

Enoncé TP optimisation globale – 2nde partie
Majeure science des données
UP 4 : exploitation mathématique de simulateurs numériques
2020-21

Rodolphe Le Riche, Charlie Sire

Ce TP porte sur l'algorithme EGO vu en classe.

1. Ouvrir le fichier *egoSimple.R* . Retrouver les étapes de l'algorithme EGO :
 - a) Création du plan d'expériences initial et construction du processus gaussien
 - b) Définition du critère d'acquisition. Ce critère est contrôlé par la variable `internalFun`.
Noter aussi que le critère d'acquisition est sous une forme à minimiser (minimiser la fonction, la moyenne, $-1*EI$).
 - c) Optimisation du critère d'acquisition
 - d) Appel à la vraie fonction coût
 - e) Mise à jour du critère d'acquisition
2. Executer *egoSimple* à travers le programme principal *main4test.R* sans faire de statistiques à ce stade (`no_test = 1`).
Dans la version donnée, le critère d'acquisition est la moyenne de krigeage (pas l'EI). Repérer sur la courbe des f les évaluations qui correspondent au plan d'expérience vs. les évaluations qui sont des minimisations de moyenne. Que voyez vous ? Commenter le temps d'exécution d'EGO par rapport à d'autres optimiseurs globaux.
3. Compléter la fonction `mEI` dans le fichier *egoSimple.R* . `mEI` signifie $-1*EI$. Changer `internalFun` pour qu'elle pointe vers `mEI`. Ceci donne le vrai algorithme EGO. Le tester d'abord sur des exécutions uniques, puis en faisant des tests statistiques (`no_test=6` en dimensions 2 et 5) et le comparer à d'autres algorithmes (EGO avec acquisition à partir de la moyenne, `normal_search`, `cmaes`, `random_search`). Attention : les fichiers produits par *main4test.R* portent le nom *optimiseur_fonction_dim.** et risquent d'être écrasés par des exécutions ultérieures. Il est conseillé de renommer ces fichiers résultats.
4. [question bonus] Proposer et coder un algorithme d'optimisation globale et le tester. On pourra essayer d'améliorer `normal_search` par exemple, ou de coupler recherches locales et globales. Un exemple est donné dans le fichier *NS_lbfgs.R* où une recherche avec `normal_search` sert à créer le point initial d'une recherche locale avec `lbfgs`. De nombreuses autres combinaisons entre recherches locales et globales sont possibles, avec ou sans processus gaussiens. On peut aussi proposer un autre critère d'acquisition pour EGO... Pour cette question, vous pouvez répondre individuellement.