CS411: Evolutionary Computation and Applications

Assignment 1: Comparison of search operators and selection schemes on a set of benchmark functions in the continuous domain

Jialin Liu (*liujl@sustc.edu.cn*)

22 February 2019

1 Overview

The main task of this assignment is to compare different crossover operators, mutation operators and selection schemes on a set of benchmark functions in the continuous domain. This assignment has 100 marks, which will take 20% in your final mark of this module. The mark you get in this assignment depends on the your algorithm's performance on the benchmark functions, the quality of your report, program and presentation.

2 Task

You will implement a structure of an evolutionary algorithm (EA), as well as

- 3 different crossover operators;
- 3 different mutation operators;
- 3 different selection schemes.

Then, you will compare the different operators on a set of benchmark functions.

2.1 Benchmark functions

We consider 6 of the benchmark functions that have been used by Yao et al. [1]. They are namely Step Function (f_6), noisy Quartic Function (f_7), Generalised Rastrigin's Function (f_9), Ackley's Function (f_{10}), Goldstein-Price Function (f_{18}) and Shekel's Function (f_{21}) in [1].

The detailed definition and formalisation of these functions can be found in the Appendix of [1]. Additionally, you can visualise the functions at `https://en.wikipedia.org/wiki/Test_functions_for_optimization` or `http://benchmarkfcns.xyz/fcns`.

2.2 Programming aspects

Programming language Matlab

- Free download: `http://lib.sustc.edu.cn/UserFiles/download/1489545490853.docx?locale=zh_CN`
- Tutorial: `https://ww2.mathworks.cn/support/learn-with-matlab-tutorials.html`
- Examples: `https://ww2.mathworks.cn/help/examples.html`

Project A Matlab project is provided on *Sakai*. The project structure is shown in Figure 1.

- The folder named "*benchmark*" contains the benchmark functions to be optimised.
- The folder named "*utils*" contains some useful functions.
- The Matlab file named "*test.m*" will be used to test your your EAs.

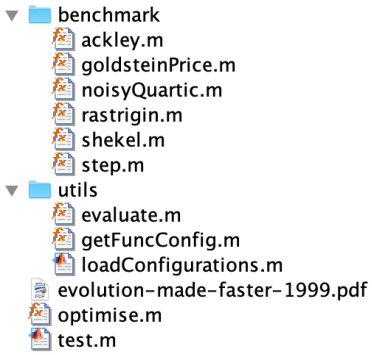

Figure 1: Project structure of Assignment 1.

- **The Matlab file named "*optimise.m*" is where you should implement your EA, including different operators, and fitness function. Normally, this is the only file that you need to work on.** The content of the given *optimise.m* is printed below:

```
function [apprx,appry]=optimise(funcName,budget,crossoverIdx,mutationIdx,selectionIdx)
warning on MATLAB:divideByZero
% Input:
% crossoverIdx: the index of crossover operator \in {1, 2, 3}
% mutationIdx: the index of mutation operator \in {1, 2, 3}
% selectionIdx: the index of selection operator \in {1, 2, 3}
%
% Output:
% apprx: the approximate optimum
% appry: the approximate optimal value
%
%% TODO: below to implement your own EA
%% You may need to provide multiple files if you are asked to implement
%% more than one algorithm or an algorithm with different configurations.
%------------- BEGIN CODE --------------

%------------- END OF CODE --------------
end

function y=fitness(funcName,x)
eval(sprintf('objective=@%s;',funcName)); % Do not delete this line
%% TODO: below to implement your own fitness function
%------------- BEGIN CODE --------------
objValue=objective(x);
y=objValue;
%------------- END OF CODE --------------
end
```

Remarks

1. Normally, you don't need to edit any file besides the file *optimise.m*. You will implement your own EAs inside *optimise.m*.
2. You are not alloed to add new files or folders. *optimise.m* is the only file that you will submit. It is possible to define multiple functions called inside the function *optimise* inside the *optimise.m* file.
3. If you find any bug, please contact the Teaching Assistant.

Evaluation of program The submitted *optimise.m* will be tested on the same Macbook Pro using Matlab R2018b. Any program that is failed to run receives 0 mark.

2.3 Report

A report (in pdf format) must be submitted, named as **report.pdf**. MS Word and LaTeX templates can be found at `https://www.ieee.org/conferences/publishing/templates.html`. You should use these templates.

The expected structure is given below.

Abstract

Introduction

Background Introducing the benchmark & related work.

Proposed Algorithm Introducing your EA(s) and operators. Pseudo-code for all operators is required.

Experimental Results and Discussion Repeat 50 times the experiment using the given test file and report the results. All parameters and corresponding values used in the evolutionary algorithm and experiments should be reported. Discussion of the results.

Conclusion and Future Work

Remark: Please be careful with the grammar, spelling and format.

2.4 Presentation

Examples of evaluation criteria are, but not limited to:

Description of the tested functions What are they? What are their characteristics? Why they are challenging? ...

Description of the algorithm Solution representation, different operators, ...

Results and discussion Analysis of results, statistical test, convergence curves, performance comparison, ...

Presentation of the slides Format, typeset, spelling, grammar, ...

Language and clearness

3 Submission

3.1 What to submit

Report and program Each student should submit one single file for report and program, repectively

- A Matlab file named as **optimise{studentnumber}.m** using the game template and respecting the required function declaration explained previously. Example: *optimise12345678.m*.
- A pdf file named as **assignment1-report{studentnumber}.pdf**. Example: *assignment1-report12345678.pdf*. The structure of the report is described in the next section.

Presentation slides Each student should submit one single file for her/his presentation. The submitted file can be of one of the following formats:

- **assignment1-presentation{studentnumber}.ppt**
- **assignment1-presentation{studentnumber}.pptx**
- **assignment1-presentation{studentnumber}.pdf**

Example: *assignment1-presentation12345678.pdf*.

3.2 Where to submit

Upload your *optimise*{*studentnumber*}*.m* file, or *assignment1-report*{*studentnumber*}*.pdf* file or presentation via *Sakai*.

3.3 Important dates

First submission of program (15%) 22:59 (Beijing time) March 10 (Sunday), 2019.

Submission of your first program, in which the structure of your EA is finished and at least one crossover operator, one mutation operator and one selection scheme are implemented. Thus, your *optimise* function can be tested on the given benchmark functions using taking *(*,*,1,1,1)* as input, and it plots/prints automatically the optimisation output for each of the benchmark functions.

Submission of presentation materials 22:59 (Beijing time) March 21 (Thursday), 2019.

Submission of the final version of your slides for presentation. No further update or edit (even minor) is allowed after this deadline.

Presentation (50%) March 22 (Friday), 2019.

Individual presentation. The order will be decided in the morning.

Second submission program (10%) & report (25%) 22:59 (Beijing time) March 24 (Sunday), 2019.

Submission of the report and the final version of program. 3 crossover operators, mutation operators and selection schemes should be implemented and can be tested on the given benchmark functions. The program should be able to plots/prints automatically the optimisation output for each of the benchmark functions.

4 Prohibition

You will get 0 as score for this assignment if any of the following cases happens:

- You submit more than one file for your program.
- You use other programming languages.
- You don't respect the naming policy of files.
- The report/program submission is delayed for 3 days (72 hours) or more.
- Plagiarism.

Contact For any question regarding this assignment, please email to *liujl@sustc.edu.cn*. The subject of the email should respect the format: **[CS411] Assignment 1 (LastName/FirstName-StudentNumber)** Example: [CS411] Assignment 1 (Liu/Jialin-12345678)

References

[1] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, 3(2):82–102, 1999.