Project 2 – NCS Algorithm and Its Application

1 Overall Description

NCS is a population-based search algorithm (like genetic algorithm) for continuous optimization problems. The difference is that NCS uses the idea of negatively correlated search to make individuals search different regions in the decision space.

In this project, what you need to do is to find the best parameter values (i.e. parameter tuning) as you can for NCS on two benchmark test functions (F6 and F12) and one application problem (OLMP). In other words, you need to find three parameter settings, each for one test function or the application.

This project is divided into two phases. In each phase, you will have at least one week for parameter tuning.

	Scoring rules	DeadLine
Lab4	Scores are given according to the performance of your parameter settings on F6 and F12. Ranks are given for students who pass the passing line according to the obtained function values on F6 and F12 and then scores are given based on the ranks.	Date: 20th October Time: 23：55
Lab5	Scores are given according to the performance of your parameter settings on F6 and F12. Ranks are given according to the compression rate on OLMP for students who pass the passing line and then scores are given based on the ranks.	Date: 27th October Time: 23：55

After Lab5, a carefully-written experiment report need to be submitted.

Parameter_Score =（Lab4+Lab5）/2

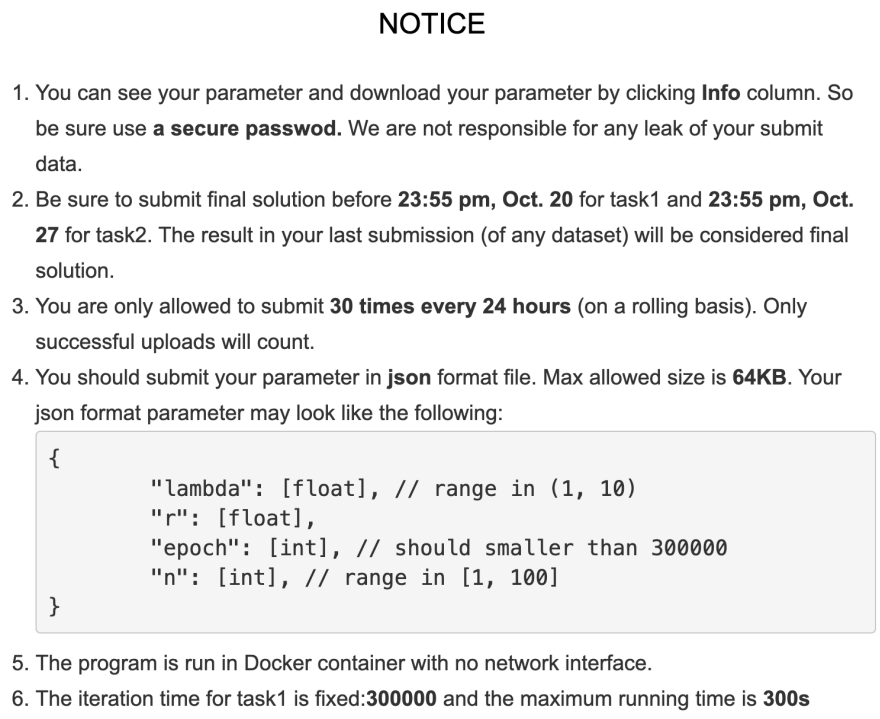
Project2_score = Parameter_Score *0.7+Report_Score*0.3

2 The Use of the Platform

2.1 Interface Introduction

The url for NCS platform is : http://10.20.107.171:8080/. You can log in with the student ID and password. The default password is the student ID. At the first login, the system will remind you to change the password. After logging in successfully, you can see the below interface.

The first interface is Notice. This gives the notices you need to pay attention to.



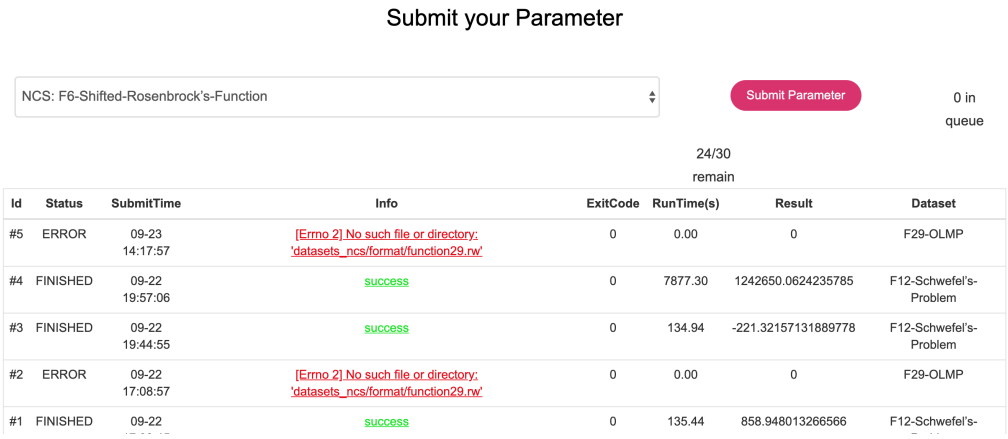
NOTICE

1. You can see your parameter and download your parameter by clicking **Info** column. So be sure use **a secure passwod.** We are not responsible for any leak of your submit data.
2. Be sure to submit final solution before **23:55 pm, Oct. 20** for task1 and **23:55 pm, Oct. 27** for task2. The result in your last submission (of any dataset) will be considered final solution.
3. You are only allowed to submit **30 times every 24 hours** (on a rolling basis). Only successful uploads will count.
4. You should submit your parameter in **json** format file. Max allowed size is **64KB**. Your json format parameter may look like the following:

```
{
        "lambda": [float], // range in (1, 10)
        "r": [float],
        "epoch": [int], // should smaller than 300000
        "n": [int], // range in [1, 100]
}
```

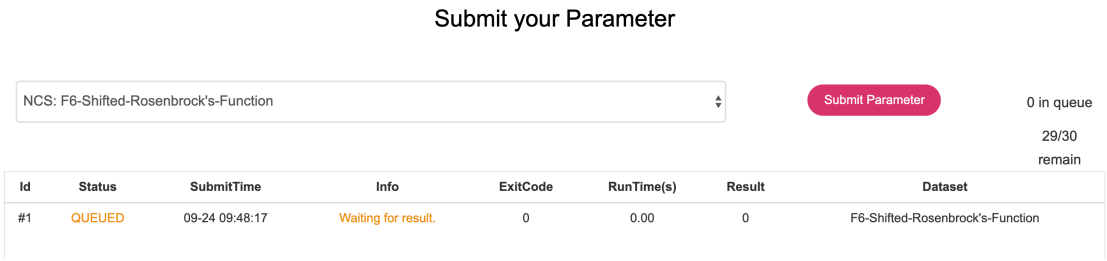
5. The program is run in Docker container with no network interface.
6. The iteration time for task1 is fixed:**300000** and the maximum running time is **300s**

The second interface is where you can submit your parameter values, as shown below:

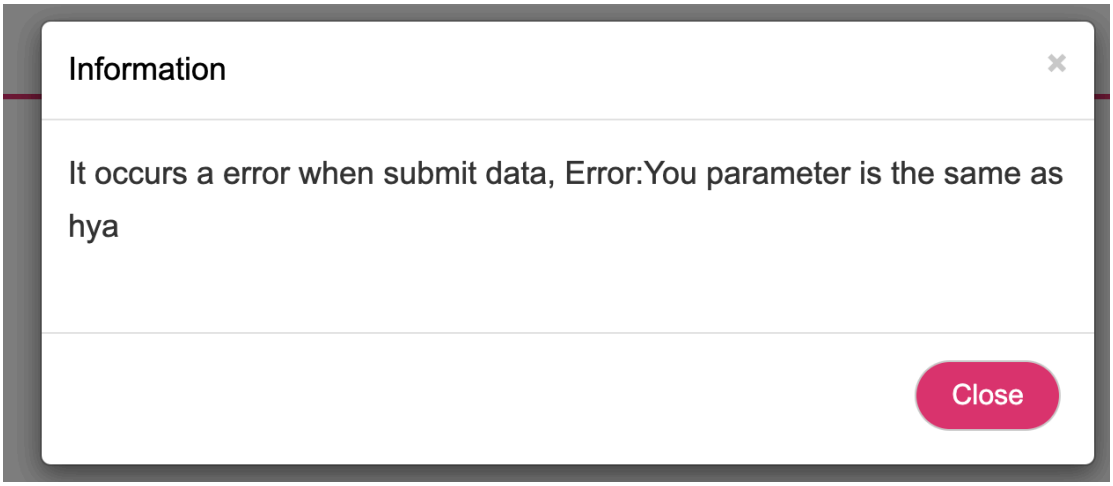


First, choose the function for which you want to submit the parameter, in the drop-down box to the left of the "Submit Parameter" button. There are three functions you can choose from: F6, F12 and OLMP. F6 and F12 are for the first project phase. OLMP is an application project for the second project phase. OLMP is still not online yet.

Second, click the “Submit Parameter” button and select your configured .json file. If the format of your file is fine, and it is different from other students, the system will prompt that you submit successfully and need to wait for the background to schedule the execution. (Because some tasks will need to run for a long time, so the task you submit need to queque sometimes, on the right side of the button, you can see how many tasks are currently in the queue if you need to queque).

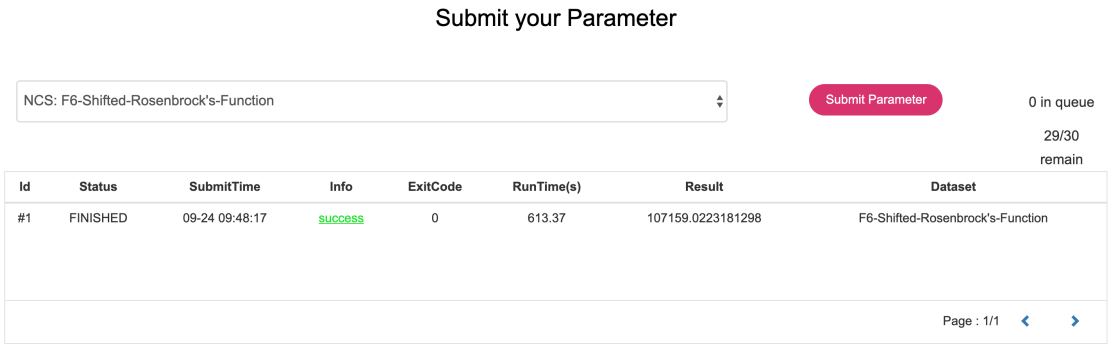


If your parameter values are the same to other students'. The system will remind you to change your parameter values.

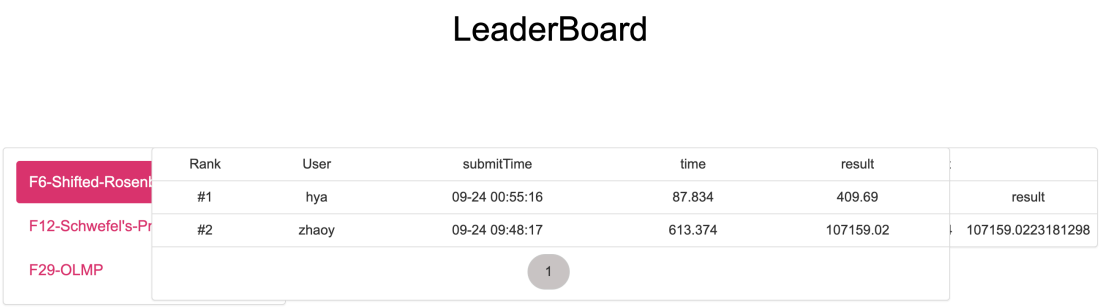


For your own submitted parameters, you can submit and run repeatedly, the system will save the best results of the historical results.

The bottom half of this page is a table that records the fuction, status, details, error codes, run time, and run results of the task. When the task you submitted is scheduled, it will start executing. After the execution, the status, time and result of the operation will be returned to the table.



The third interface is the leaderboard, which shows the ranking of the running results of the students on the current function from high to low. Both F6 and F12 are looking for the minimum function value, so the smaller the obtained result, the higher the rank you can get.



2.2 Json File

Parameter values are saved in .json file, the format is like below:

```
{
    "lambda":1,
   "r": 0.99,
   "epoch": 100,
   "n":  80
}
```

where：

lambda: the expected value that λ finally wants to achieve in the NCS paper, the setting range is (0,10)

r：In the formula 3.6 of the NCS paper, the coefficient for multiplication, combined with the running time, has no setting range.

Epoch: for how many iterations that each random local search is run

n：the number of individuals in the population, the setting range is [1,10]

implicit parameter：

Tmax is fixed to 300000

2.3 Notes

1. For Lab4, the parameter values you submit must not let the server run for more than 5 minutes. For Lab5, the parameter values you submit must not let the server run for more than 2 minutes. Otherwise, the server will end your process.

2. To avoid the effect of the randomness in the algorithm on your results, the same random number seed is used for all students.

3. Passing line：

F6: <=410.8

F12: <=1110

OLMP: compression rate>=0.8 (valid rate is in [0,1], other values are invalid)

3 Documents and Codes

1、Python version: 3.6

2、NCS paper, benchmark description and NCS Code:

2016TangYangYao_JSAC.pdf

Problem_Definitions_and_Evaluation_Criteria_for_th.pdf

ncs_code.zip

Run the ncs_code with command (-d 12: means F12):

python3 -m algorithm_ncs.ncs_client -d 12 -c algorithm_ncs/parameter.json

The following command is about run NCS for 15 times without fixed random number seeds:

python3 -m algorithm_ncs.ncs_c

4 Report Requirements

1）Refer to《ProjectReport_Template.docx》

2）Or Use the Style of《Transactions-instructions-only.pdf》和《ieeecitationref.pdf》

3）Excellent Report form Last Year《Project2_report_11410151.pdf》

4）To Add：Report Scoring Criteria