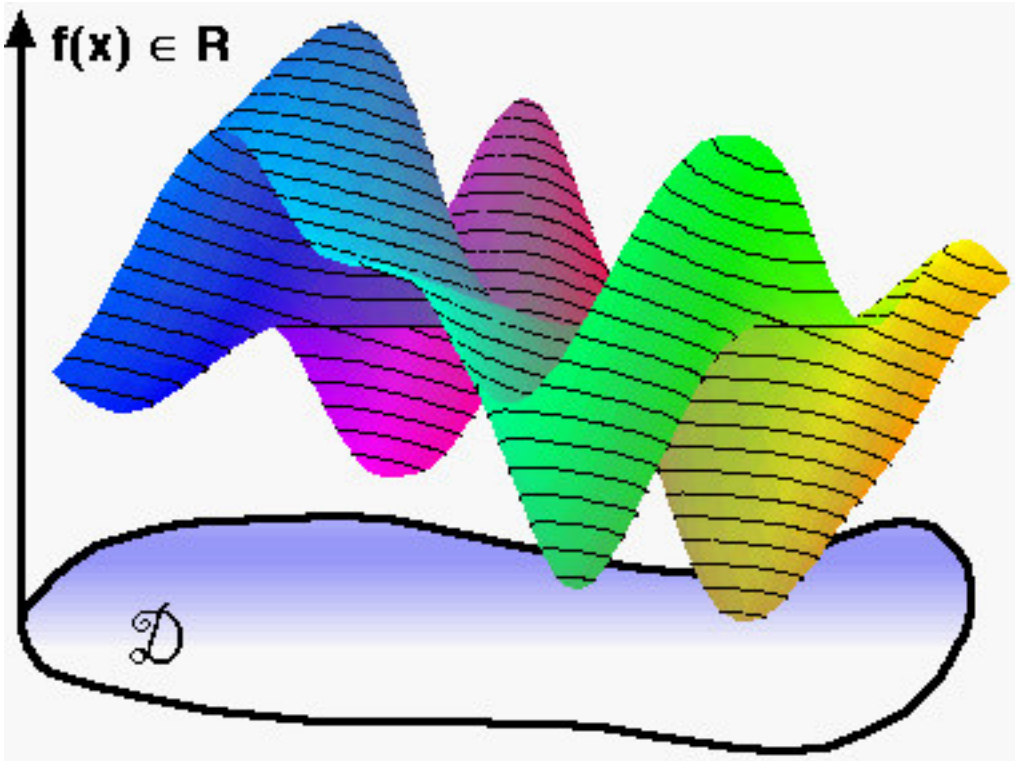
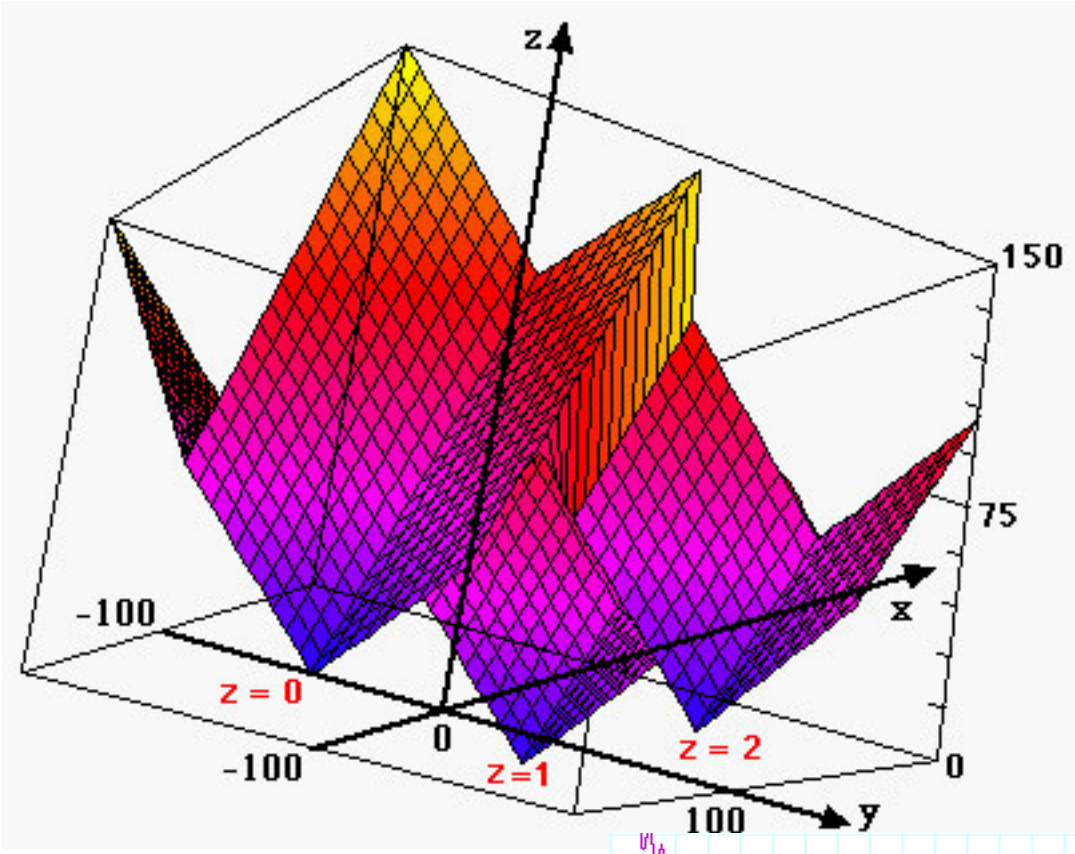
COMPARAISON ENTRE PSO ET AUTRES HEURISTIQUES D'OPTIMISATION AVEC OPERATEURS IMPLICITES

Critère de comparaison des méthodes

Moyenne sur 20 ou 100 « runs » aléatoires du nombre d ’évaluations de la fonction à minimiser, pour atteindre un seuil





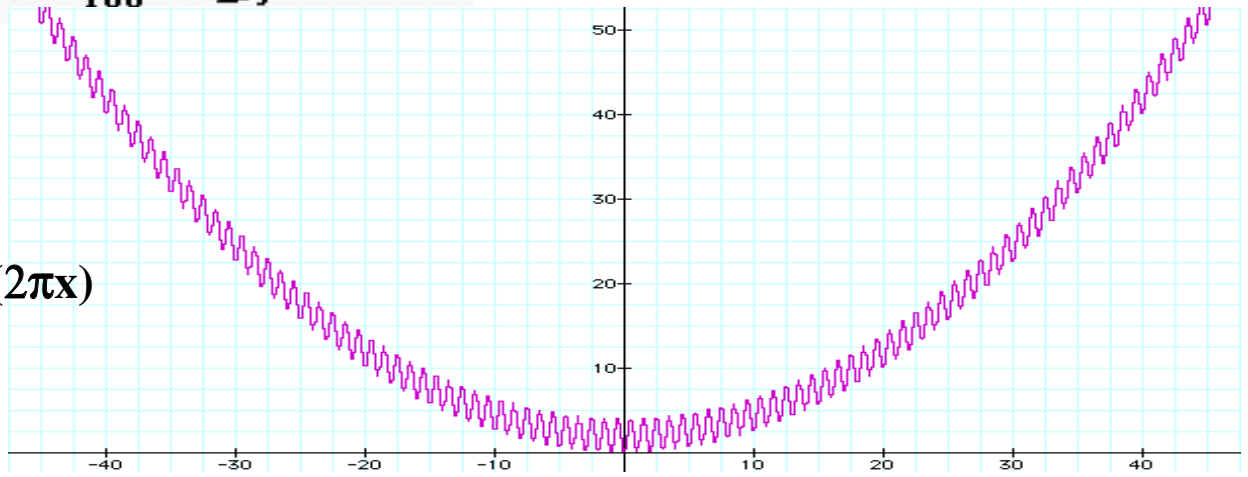
Fonctions Tests

Fonction "tripode"
avec 3 minimums attractifs
pour x, y ∈ [-100, 100]

f(x, y) = if y < 0 then |x| + |y + 50| else

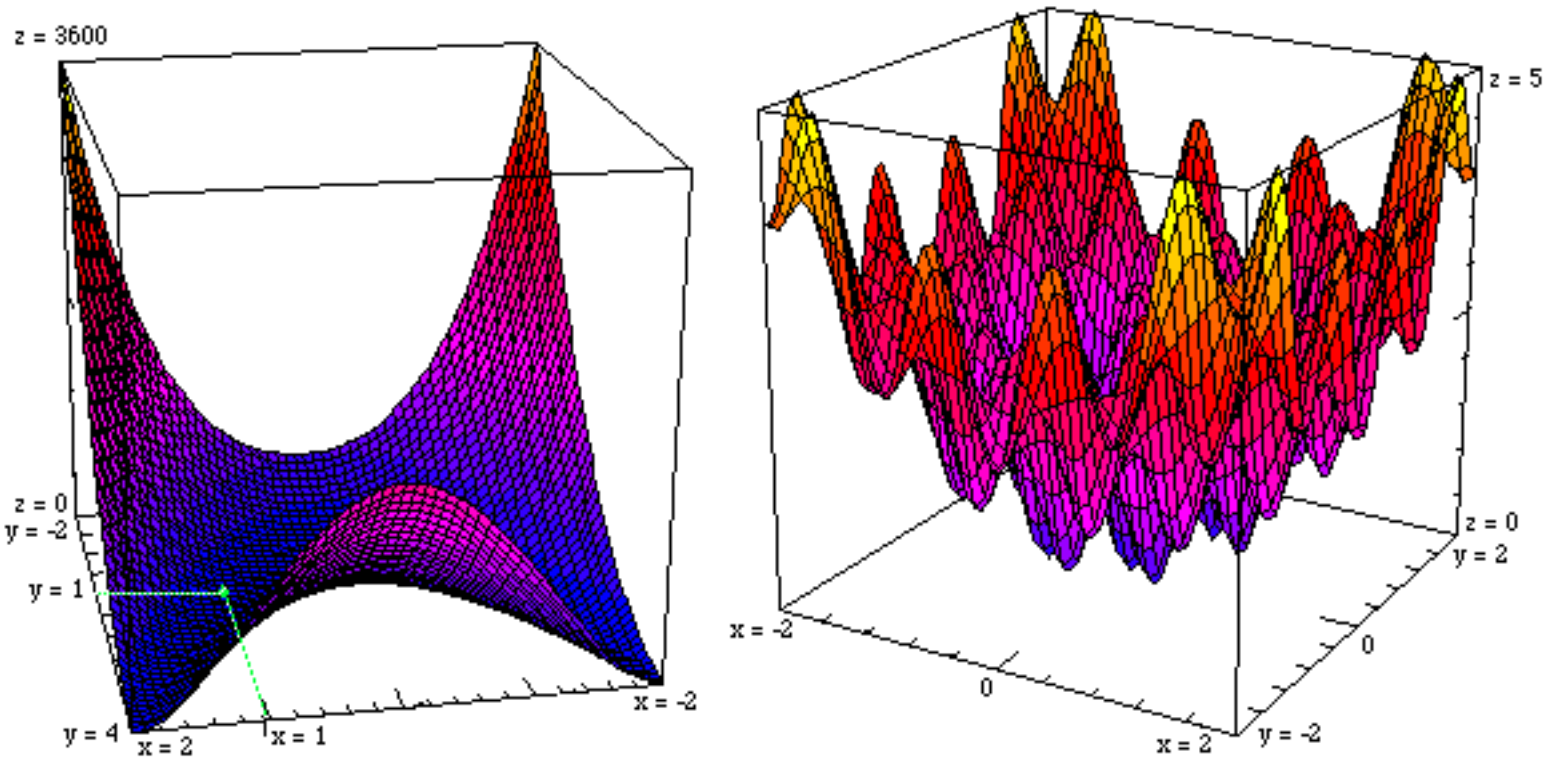
if x < 0 then 1 + |x + 50| + |y - 50|

else 2 + |x - 50| + |y - 50|



Fonction de Rastrigin
$F_R(x) = x^2 / 40 + 2 - 2\cos(2\pi x)$
en dimension 1

Fonctions tests



Fonction de Rosenbrock $F_R(x, y) = 100(x^2 - y)^2 + (1 - x)^2$

minimum 0 pour (1, 1), dans une vallée en forme de banane

Fonction de Griewank $F_G(x) = (x^2 + y^2) / d - \cos 2\pi x . \cos 2\pi y + 1$

(dimension 2 pour d = 2)

Méthodes de résolution

- **Méthodes combinatoires (exhaustives mais explosives)**
- **Méthodes constructives (backtracking)**
- **Méthodes locales**
  - **Descente de gradient** (si « bonnes » propriétés mathématiques)
  - **Hill-climbing** (exploration discrète des voisins)
  - **Nissen algorihm** (« ancêtre » maintenu durant une «ère»)
  - **Recuit simulé et recherche Tabu** (Kirkpatrick 1983)
- **Méthodes évolutionnaires (méthodes globales)**
  - **Evolutionary Programming** (Fogel 1966)
    sélection pour tournoi et mutations
  - **Standard genetic algorithm** (Holland 1975, Goldberg 1989)
    sélection par roulette biaisée, croisement, mutation
  - **Genetic Programming** (Koza 1992)
    codage d'arbres : expressions lisp
  - **Stratégies d'évolution ES(μ + λ) et ES(μ, λ)** (Schwefel 1973-90)
    codage réel, petite population
  - **Steady-state algorithm SSGA(μ, τ)** (Whitley, Kauth 1988)
    mise à jour régulière à chaque génération

Algorithmes classiques

Algorithme génétique standard GA(μ, p_c, p_m)

Codage binaire, grande population, Sélection, Croisement, Mutation

1) P_0 ensemble de μ chromosomes au hasard

2) Evaluation de la « fitness » ou « coût » f pour chaque x de P_t et tri P_t

3) Sélection de chromosomes en fonction de leur fitness

Crossover avec probabilité p_c
Mutation sur les enfants avec probabilité p_m
Remplacement des parents par les enfants (« generational mode »)

4) Si non fin alors t incrementé, goto 3

5) Fin

Stratégies d'évolution ES(μ + λ) and ES(μ, λ)

Codage réel, petite population, pas de sélection

3) Chaque individu x produit λ/μ enfants par opérateurs génétiques

P_{t+1} est les μ premiers parmi les λ enfants pour ES(μ, λ)

ou parmi parents + enfants pour ES(μ + λ)

Steady State Genetic Algorithm SSGA(μ, τ)

3) Mise à jour en prenant les τ meilleurs enfants et μ-τ meilleurs parents

Le "space partition algorithm" SPA

- Partition de l' espace en blocs réguliers
- Un individu est un bloc
- Evaluation d'un bloc par échantillon
- Génération : diviser les "bons" blocs et fusionner les "mauvais"
- Implémentation difficile, non performant

L' algorithme macroevolutionaire MGA (μ, ρ, τ)

- 1 Population de μ individus aléatoirement choisis.
- 2 W matrice de "connectivité" $w_{ij} = (f(x_i) - f(x_j)) / \| x_i - x_j \|$
- 3 A la génération t, si entrée $\Sigma_{j = 1..\mu} w_{ij}(t) < 0$ (x_i meilleur), alors $x_i(t+1) = x_i(t)$, sinon x_i remplacé avec probabilité τ par un nouvel individu aléatoire, (colonisation). Dans l' autre cas, (proba 1 - τ)

  $$x_i(t+1) = x_k(t) + \rho\lambda(x_k(t) - x_i(t))$$

  (crossover continu avec un x_k aléatoire et λ aléatoire dans [-1, 1])
- 4 Température τ fixe ou décroissante pour réduire l' exploration en faveur de l' exploitation. ($ex\tau(t) = 1 - t/ng_{max}$, ainsi τ abaissé à chaque génération).
- 5 aller en 2

- Stagnation, lenteur.

L' évolution différentielle DE (μ, p_c, χ)

1 P est aléatoirement initialisée

2 A chaque génération, pour l'individu $x = (x_1, ..., x_{dim})$ un rang de composante k et trois autres individus y, z et t aléatoirement choisis tous distincts,

mise à jour : $x'_j = t_j + \chi(y_j - z_j)$ pour j = k, mais aussi pour j <> k avec une probabilité $p_c = 0.5$, sinon $x'_j = x_j$.

(Sorte de "tetra-crossover" qui apporte une large perturbation)

3 Mise à jour des générations élitiste

Pas de tri, résultats assez bons

χ est recommandé est 1

DE homogène : chaque composante est modifiée avec probabilité p_c, le choix de k devenant inutile.

Le « self organizing migration algorithm »
SOMA(μ, r, step, mass, ml, χ)

1) P_0 ensemble de μ points aléatoires
2) Meilleur individu m = centre du premier groupe {x / d(x, m) < r}.
Du reste de P, est de nouveau choisi le meilleur et un autre groupe est formé....
3) Chaque particule x commence des sauts vers son leader avec une distance fixe « step »
Si mass = 1, cette migration s' arrête sur la première position derrière le leader, si mass = 2 à la seconde …
Nombre de sauts < ml
Mise à jour x' de x est la meilleure de la succesSsion.
4) si non(condition d' arrêt) aller en 2 (groupes redéfinis)

Expérimentation avec $step(t) = step_0.e^{-0.05t}$
et assez forte perturbation grâce à une probabilité χ = 0.5 de créer de nouveaux individus aléatoires durant la migration.

L' optimisation par essaim de particules PSO (μα, β, γ)

1 P_0 de μ solutions $x_i(0)$ aléatoires et chaque particule a une position $x_i(t)$ et une vitesse $v_i(t)$.
2 Chaque génération t, la "fitness" f de chaque $x_i(t)$ est calculée, et si g désigne la position du meilleur, pour chaque particule i, on regarde dans son voisinage pour chercher le meilleur n "leader du groupe", de telle sorte que les règles de mise à jour soient :

$$v_i(t+1) = \alpha v_i(t) + \beta(n - x_i(t)) + \gamma(g - x_i(t))$$
$$x_i(t+1) = x_i(t) + v_i(t+1)$$

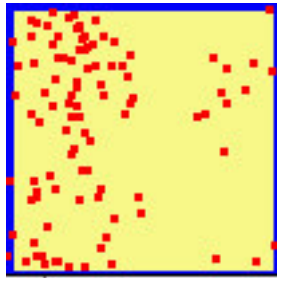
Il est possible de réduire le problème du choix des paramètres en prenant :

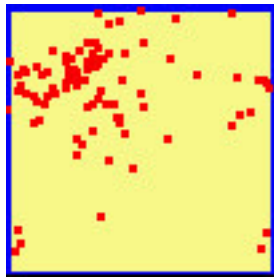
$$v_i(t+1) = \chi[\alpha v_i(t) + \beta\phi_1(n - x_i(t)) + \beta\phi_2(g - x_i(t))]$$

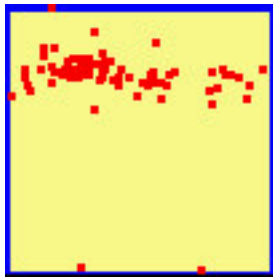
où χ = coefficient de constriction $\in$]0,1] et ϕ_1, ϕ_2 aléatoires

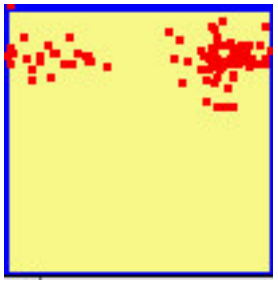
Population dans [-100, 100]

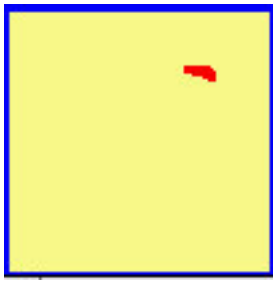
Générations 1 puis 4, 15, 30, 40 (minimisation de la fonction « tripod »)





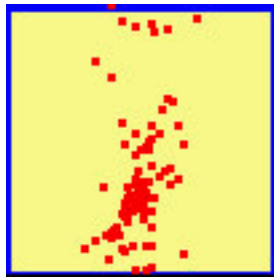


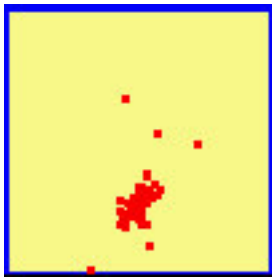


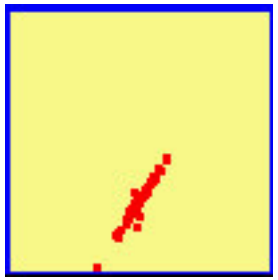


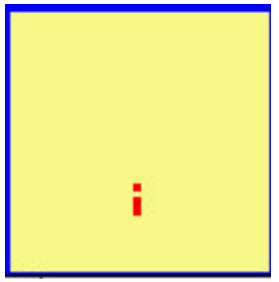
PSO nn = 5 voisins géométriquement définis

PSO avec nn = 5 voisins arbitrairement définis





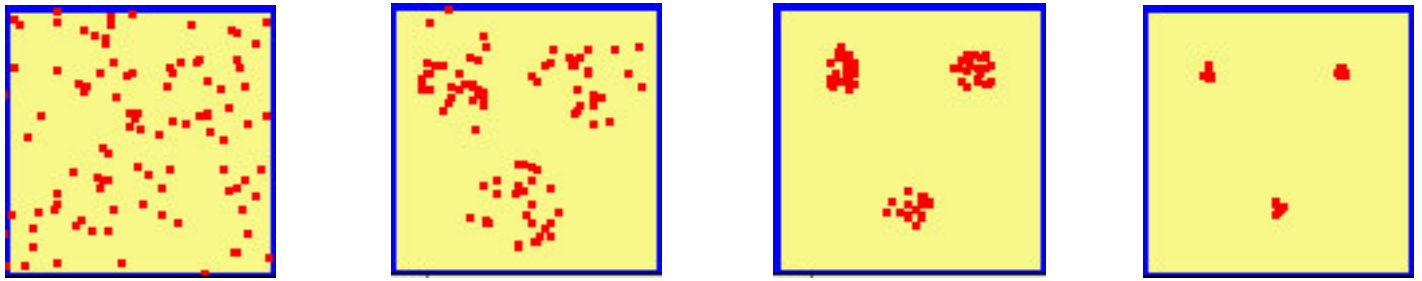




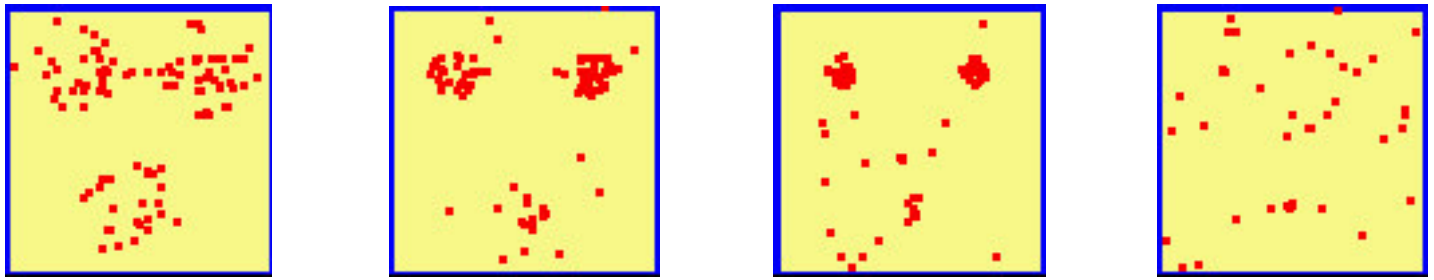
Elimination (clearing)

Representation Vecteurs de listes de chiffres $\in [0,1[^{dim}$
Exemple [3;3;3;3;3;3 ;3;3;3;3;3;3] correspond à 1/3, et à 8 par dilatation dans [7, 10]

Taux d'élimination = min (taux des chiffres communs vus dans chaque couple de composantes)
Exemple prox([1;2;3;4;5;6;7;8;9;0], [1;2;3;4;8;5;6;7;8;9;0]) = 40%
Elimination Individus trop similaires x, x' $\in [0, 1]^n$ éliminés au profit du meilleur
Sans élimination, générations 0, 1, 4, 30 (SSGA $\mu = 100$ $\tau = 33\%$)



Avec élimination, générations 1, 4, 9, 30



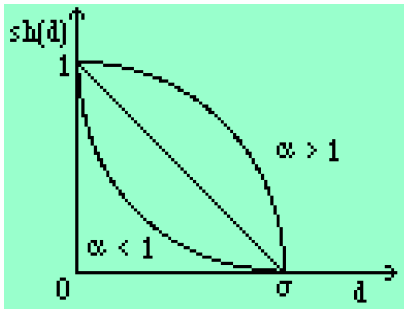
Comparaison entre SPA, et SSGA avec nichage

$n_i = \text{card}\{j \in P / d(i, j) < \sigma\}$ (mesure de la densité autour de i)

f(i) substitué par f' (i) = n_if(i) avantage les individus isolés

Extension à des voisinages flous, $sh(x) = 1 - x^{\alpha}$

$$f'(c_i) = f(c_i) \cdot \Sigma_{1<k<n}\, sh\,[d(c_i, c_k) / \sigma]$$



	SPA p=10	SPA p=50	SPA p=100	SSGA σ=0.5 α=0.1	SSGA σ=0.5 α=1	SSGA σ=0.5 α=10	SSGA σ=1.5 α=0.1	SSGA σ=1.5 α=1	SSGA σ=1.5 α=10	SSGA
Parabola dim 1	730	760	731	1340	1420	1302	1277	1183	1067	812
... ... dim 3	61074	61745	61880	2702	2289	2240	2718	1879	1931	1953
Rastrigin dim 1	2356	2663	2823	4266	4081	2948	3012	2760	2612	2150
... ... dim 3	52656	52309	51885	6867	12662	14269	4685	12050	1929	1172

SPA médiocre et le nichage n'apporte rien à SSGA

Tests entre algorithmes avec opérateurs implicites et SSGA

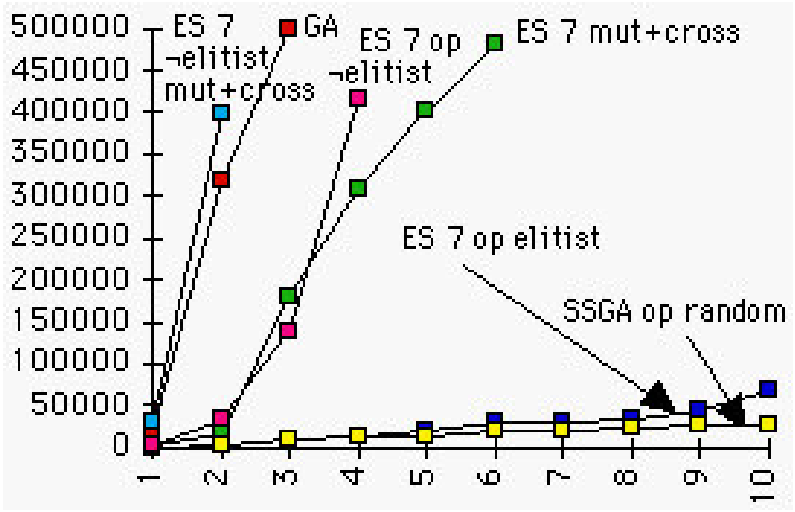
μ = 100	MGA	PSO geom	PSO social	SOMA r=(b-a)/5	SOMA (b-a)/10	DE hom	DE comp	SSGA 33%	SSGA 33% elim
De Jong dim 3	4880	3030	2060	2854	3251	2775	2450	765	853
Parabola dim 3	5261	6545	8790	6081	5304	7630	6775	1310	2183
... ... dim 10	max	max	25741	max	max	46265	49265	12880	11204
Tripod dim 2	max	14210	12685	28405	max	16120	11525	9800	11697
Rosenbrock dim 2	22658	3855	4575	32558	46059	20240	10185	20645	15939
Rastrigin dim 2	23550	12705	13225	16552	15245	10815	8395	1525	2546
... ... dim 10	max	max	max	max	max	max	max	3320	2465
Griewank dim 2	47856	95295	85725	96522	18470	9710	7110	5475	3070
... ... dim 10	max	max	max	max	max	max	max	13826	11914

SSGA(100, 33)

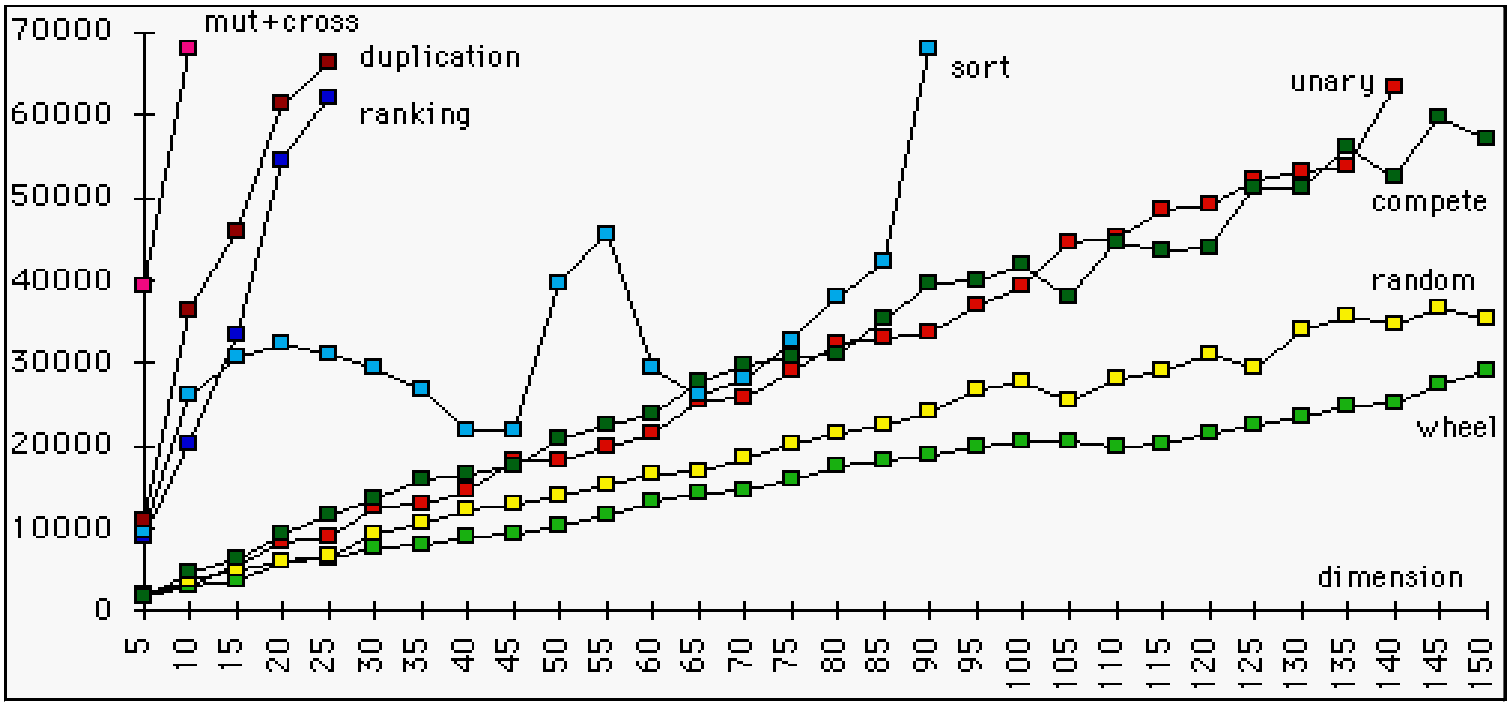
Bonne prestation de PSO social et DE mais moins bon que SSGA

Algorithmes évolutionnaires classiques

Nombre moyen d'évaluations pour Griewank suivant la dimension



et pour Rastrigin



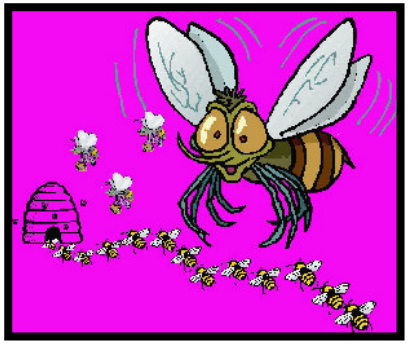
Conclusions

PEU D'INDIVIDUS
BEAUCOUP D'OPERATEURS

Avec le codage utilisé et suivant le critère de comparaison :

Performances médiocres pour l'idée de découpage de l'espace de recherche

Bon concept de rester autour des minima locaux
Bon concept pour les croisements à plusieurs
Maintenir toujours l'exploration



ELIMINATIONS DES VOISINS

Toujours garder le meilleur