TP 3 Optimisation: stratégie d'évolution

L'objectif de cette séance est de programmer une stratégie d'évolution pour la minimisation globale de fonctions à n variables avec le logiciel Python et d'implémenter un premier principe d'adaptativité des paramètres.

Exercice 1

Ecrire un programme Python réalisant une stratégie d'évolution de type (μ, λ) avec les opérations de croisement et de mutation suivantes :

— croisement (ou recombinaison) à μ points :

$$X = \frac{1}{\mu} \sum_{i=1}^{\mu} X_i$$

— mutation normale et isotrope de pas σ :

$$\tilde{X} = X + \sigma u$$

où u suit une loi normale centrée réduite dans chaque direction d'espace.

La sélection choisie sera de type 'comma', à savoir qu'on sélectionne pour l'étape suivante les μ meilleurs éléments parmi les λ enfants.

Exercice 2

Rajouter au programme précédent une méthode d'adaptation du paramètre σ basé sur la loi du 1/5, à savoir que σ est recalculé suivant la proportion p_s du nombre de mutations réussies dans une génération par la formule :

$$\sigma \to \sigma \exp\left(\frac{1}{3} \frac{p_s - \frac{1}{5}}{1 - \frac{1}{5}}\right)$$

Exercice 3

Appliquer le résultat à la fonction de Rastrigin :

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i)) + 10n$$

puis à la fonction de Rosenbrock :

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$$

On pourra prendre $\lambda = 4 + floor(3\ln(n))$, $\dfrac{\mu}{\lambda} = \dfrac{1}{2}$, $X_{init} \sim \mathcal{U}([0,1])$ et $\sigma_{init} = 0.5$.

Comparer les résultats obtenus avec ceux issus du code CAM-ES, qu'on pourra télécharger sur le site :
http://cma.gforge.inria.fr/cmaes_sourcecode_page.html