autour des meilleures performances (sphériques ou gaussiennes ou autres).

Il était donc normal que l'on assiste à un foisonnement de variantes. Pour simplifier, on peut considérer qu'il y en a de deux types, même si la frontière est parfois un peu floue :

- celles qui cherchent uniquement à améliorer les performances, par exemple en utilisant plusieurs essaims,

- celles qui cherchent à étendre le domaine de compétence de l'OEP, par exemple aux problèmes combinatoires ou au multiobjectif.

Diapositive 5



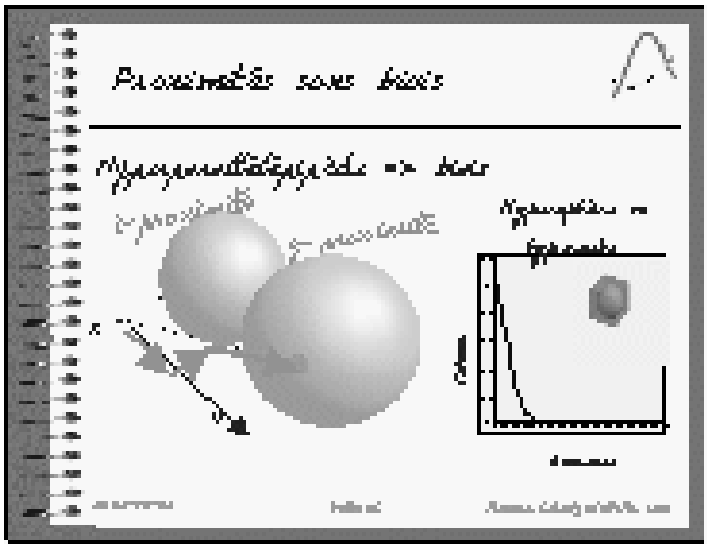
Ce tableau, très incomplet, montre quelques-unes des principales familles.

A mon sens, les avancées les plus intéressantes concernent les *méthodes hybrides*, qui combinent l'OEP et des techniques de recherche locale pour des problèmes combinatoires, et les *méthodes adaptatives*, qui tentent d'éviter à l'utilisateur la nécessité d'ajuster différents paramètres.

Le multi-essaim, également, semble assez prometteur, en ce qu'il met en jeu un mécanisme de répulsion entre essaim, qui favorise l'exploration de l'espace de recherche.
Les méthodes hybrides sortent du cadre de ce tutoriel, consacré à l'OEP « pure », si j'ose dire. D'ailleurs, à ce jour, les résultats sont équivalents, mais sans plus, à ceux obtenus par d'autres méthodes relevant également de ce que l'on appelle parfois l'intelligence distribuée ou collective, tels les algorithmes génétiques ou les colonies de fourmis. Il est vrai aussi, qu'à ma connaissance, très peu de chercheurs travaillent dans ce domaine et il reste, à l'évidence, beaucoup à faire.

Je m'attarderai donc sur les méthodes adaptatives, mais, au préalable, dirai quelques mots des distributions statistiques sans biais qui commencent à être exploitées par différents auteurs. En effet, ces deux éléments sont au cœur du programme Tribes que nous allons évoquer rapidement ensuite.

Diapositive 6



Dans la version classique, un point est choisi au hasard, de façon uniforme, dans un hyperparallélépipède autour de la meilleure position jamais atteinte par la particule, p_i (et de même pour la meilleure performance p_g dans le voisinage).

Cette technique n'est pas justifiée d'un point de vue théorique, mais utilisée seulement pour sa simplicité de programmation avec la plupart des langages, qui disposent seulement d'une fonction « distribution uniforme dans un intervalle » (souvent appelée rand()).

Une forme sphérique est sensiblement meilleure (je ne considère ici, pour simplifier, que le cas de dimensions de même nature), pour deux raisons :

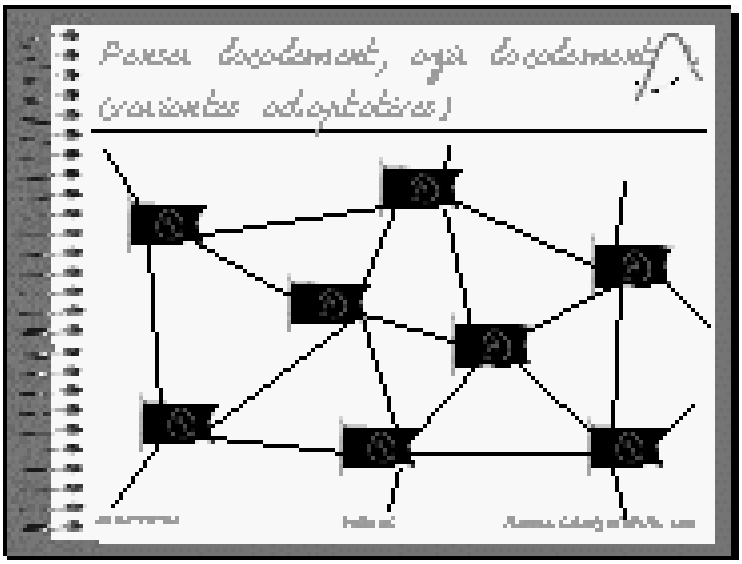
- elle ne privilégie arbitrairement aucune direction de l'espace

- elle permet de localiser plus finement le point choisi. En effet, si l'on compare les volumes inclus dans une hypersphère et son hypercube tangent, on constate que leur rapport décroît très rapidement (cela n'est pas intuitif, d'ailleurs).

L'inconvénient est qu'il est plus compliqué de programmer la fonction « un point au hasard uniformément dans une hypersphère », tout au moins si l'on ne veut pas des temps de calcul prohibitifs quand le nombre de dimensions augmente. Soit dit en passant, c'est un excellent exercice de statistiques !

Plus généralement on peut utiliser des distributions à support infini, type gaussiennes. On peut alors montrer que l'écart-type optimum est très près de norme(p_i-p_g).

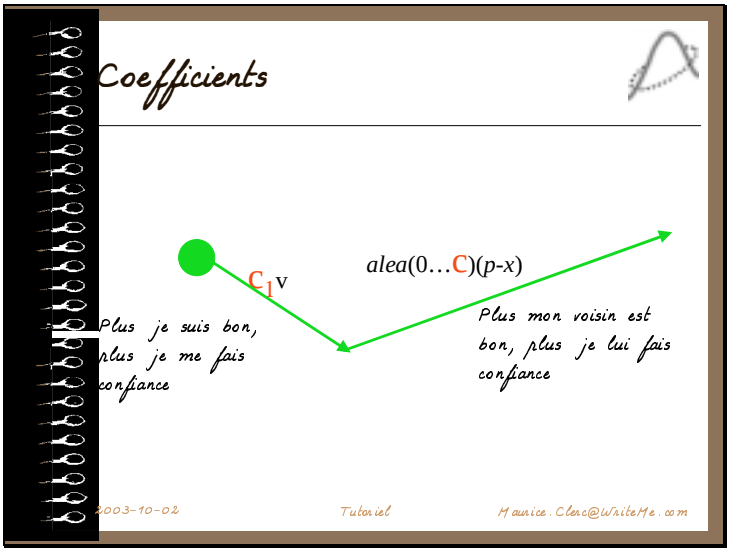
Diapositive 7



Les méthodes adaptatives représentent une évolution (c'est le cas de le dire) intéressante. Comme vous l'avez remarqué, l'OEP ne nécessite pas d'autorité centrale, il s'agit plutôt d'une anarchie (au sens premier du terme), qui s'auto-organise. Sauf dans quelques rares cas, d'ailleurs peu concluants, les méthodes adaptatives essaient de préserver cette particularité.

Examinons d'abord les principes de que l'on pourrait appeler les *adaptations faibles,* qui conservent entièrement les équations de mouvement et cherchent seulement à modifier leurs paramètres au cours du temps.

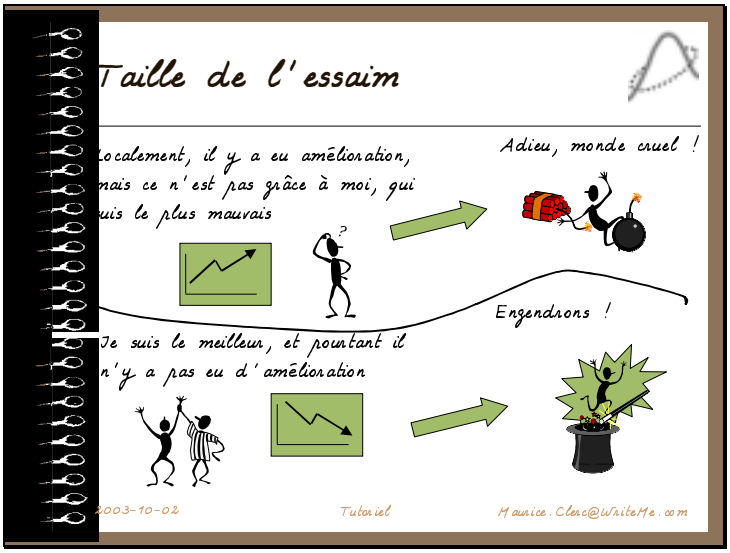
Diapositive 8



La plupart de ces méthodes jouent seulement sur les coefficients de confiance, soient, assez arbitrairement, selon une loi prédéfinie d'évolution au cours du temps (typiquement, le coefficient c_1 décroissant au cours du temps), soit, plus logiquement, selon un schéma-type consistant à dire que plus quelqu'un est bon, plus on lui fait confiance.

Néanmoins, même si de tels schémas peuvent être transcrits sous forme de règles floues, ils nécessitent eux-mêmes un paramétrage. L'algorithme peut être plus efficace, mais la charge de l'utilisateur n'a pas diminué, on pourrait presque dire « au contraire ».

Diapositive 9

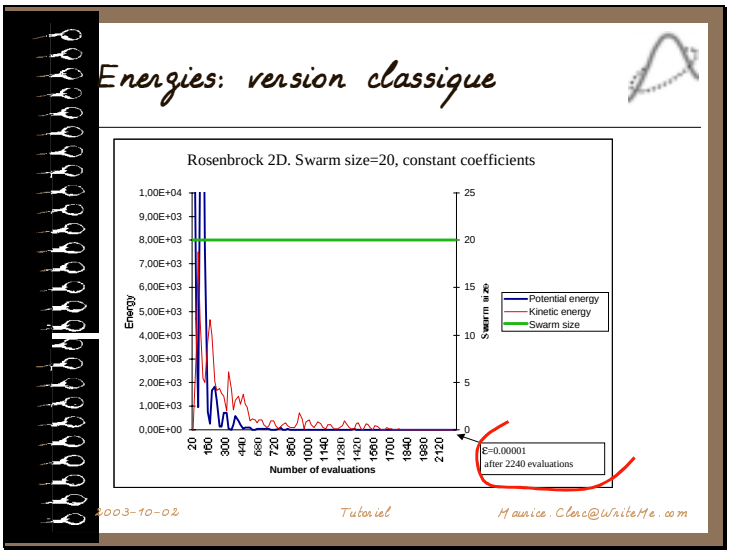


Les *adaptations fortes* modifient également la structure même de l'essaim.
Jusqu'ici la taille de l'essaim était constante, le principe étant la coopération totale sans sélection, sans « élimination des faibles », l'idée (confirmée par les tests), étant que les faibles d'aujourd'hui peuvent être les forts de demain.
L'inconvénient est que, là encore, c'est à l'utilisateur de définir cette taille d'essaim.
L'application de règles de sélection et de génération permet de s'affranchir de cette

contrainte. Avec de telles règles, typiquement, on peut parfaitement démarrer avec un essaim réduit à une seule particule.

Le but du jeu, évidemment, est de trouver des règles qui ne nécessitent pas elles-mêmes de nouveaux paramètres. En pratique, on essaie de n'utiliser que des règles qualitatives et des relations d'ordre.

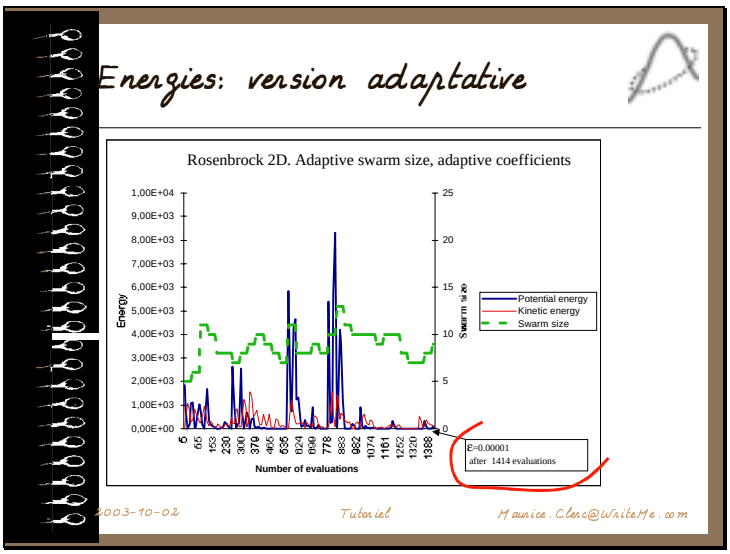
Diapositive 10



Examinons un exemple très classique, pour illustrer l'influence de ces adaptations. En attribuant une masse 1 aux particules, on peut calculer leur énergie cinétique et leur énergie potentielle (écart de position par rapport à une valeur de référence, le minimum connu zéro dans cet exemple).

Ici la taille de l'essaim est de 20. Les particules convergeant toutes plus ou moins vers le minimum, les énergies totales décroissent assez rapidement. Sur la fin, l'énergie cinétique faible implique que les particules ont quelque difficulté à effectuer les derniers déplacements pour atteindre la précision demandée (10^{-5}, en l'occurrence).

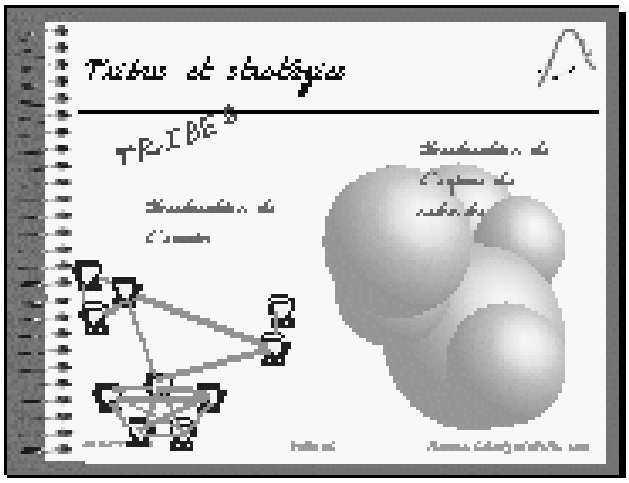
Diapositive 11



Avec une version adaptative, le processus est sensiblement différent. En particulier, de temps en temps, de nouvelles particules sont générées (ici complètement au hasard), ce qui augmente l'énergie cinétique et, partant, ce que l'on pourrait appeler le « pouvoir exploratoire » de l'essaim.

L'exemple montré ici est particulièrement favorable, en ce sens que, en plus d'avoir moins de paramètres à définir, on obtient plus rapidement une solution. C'est assez souvent le cas, mais pas toujours. Il est évident que, en effectuant manuellement des centaines d'essais, on peut souvent trouver un jeu de paramètres qui rende la version non adaptative plus performante pour un problème donné, mais, précisément, le but est d'éviter ces centaines d'essais préalables.

Diapositive 12



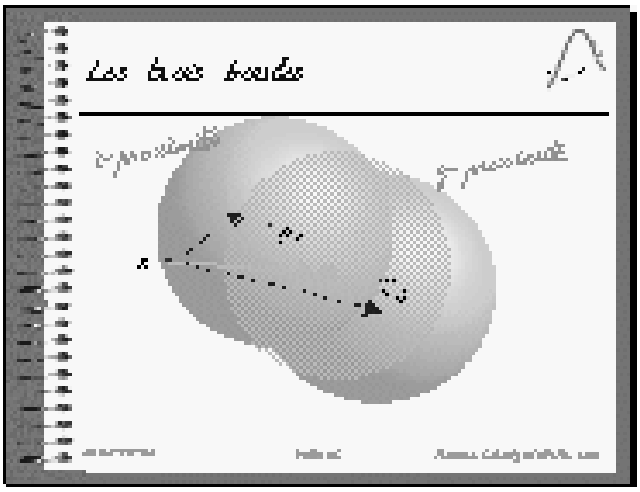
Juste quelques mots au sujet du programme Tribes, qui justifierait à lui seul tout un exposé. C'est une OEP avec adaptation forte. A chaque instant, l'essaim est structuré en tribus. Dans chaque tribu les relations d'information forment un graphe complet. Par contre, d'une tribu à une autre, il n'existe que deux liens d'information, un dans chaque sens.

Une tribu qui « réussit » peut se permettre d'éliminer le plus mauvais de ses membres. Les tribus qui régressent tentent de se sauver un générant chacune au loin une particule, tout en gardant le contact. Ces nouvelles particules forment une nouvelle tribu.

Par ailleurs, chaque particule utilise une stratégie de mouvement qui dépend de son passé proche. Il est en effet intuitivement évident qu'une « bonne » particule, qui n'arrête pas de progresser, ne devrait pas choisir son prochain mouvement comme le fera une « mauvaise » particule, ce qui est pourtant le cas dans les OEP classiques.

Dans la version la plus complète, la génération d'une nouvelle particule se fait selon une « génération aléatoire guidée », de préférence dans une région de l'espace de recherche peu explorée.

Diapositive 13



Naturellement Tribes peut utiliser les proximités à base d'hypersphères, mais comporte aussi différentes options, comme des distributions gaussiennes ou des variantes de la méthode des pivots.

Le schéma des trois boules illustre simplement le fait que si l'on prend deux points au hasard (uniformément) dans deux hypersphères et que l'on fait une combinaison linéaire des vecteurs correspondants, l'ensemble des résultats possibles est aussi une hypersphère. Cependant la distribution résultante n'est *pas* uniforme, ce qui, intuitivement, est plutôt favorable.

En pratique, on utilise Tribes en démarrant avec une seule particule et sans donner aucun paramètre, puisque les variations de la taille de l'essaim et la définition des liens d'information sont prises en charge automatiquement.

Encore une fois, pour un problème donné, les résultats sont souvent un peu moins bons que ceux obtenus avec des versions d'OEP aux paramètres finement ajustés, mais, au moins, l'utilisateur n'a rien d'autre à faire que de définir la fonction à minimiser, l'espace de recherche et la précision souhaitée.