

Laboratoire 6 - métaheuristiques de construction et PSO

Exercice 1 - Explication de concepts

Expliquez brièvement les concepts suivants.

1. Les avantages et les inconvénients de considérer un voisinage large dans une recherche locale.
2. Les avantages et les inconvénients de considérer plusieurs voisinages dans une recherche locale.
3. Un avantage et un inconvénient de la stratégie *general variable neighborhood search* par rapport à une stratégie *basic variable neighborhood search*.
4. Les mécanismes utilisés par une recherche à voisinage large pour sélectionner un voisin dans un voisinage de taille très grande.
5. Les trois composantes principales de la mise à jour des *vitesses*, et leur rôle, dans une recherche à essaims de particules (PSO).

Exercice 2 : Optimisation par essaims de particules - Implémentation

Vous avez découvert par vous même la métaheuristique d'optimisation par essaims de particules (PSO). Cette question a pour objectif de vous illustrer le fonctionnement de cet algorithme sur une fonction $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. Une première observation est que cette fonction comporte deux variables continues, et non-discrètes comme nous rencontrons tout au long du cours. Ainsi, l'optimisation par essaims de particules est capable de gérer des problèmes à variables continues et est une méthode d'*optimisation globale*. De manière plus générale, PSO est capable de gérer des variables discrètes, mais plusieurs adaptations sont requises. La fonction à minimiser est la fonction d'*Ackley* en 2 dimensions¹.

$$f(x, y) = -20 \times \exp\left(-0.2\sqrt{0.5(x^2 + y^2)}\right) - \exp\left(0.5(\cos(2\pi x) + \cos(2\pi y))\right) + 20 + e^1 \quad (1)$$

Cette fonction a été choisie car elle présente de nombreux minimas locaux et un minimum global situé en (0,0). En l'absence d'autres contraintes, tout couple de réels (x, y) offre donc une solution faisable. On est donc dans le cadre d'un problème d'optimisation pure.

Trois fichiers python vous sont fournis :

- `pso.py` : implémentation de la classe PSO, à savoir un solveur basé sur cette méthode.
- `utils.py` : définition de quelques fonctions utiles comme la génération d'une solution aléatoire ou la fonction de calcul de coût.
- `main.py` : exécution du solveur.

Il vous est demandé de comprendre l'implémentation de l'algorithme, et de regarder l'impact des différents paramètres (nombre de particules, les différents coefficients, etc.), par exemple sur la convergence de l'algorithme, son temps d'exécution, etc. Vous pouvez exécuter votre code via une ligne de commande

```
1 python3 main.py --mon_argument=ma_valeur
```

Finalement, vous pouvez modifier le code afin d'y introduire d'autres mécanismes détaillés dans la lecture sur Moodle.

1. <https://www.sfu.ca/~ssurjano/ackley.html>