

## Manual de Utilizador - Projeto de Computação Evolucionária

### TP12- Crossover Operators for reals (I)

Este manual foi escrito para que permita ao leitor uma replicação da experiência (completamente descrita no relatório) e também a reprodução e verificação dos resultados lá apresentados.

A experiência foi realizada com o propósito de comparar dois operadores de recombinação - One Point e Arithmetical - criando para isso duas versões de um algoritmo evolucionário padrão fornecido pela cadeira.

O código utilizado tanto para o algoritmo evolucionário como para os testes estatísticos e descrição das amostras foi parcialmente fornecido pelo Professor. Com este manual, queremos transmitir essencialmente como executá-lo.

### Dependências

O código - desenvolvido em Python 3 - deve estar inserido num ambiente que contenha as seguintes livrarias:

- random
- operator
- math
- matplotlib
- copy
- pandas
- numpy
- scipy

### main.py

Este é o ficheiro principal do projeto. Contém o algoritmo evolucionário padrão, a escolha dos seus parâmetros e três funções para cada problema de benchmark utilizado (Rastrigin, Schwefel e Griewank).

Ao ser executado, por exemplo através de `python main.py` numa linha de comandos, aparecerá para escolher o problema desejado pelo utilizador. De seguida, dois algoritmos diferentes executarão (um para cada crossover). No final, aparecerá um gráfico que permite a visualização da evolução das gerações e os resultados das 30 execuções para cada algoritmo irão ser escritas em dois ficheiros: `nomedoproblema_one.txt` e `nomedoproblema_arit.txt`.

### proj\_stat\_analysis.py

Este ficheiro contém toda a análise estatística demonstrada no relatório do projeto. É de notar que este ficheiro utiliza muitas das funções presentes no ficheiro `stat_alunos.py` que o Professor disponibilizou.

Primeiramente, é importante frisar que os ficheiros de resultados contêm duas colunas separadas por um espaço. A primeira coluna contém 30 resultados do *best fitness overall*

enquanto que a segunda contém 30 resultados que representam cada um a geração onde primeiro apareceu este melhor indivíduo.

Este ficheiro, ao ser executado (python proj\_stat\_analysis.py), deparar-se-á com a escolha do problema a analisar e da métrica de avaliação também a analisar (melhor fitness ou geração onde este apareceu).

De seguida, tais resultados são carregados do ficheiro para *arrays* e é apresentada ao utilizador uma figura com histogramas e *boxplots* dos dados do algoritmo evolucionário para o problema selecionado e para os dois tipos de recombinação. De seguida, é efetuado um teste de normalidade (Shapiro-Wilk) e um de homogeneidade de variância (Levene) a cada uma das duas distribuições para averiguar se o teste de hipóteses a realizar será paramétrico ou não. Caso estes 3 testes anteriores retornem valores maiores do que 0.05, o teste paramétrico Dependent T-test é feito. Caso contrário, e é altamente provável que isto aconteça, é feito o teste não paramétrico de Wilcoxon. No final, um destes dois testes retornará dois resultados - o t-statistic e o p-value. Dependendo do valor deste p-value, a hipótese nula em questão será rejeitada ou não.