

**Enoncé TP optimisation globale – Part 1**  
**Majeure science des données**  
**UP 4 : exploitation mathématique de simulateurs numériques**  
**2020-21**

Rodolphe Le Riche, Charlie Sire

*Connaissances apportées par ce TP :*

- A) *Comment tester un optimiseur global*
- B) *Optimiseur local versus optimiseur global*
- C) *Observer l'effet de la variance (le « pas ») dans un optimiseur stochastique simple*

**A) Comment tester un optimiseur global. Prise en main des programmes en R.**

Cette partie A ne nécessite pas de compte-rendu, c'est une prise en main des programmes. Les parties B et C donnent elles lieu à un petit compte-rendu.

**A.1.** Télécharger les fichiers mis à disposition sur Campus ( ICM / Majeures / Majeure Sciences des données / Exploitation mathématique de simulateurs numériques / Optimisation Globale )

Les décompresser, et parcourir la présentation des codes ci-dessous, qui est un extrait du fichier readme.txt .

-----  
**MAIN PROGRAMS**

The 3 files 3Dplots.R, main4tests.R, postproc\_tests\_optimizers.R :

- \* 3Dplots.R : make 3D plots of 2D functions and superpose 1 trajectory of an optimizer.
- \* main4tests.R : repeat optimizations to study the behavior of a particular optimizer with a particular objective function.
- \* postproc\_tests\_optimizers.R : do summary plots of one or many repeated optimizations that were previously performed with main4test.R .

\* In main4tests.R describe the function you want to optimize (with the string nameoffun) and the optimizer you want to use (with nameofoptim). Also choose the number of repetitions of the optimization (no\_tests)

In postproc\_tests\_optimizers.R compare the results of different optimizations by filling the vector of strings filenames.

-----  
**BASIC USE**

- \* In 3Dplots.R and main4tests.R :
- fill in the name of the objective function (string nameoffun) and the name of the optimizer (string nameofoptim). Possible names for nameoffun are the functions given in test\_functions.R . Possible names for nameofoptim are the optimization algorithms R functions present in the directory. Examples of both are given in comments.
- fill in other characteristics of the objective function:

```

        number of dimensions (variable zdim, which must be = 2 in
        3Dplots.R), position of the optimum (variable glob_xstar), noise
        characteristics (glob_noisy, glob_tau)
- fill in parameters of the optimizer: they are all fields of the param
  list, cf. param specific to each optimizer. Typically, one finds
  lower and upper bounds on the optimization variables (LB and UB),
  and maximum number of calls to the objective function (budget),
  and the rest depends on nameofoptim.
- then execute the file.
- The results of 3Dplots.R are R plots which are also saved as image files
  fileofplot.png and contour.png . You may want to copy these files
  elsewhere or change their names otherwise they will be overwritten
  by the next call to 3Dplots.R .
- The results of main4tests.R are stored in the files
  "nameofoptim"_"nameoffun"_{some characteristic params}.RData with
  plots in "nameofoptim"_"nameoffun"_{some characteristic
  params}.pdf . The data in the .RData file is meant to be reloaded
  later (cf below). The names of the result files are weakly unique,
  it is probably safer to change them (or move them to another
  directory) before using main4tests.R again.

* In postproc_tests_optimizer.R :
- fill in the names of the .RData files you want to process in order to
  compare the performance of the optimizers (string vector
  filenames). Source the file and enjoy.

```

---

#### LIST OF FILES

```

3Dplots.R : plots 2D functions in 3D and superimposes optimizer
trajectories
cmaes.R : CMA-ES optimization algorithm
lbfgs.R : memory light BFGS optimization algorithm
main4tests.R : main program to call optimizer. Can perform many
optimizations.
normal_search.R : a fixed step ES-(1+1) search algorithm
of_wrapper.R : wraps calls to the objective function in order to store all
calls thru global variables when necessary
postproc_tests_optimizers.R : creates plots to compare different
optimizers
readme.txt : some explanations
random_search.R : a random search algorithm
utility_optim_functions.R : set of utility functions for plotting and
casting our data structures
test_functions.R : a set of test functions (potentially noisy) with
control on the location of the optimum

```

**A.2. Ouvrir 3Dplots.R** . Ce fichier permet de visualiser des fonctions 2D et des trajectoires d'optimiseurs. Utiliser l'optimiseur local « random\_search » sur une fonction convexe (la fonction "sphere" ou "quadratic") puis sur une fonction multimodale au choix. Observer et comprendre les sorties et les réglages possibles. Changer l'optimiseur en « lbfgs », observer.

**A.3.** Ouvrir « main4tests.R ». Nous allons maintenant faire des expériences numériques en répétant des optimisations. Choisir comme optimiseur « random\_search », et faire des tests sur la fonction de votre choix en dimension 5. Répéter les expériences avec l'optimiseur « lbfgs ». Utiliser le fichier « postproc\_tests\_optimizers.R » pour faire des courbes synthétiques qui comparent les deux optimiseurs.

Vous savez maintenant vous servir des programmes.

## **B) Optimiseur local versus optimiseur global**

### **B.1.** En 2D, visuellement.

Au moyen de 3Dplots.R , comparer l'optimiseur local « lbfgs » à l'optimiseur global « random\_search » sur la fonction unimodal « quadratic ». Répéter la comparaison sur une fonction multimodale ("ackley", "michalewicz", « schwefel », « rastrigin »). Avec lbfgs, faire varier le point initial. Que se passe-t-il ?

### **B.2.** En 5D, avec répétition des tests

Choisir une fonction multimodale en dimension 5 et comparer les performances de l'optimiseur local « lbfgs » aux optimiseurs globaux random\_search et cmaes. Faire varier les points initiaux pour lbfgs et cmaes.

## **C) Effet du pas de l'algorithme stochastique ES-(1+1)**

L'algorithme ES-(1+1) vu en cours est en partie codé dans le fichier normal\_search.R

**C.1.** Compléter le programme normal\_search.R : remplacer la ligne `stop("remplacer cette ligne par \"par <- ...\"")` par le bon code, comme vu en cours dans l'optimiseur ES-(1+1).

**C.2.** Observer en 2D (avec 3Dplots.R) que quand sigma est petit, normal\_search se comporte comme un optimiseur local et vice versa.

**C.3.** Etudier l'effet de la taille du pas (sigma) de normal\_search en 5D sur 1) la fonction sphère (comme pour la théorie du cours) et 2) une fonction multimodale de votre choix. Utiliser « main4tests.R » pour faire les tests et « postproc\_tests\_optimizers.R » pour faire des courbes synthétiques. Discuter les résultats.