

L'optimisation par essaim particulaire

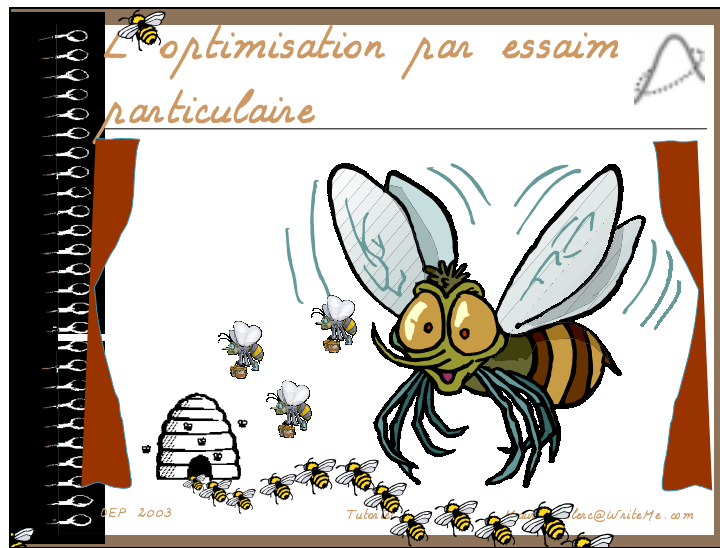
Tutoriel pour OEP 2003

Maurice Clerc
France Télécom R&D

`courriel : Maurice.Clerc@WriteMe.com`

Partie I : métaphores, premières formulations, paramétrages

Diapositive 1

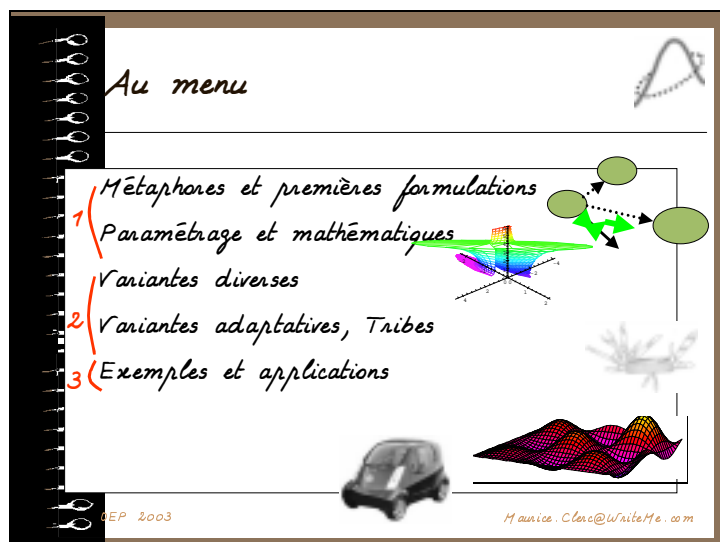


L'OEP est relativement récente, puisque les premiers articles la concernant datent de 1995, l'appellation d'origine étant, en anglais, *Particle Swarm Optimization*.

Il s'agit d'une méthode faisant appel à une population d'agents, appelés ici particules, mais, comme on le verra, par rapport à d'autres heuristiques de la même famille, elle présente quelques particularités intéressantes, entre autres la notion que l'efficacité est due à la collaboration plutôt qu'à la compétition.

On fait parfois la confusion entre l'OEP et le filtrage particulaire (cas particulier des Systèmes Particulaires Interactifs) qui, comme chacun sait (!) est une méthode optimale d'estimation de processus non linéaires observés au travers de capteurs. Il y a des ressemblances (population de particules, déplacements aléatoires), même si le filtrage est plus spécifique. Néanmoins, les deux domaines de recherche s'ignorent encore largement. Ce séminaire sera peut-être l'occasion d'un rapprochement fructueux.

Diapositive 2

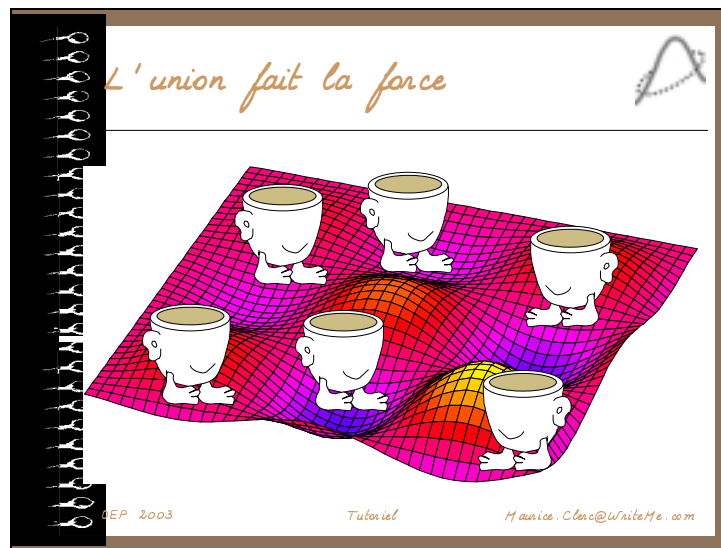


Cette présentation comporte trois grandes parties. Après chacune d'entre elles, vous pourrez poser quelques questions auxquelles j'essaierai de donner une réponse, même si, comme il est probable, ce sera simplement « ce point sera évoqué un peu plus tard ».

La première partie traite des principes de base de la méthode et de sa formalisation. Dans la seconde, j'essaierai de présenter un panorama, forcément incomplet, des principales variantes, en particulier les variantes adaptatives. Je dirai quelques mots du programme Tribes, qui sera utilisé dans l'atelier de cet après-midi.

Les exemples concernent des problèmes plus ou moins artificiels, alors que les applications s'intéressent à des problèmes réels. Comme nous le verrons, la force de l'OEP est que l'on peut traiter des applications très variées d'une manière uniforme.

Diapositive 3



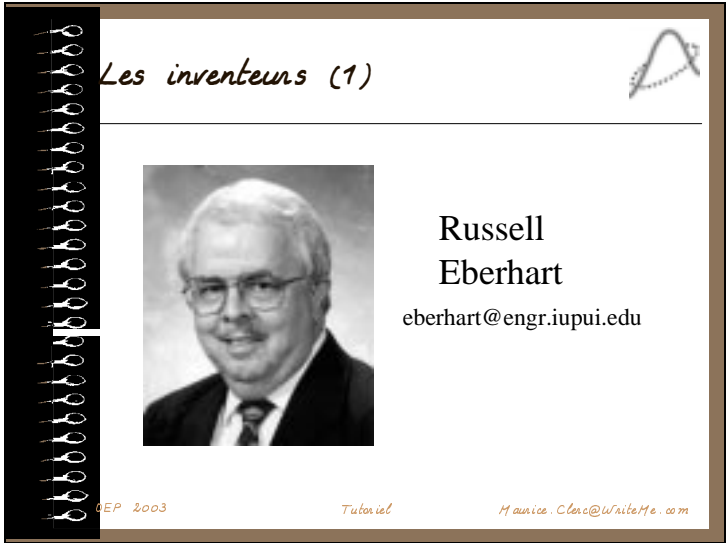
Si vous devez effectuer une certaine tâche, comme explorer un espace de recherche et que, pour ce faire, vous n'avez à votre disposition qu'un agent fort peu intelligent, c'est pratiquement sans espoir. Maintenant, si vous en avez plusieurs, on peut envisager différentes stratégies pour qu'une telle équipe d'imbéciles se comporte, globalement, de manière astucieuse.

Avec l'OEP les règles à appliquer sont très simples et, de plus, il n'est même pas nécessaire d'avoir un contremaître pour diriger l'équipe.


Bien sûr, il y a de nombreuses autres heuristiques utilisant des populations d'individus (algorithmes génétiques, colonies de fourmis, etc.), toutes fondées sur des propriétés d'auto-organisation, mais ayant chacune son domaine de compétence. L'exposé de Johann Dréo et Patrick Siarry précisera ces aspects.

Notons d'emblée, mais nous en reparlerons, que l'OEP, quant à elle, est particulièrement efficace dans des espaces de recherche continus ou mixtes (certaines variables étant discrètes). Toutes ces méthodes ne sont d'ailleurs peut-être que des facettes partielles d'une même métaheuristique qui reste à découvrir. On peut déjà, dans certains cas, établir des correspondances formelles, mais surtout, envisager des enrichissements réciproques, par exemple autour des notions de mémoire ou de probabilités conditionnelles.

Diapositive 4



Les inventeurs (1)



Russell
Eberhart
eberhart@engr.iupui.edu

SEP 2003 Tutoriel Maurice.Clerc@unite.fr.com


En fait, l'OEP est née presque par hasard. Au départ, les inventeurs souhaitaient simuler des interactions sociales, essentiellement en s'appuyant sur la technique des algorithmes culturels adaptatifs de Robert Reynolds et Robert Axelrod. Cette technique est souvent décrite comme une sorte d'algorithme évolutionnaire, avec une population d'individus (les agents), dans laquelle, à chaque pas de temps, les « meilleurs » (selon un critère prédéfini) sont plus ou moins imités par les autres. Un aspect essentiel, qui la différencie, par exemple, des algorithmes génétiques classiques, est l'existence d'une mémoire, à laquelle ne contribuent que les meilleurs éléments. Le modèle est censé expliquer et reproduire certains comportements sociaux, mais s'est révélé un peu simpliste.

Par contre, convenablement modifié, il peut résoudre des problèmes d'optimisation, car le processus fait apparaître des individus de plus en plus performants : c'est ainsi que l'OEP est née, sous le nom, donc, de *Particle Swarm Optimization (PSO)*, grâce à la collaboration de deux personnes de profils très différents, Russ Eberhart et Jim Kennedy.

En particulier, comme on le verra, il s'est d'emblée révélé très efficace pour l'éducation (ou l'apprentissage) de réseaux neuronaux, ce que Russ, en tant qu'ingénieur et enseignant à l'Université Purdue à Indianapolis, s'est empressé de mettre en application.

Diapositive 5

Les inventeurs (2)



James
Kennedy

Kennedy_Jim@bls.gov

OEP 2003 Tutoriel H. Anice, C. Laro@WriteMe.com

Jim Kennedy est psychologue social et statisticien, travaillant pour le compte du gouvernement à Washington DC.

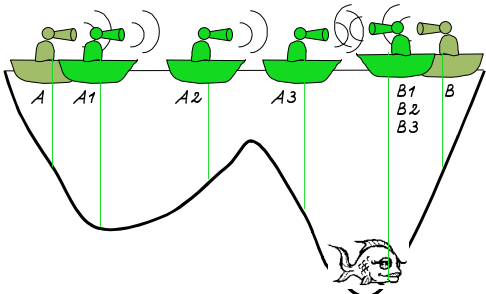
En 2001, Russ et lui ont publié un livre, *Swarm Intelligence*, qui, comme vous le voyez, a l'air de passionner ses enfants, Bonnie et Jamey.

Ce livre, en fait essentiellement écrit par Jim, comporte un peu plus de 500 pages, dont environ 130 (la seconde partie) sont consacrées à l'OEP proprement dite. Dans toute la première partie et également dans les conclusions, sont présentées des idées générales sur l'optimisation, sur ses relations avec les comportements sociaux, sur les techniques évolutionnaires, etc. Elles n'ont pas toutes contribué à l'OEP, loin de là, mais sont toutes intéressantes.

On notera l'annexe, qui résume des méthodes statistiques élémentaires qui devraient être systématiquement utilisées lorsque l'on étudie le comportement d'un algorithme comportant une part d'aléatoire. Ce n'est pas toujours le cas et l'on voit régulièrement des articles préconisant telle ou telle méthode d'optimisation, sur la seule base de quelques tests bien insuffisants.

Diapositive 6

Exemple de coopération



OEP 2003 Tutoriel H. Anice, C. Laro@WriteMe.com

Pour illustrer ce que coopération signifie en OEP, voici un exemple très simple. J'utilise ici volontairement une autre métaphore que celle des insectes sociaux, pour insister sur le fait que ce sont les principes qui comptent et non pas leur habillage, biologique ou autre.

Comme d'habitude, le gros poisson est au plus profond de l'étang. A chaque pas de temps, chaque pêcheur indique à son collègue l'endroit le plus profond qu'il a déjà trouvé.

Au début, pas de différence, et chacun suit sa propre route indépendamment.

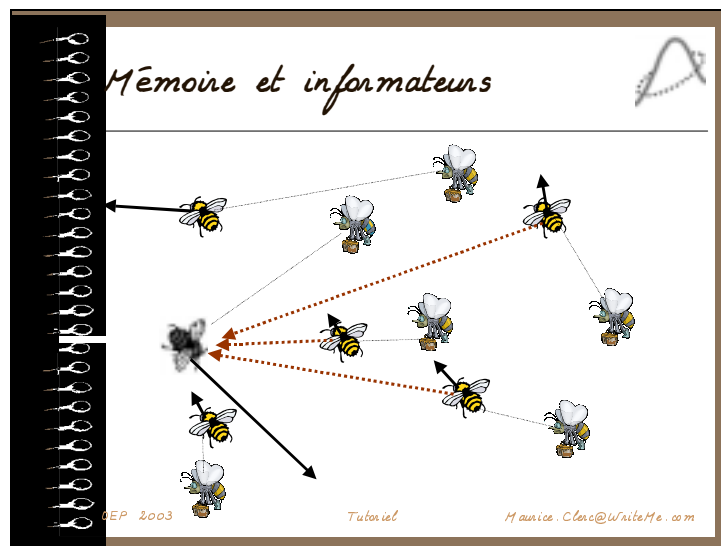
Ensuite, le pêcheur B a trouvé un meilleur emplacement, alors le pêcheur A tend à aller vers lui, assez vite. Ce faisant, il détériore sa position. Dès lors, il a un compromis à faire entre la meilleure position qu'il a trouvée jusque-là et celle trouvée par B. Il continue d'aller vers B, mais plus lentement, et de même le pas de temps suivant

Le point important est que, ce faisant, il échappe au minimum local.

Naturellement, c'est un exemple un peu caricatural, mais on y trouve les notions essentielles : position, vitesse (ou, plus précisément, un opérateur modificateur de position), échange d'informations, mémoire (très limitée), capacité rudimentaire à combiner les informations pour prendre une décision.

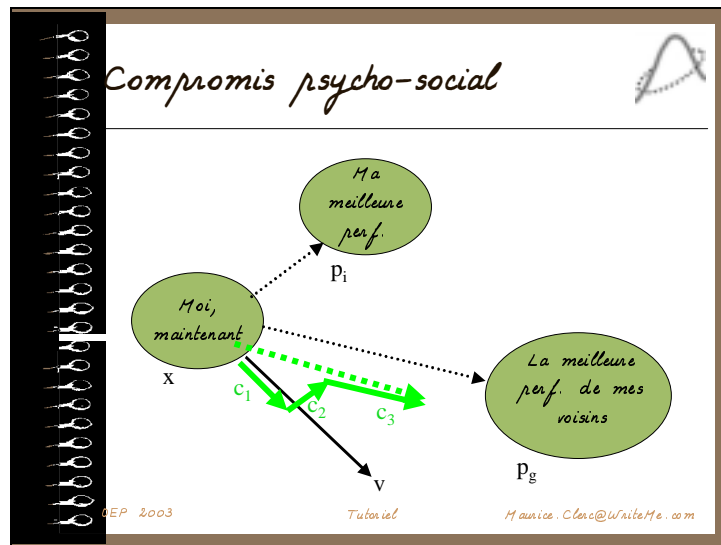
L'idée est que tout cela doit rester simple.

Diapositive 7



Faisons maintenant une première présentation informelle du modèle. On considère, dans l'espace de recherche, un essaim de particules, ici cinq. Chaque particule est en train de bouger, c'est-à-dire qu'elle a une vitesse. Egalement, chaque particule a une petite mémoire, lui permettant de se souvenir de sa meilleure performance, en position et en valeur. Enfin, chaque particule dispose d'un groupe d'informatrices, historiquement appelé son voisinage.

Diapositive 8



A partir de quelques informations dont elle dispose, une particule doit décider de son prochain mouvement, c'est-à-dire décider de sa nouvelle vitesse.

Pour ce faire, elle combine linéairement trois informations :

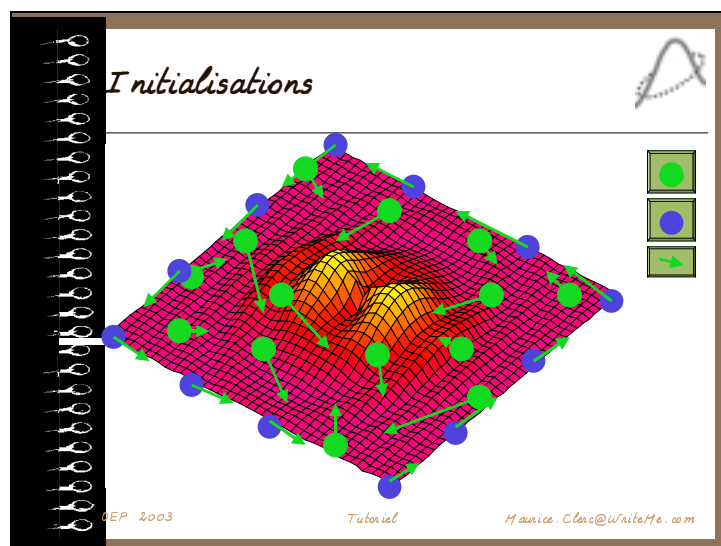
- sa vitesse actuelle
- sa meilleure performance
- la meilleure performance de ses voisines (ses informatrices).

à l'aide de trois paramètres parfois appelés *coefficients de confiance*, qui pondèrent trois tendances :

- tendance à suivre sa propre voie
- tendance conservatrice (revenir sur ses pas)
- tendance « panurgienne » (suivre le meilleur voisin)

Reprenons tout le processus de manière plus formelle.

Diapositive 9

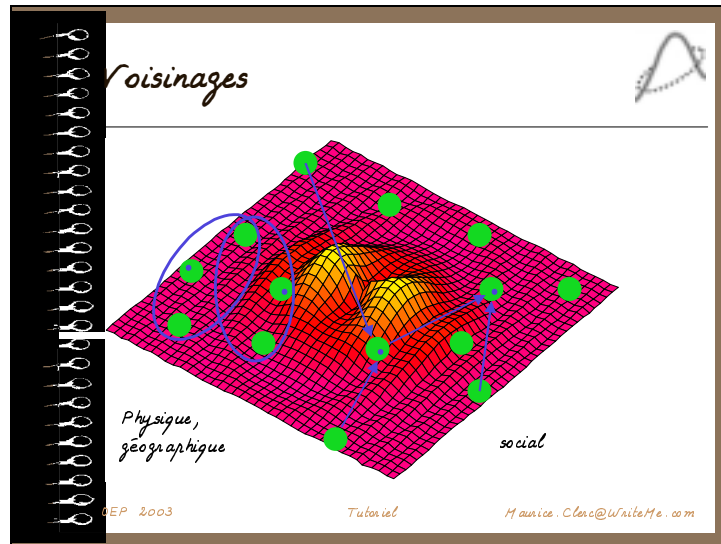


Première étape, on initialise l'essaim de particules dans l'espace de recherche. On peut soit le faire de manière aléatoire (ce qui n'est pas forcément si simple selon la forme de l'espace de recherche), soit de manière régulière, en particulier sur la frontière. En général, le plus efficace est une combinaison des deux.

Deuxième étape, on initialise également les vitesses, de manière aléatoire.

On suppose pour l'instant que l'essaim est de taille constante. Il s'agit donc également d'un paramètre du modèle, à définir.

Diapositive 10

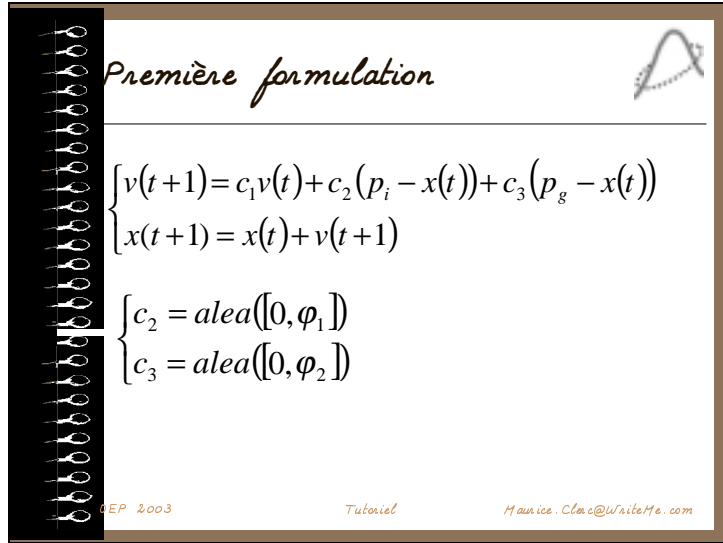


Ensuite, pour chaque particule, on définit son voisinage. Il y a deux grandes méthodes. Soit un voisinage géographique, qui doit être recalculé à chaque pas de temps, et qui suppose l'existence d'une distance dans l'espace de recherche, soit un voisinage « social » défini une fois pour toutes.

C'est le voisinage social qui est le plus utilisé, pour plusieurs raisons :

- il est plus simple à programmer,
- il est moins coûteux en temps calcul,
- de toutes façons, en cas de convergence, un voisinage social tend à devenir un voisinage géographique.

Diapositive 11



Première formulation

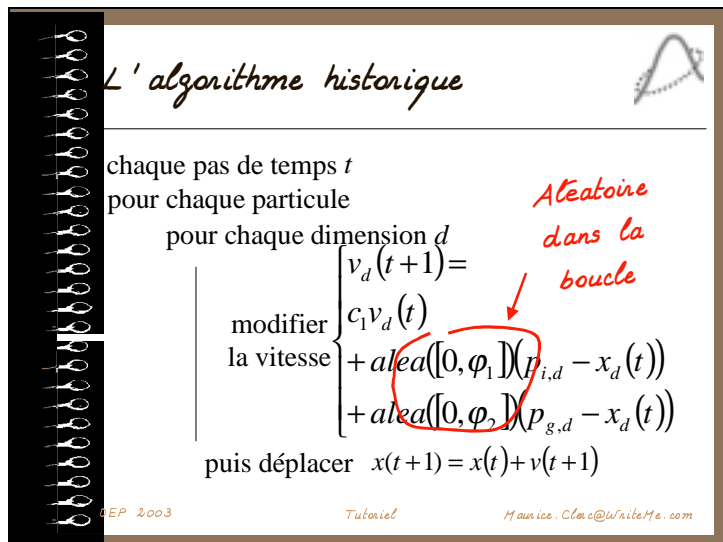
$$\begin{cases} v(t+1) = c_1 v(t) + c_2 (p_i - x(t)) + c_3 (p_g - x(t)) \\ x(t+1) = x(t) + v(t+1) \end{cases}$$
$$\begin{cases} c_2 = \text{alea}([0, \varphi_1]) \\ c_3 = \text{alea}([0, \varphi_2]) \end{cases}$$

SEP 2003 Tutoriel Maurice.Clerc@univ-lyon1.fr

Les équations complètes du mouvement d'une particule peuvent alors s'écrire ainsi. La nouvelle vitesse est calculée comme combinaison linéaire de trois éléments, comme vu auparavant, puis appliquée à la position courante pour donner la nouvelle position.

L'efficacité de la méthode tient en particulier au caractère aléatoire des deux derniers coefficients. Voyons tout de suite le premier algorithme qui, historiquement, résulte de cette formulation.

Diapositive 12



L'algorithme historique

chaque pas de temps t
pour chaque particule
pour chaque dimension d

modifier
la vitesse

$$v_d(t+1) =$$

$$\begin{aligned} &c_1 v_d(t) \\ &+ \text{alea}([0, \varphi_1]) (p_{i,d} - x_d(t)) \\ &+ \text{alea}([0, \varphi_2]) (p_{g,d} - x_d(t)) \end{aligned}$$

puis déplacer $x(t+1) = x(t) + v(t+1)$

Aléatoire dans la boucle

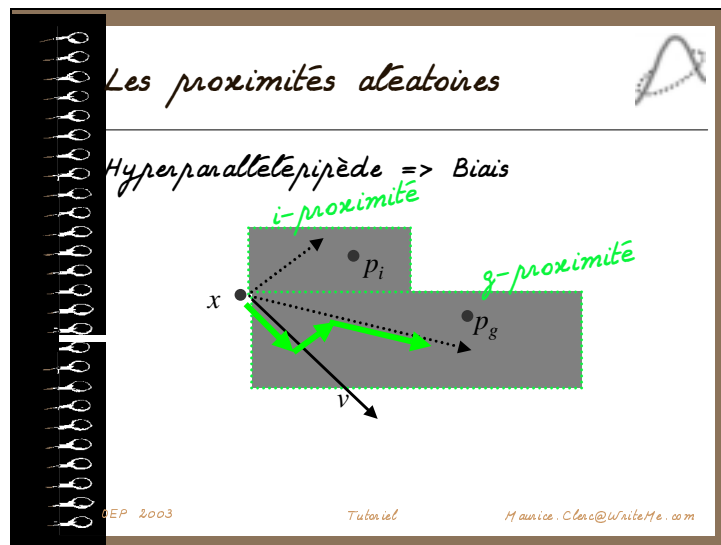
SEP 2003 Tutoriel Maurice.Clerc@univ-lyon1.fr

En définitive, le pseudo code de cet algorithme ne comprend que quelques lignes. Le programme complet inclut également la recherche du meilleur voisin, à chaque pas de temps, mais reste quand même très court, puisqu'il s'agit d'une simple boucle de parcours d'un voisinage.

Egalement, toujours à chaque pas de temps, il faut mémoriser, pour chaque particule, sa meilleure performance.

On notera ici un point important, à savoir que les deux coefficients aléatoires sont recalculés *pour chaque dimension*. Ceci va nous donner l'interprétation géométrique du choix de la nouvelle vitesse.

Diapositive 13

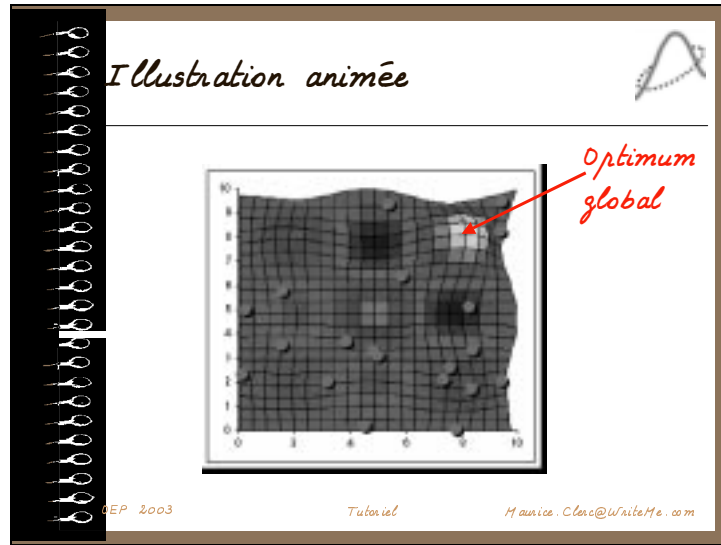


Le premier rectangle (plus précisément hyperparallélépipède), appelé i-proximité, nous donne toutes les positions possibles pour le terme aléatoire correspondant au retour à proximité de la meilleure performance de la particule. L'une d'elles est choisie au hasard, selon une distribution uniforme.

De la même manière, une autre position est choisie dans un hyperparallélépipède (g-proximité) autour de p_g , meilleure performance des voisines.

Telle quelle, la méthode est déjà efficace, mais on voit tout de suite une amélioration possible. L'utilisation d'hyperparallélépipèdes implique en effet un biais, puisque le domaine des choix possibles dépend du système de coordonnées. De plus, il s'agit d'une forme assez arbitraire, qui résulte seulement de la facilité de codage d'une distribution uniforme sur un intervalle. Nous verrons dans les variantes comment on peut améliorer cet aspect.

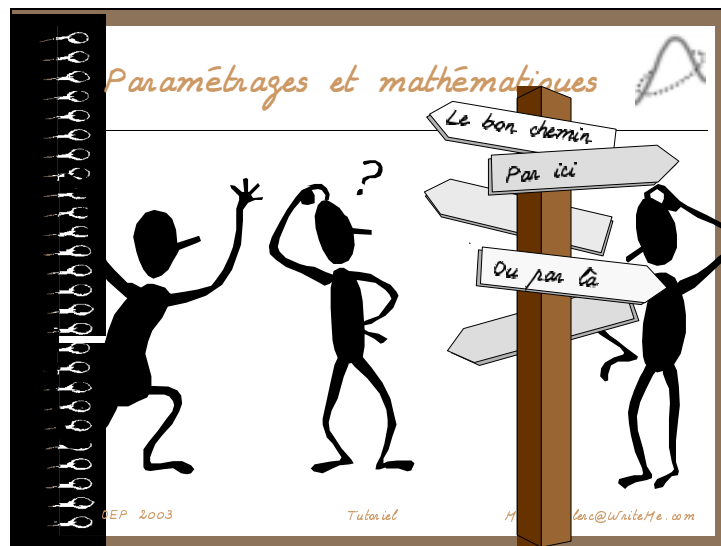
Diapositive 14



Quoiqu'il en soit, la méthode originelle telle qu'elle vient d'être présentée fonctionne déjà de façon très satisfaisante et est encore couramment utilisée.

Pour terminer cet aperçu du modèle historique, voici une petite animation, qui illustre non seulement comment l'essaim trouve l'optimum global, mais aussi comment il se décompose temporairement en sous-essaims sur des minimums locaux, particularité qui peut être mise à profit dans certaines applications.

Diapositive 15

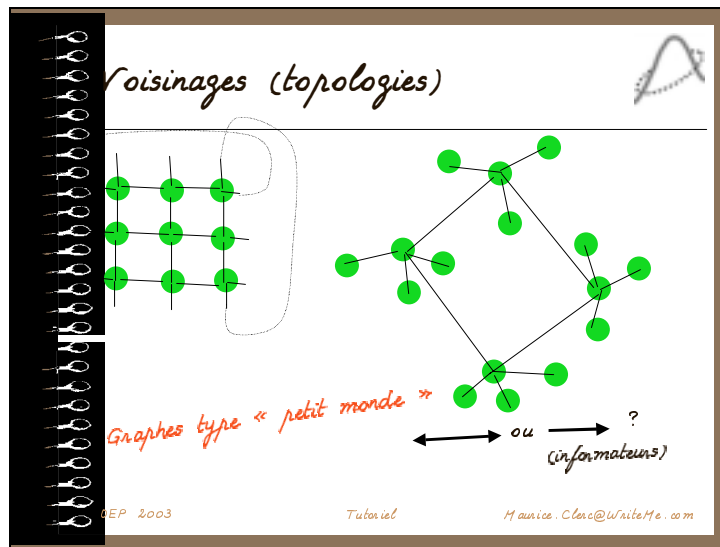


La version « historique », comme on l'a vu, comporte un certain nombre de paramètres que l'utilisateur est contraint de définir a priori :

- la taille de l'essaim (le nombre de particules)
- la définition des voisinages
- les coefficients de « confiance » de la pondération linéaire.

Dans un premier temps, des règles empiriques ont été déterminées. En particulier, on s'est aperçu qu'il était nécessaire de limiter la vitesse à une valeur V_{max} (un nouveau paramètre, donc), faute de quoi, dans certains cas, le système pouvait « exploser ».

Diapositive 16

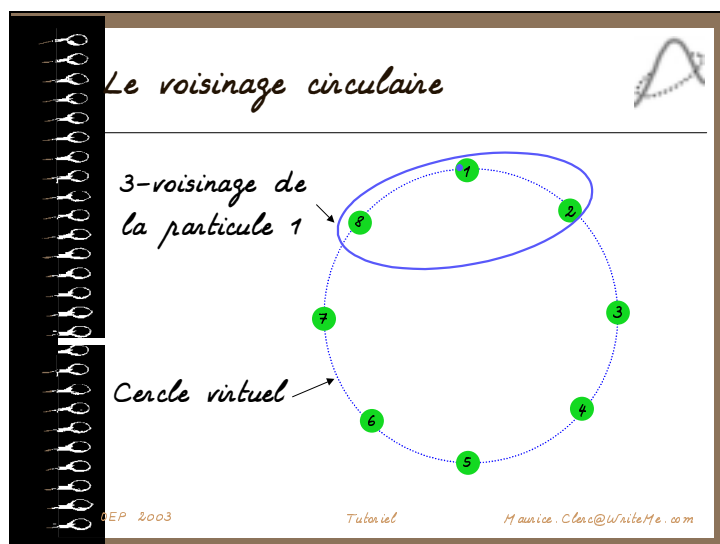


En ce qui concerne les voisinages, on peut envisager différentes topologies pour les relations entre particules, par exemple un graphe régulier de type quadrillage ou tore. Une voie de recherche actuelle concerne les graphes de type « petit monde », c'est-à-dire ayant les caractéristiques suivantes :

- un rapport nombre de nœuds/nombre d'arcs élevé
 - un diamètre et une longueur caractéristique faibles
 - un coefficient d'agglomération élevé
- car on les retrouve souvent (relations sociales, liens Internet, etc.)

Si les liens sont symétriques, on parlera effectivement de voisinage, sinon, on tendra à préférer le terme d'informaticiens.

Diapositive 17



Le voisinage le plus utilisé est celui dit « circulaire ».

Les particules sont numérotées, disposées virtuellement sur un cercle et la définition des voisinages se fait une fois pour toutes en fonction des numéros.

Si les voisinages sont de taille k (y compris la particule elle-même), le graphe correspondant est régulier de degré $k-1$.

Le voisinage de taille 3 de la particule 1 est (8,1,2), celui de taille 4 serait soit (7,8,1,2) soit (8,1,2,3) etc.

Diapositive 18

Paramètres (constriction)

$$\begin{cases} v(t+1) = v(t) + \varphi(p - x(t)) \\ x(t+1) = x(t) + v(t+1) \end{cases} \quad M = \begin{bmatrix} 1 & \varphi \\ -1 & 1 - \varphi \end{bmatrix}$$

$$y(t) = p - x(t)$$

système canonique

$$\begin{cases} v(t+1) = v(t) + \varphi y(t) \\ y(t+1) = -v(t) + (1 - \varphi)y(t) \end{cases}$$

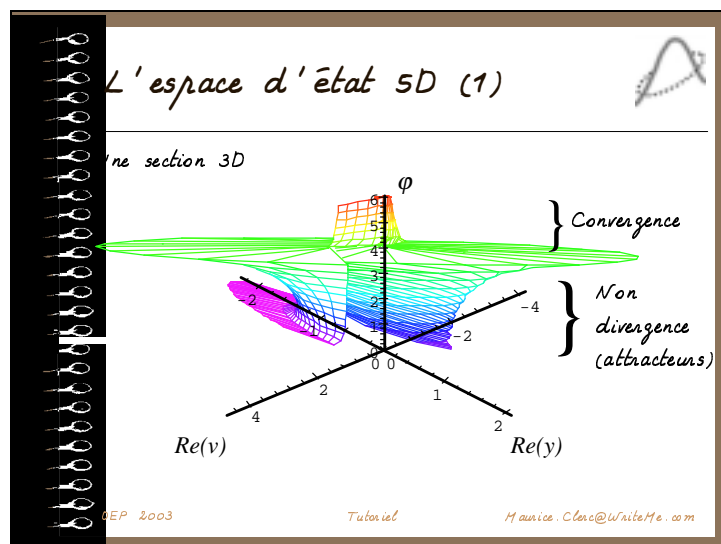
$$\begin{cases} e_1 = 1 - \frac{\varphi}{2} + \frac{\sqrt{\varphi^2 - 4\varphi}}{2} \\ e_2 = 1 - \frac{\varphi}{2} - \frac{\sqrt{\varphi^2 - 4\varphi}}{2} \end{cases}$$

SEP 2003 Tutoriel Maurice.Clerc@unite-he.com

Ce n'est pas ici le lieu de faire une présentation mathématique détaillée d'autant que l'exposé de Ioan Cristian Trelea concernera précisément le choix des coefficients.

Le point le plus important à retenir est qu'il est possible de réécrire les systèmes d'équations sous une forme canonique, mettant en évidence une matrice correspondant à un système oscillatoire amorti, dont on examine les valeurs propres. L'étude peut se faire soit algébriquement, soit analytiquement (il s'agit en fait d'un système différentiel du second degré).

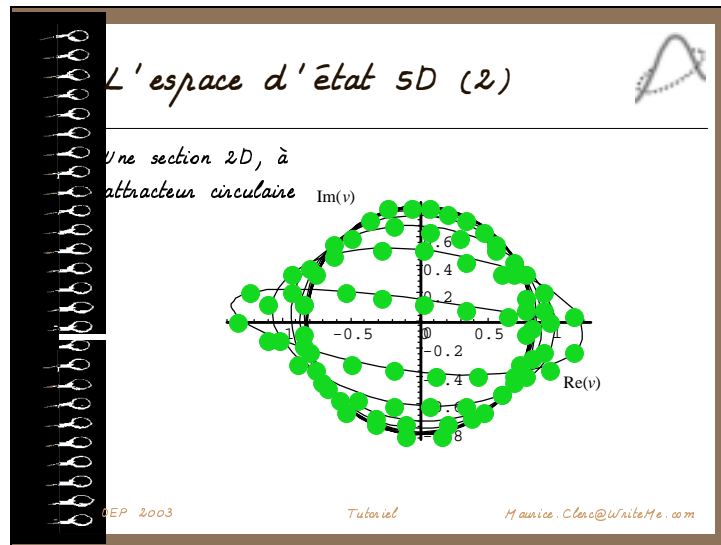
Diapositive 19



On peut alors représenter la trajectoire d'une particule dans un espace d'état à deux dimensions complexes et une réelle, c'est-à-dire à cinq dimensions réelles.

En particulier, on peut mettre en évidence, selon les valeurs du coefficient φ , les zones de convergence et de non-divergence. Ce dernier terme signifie que la particule tend vers un attracteur non réduit à un point.

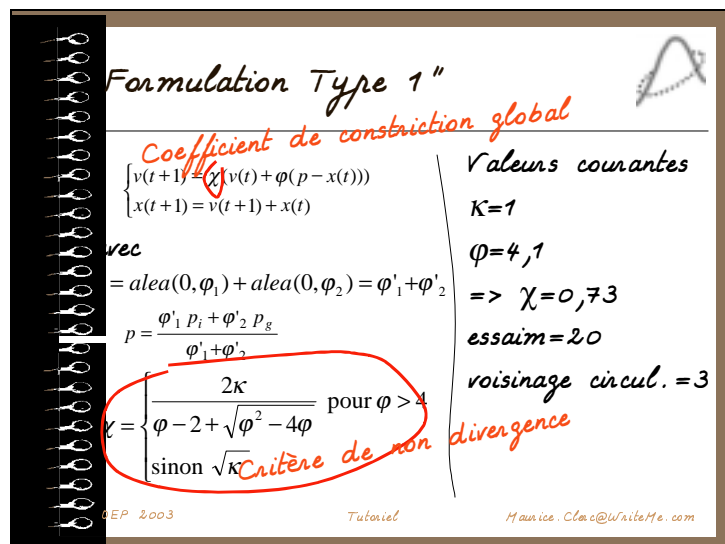
Diapositive 20



Une des sections de cet attracteur est très simple, puisqu'il s'agit d'un cercle, dont le rayon peut être calculé en fonction des valeurs propres de la matrice M vue plus haute.

D'où l'idée de manipuler ces valeurs propres pour que la taille du cercle corresponde à l'erreur admissible sur la solution.

Diapositive 21



Il y a une infinité de manières de le faire, qui peuvent être regroupées en quelques grands types de formulations. La plus simple est celle appelée Type 1". Elle consiste à multiplier la formule donnant la nouvelle vitesse par un coefficient dit de « constriction », essentiellement calculé à partir du paramètre φ .

On est alors certain que le système ne divergera pas, sans avoir à imposer une vitesse maximale.

Néanmoins, bien sûr, l'utilisateur doit toujours définir quatre paramètres. Les valeurs les plus couramment utilisées, trouvées empiriquement, sont une taille d'essaim de l'ordre de vingt, un voisinage circulaire de petite taille, et un coefficient ϕ juste au-dessus de la valeur critique 4.