

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

MASTER ICT

MI1.07 –SOFTWARE DEVELOPMENT PROJECT

Huy-Duc LE

University of Science and Technology of Hanoi
Hanoi, Vietnam
lehuyduc3@gmail.com

Vinh-Nam HUYNH

University of Science and Technology of Hanoi
Hanoi, Vietnam
protosnamjune2nd@gmail.com

Duc-Quyen NGUYEN

University of Science and Technology of Hanoi
Hanoi, Vietnam
hakonaryuuj@gmail.com

Vu-Hung NGUYEN

University of Science and Technology of Hanoi
Hanoi, Vietnam
hungnga25197@gmail.com

March 2020

GROUP PROJECT – REPORT

Project REDUCE Manual

The aim of this documentation is to deliver instructions to the user of our Software Development Project – the REDUCE algorithm.

Table of Contents

Chapter 1	Software Overview.....	3
1.1	Goals	3
1.2	Constraints.....	3
Chapter 2	Software Manual.....	4
2.1	Graphical User Interface	4
2.2	Detail features.....	5

Chapter 1

Software Overview

1.1 Goals

Reduce is a parallel design pattern that consists of compute a value from a set of values. One big problem here is linked to the floating point representation of real numbers. Indeed, it is well-known that the sum of several floats is rarely correct!

For example: Consider 2 arrays of float number

$$A = \{1.000000000, 2.000000000, 3.000000000\},$$

$$B = \{3.000000000, 2.000000000, 1.000000000\}.$$

Let's take sum of all elements inside each array. We may obtain $\text{res_A} = 6.000000001$ and $\text{res_B} = 5.999999999$. These 2 results are not exactly the same!

In order to retrieve a “good” approximation of the result, many algorithms may be proposed.

The goal of this project is to compare some of those algorithms, in quality and complexity, respectively.

1.2 Constraints

- The project must be written in C++ and Qt.
- The user should be able to access saved-experiments-result files.

Chapter 2

Software Manual

2.1 Graphical User Interface

The software GUI contains 3 main regions. They are corresponding to Progress tracking, Main input interface and Output box in succession.

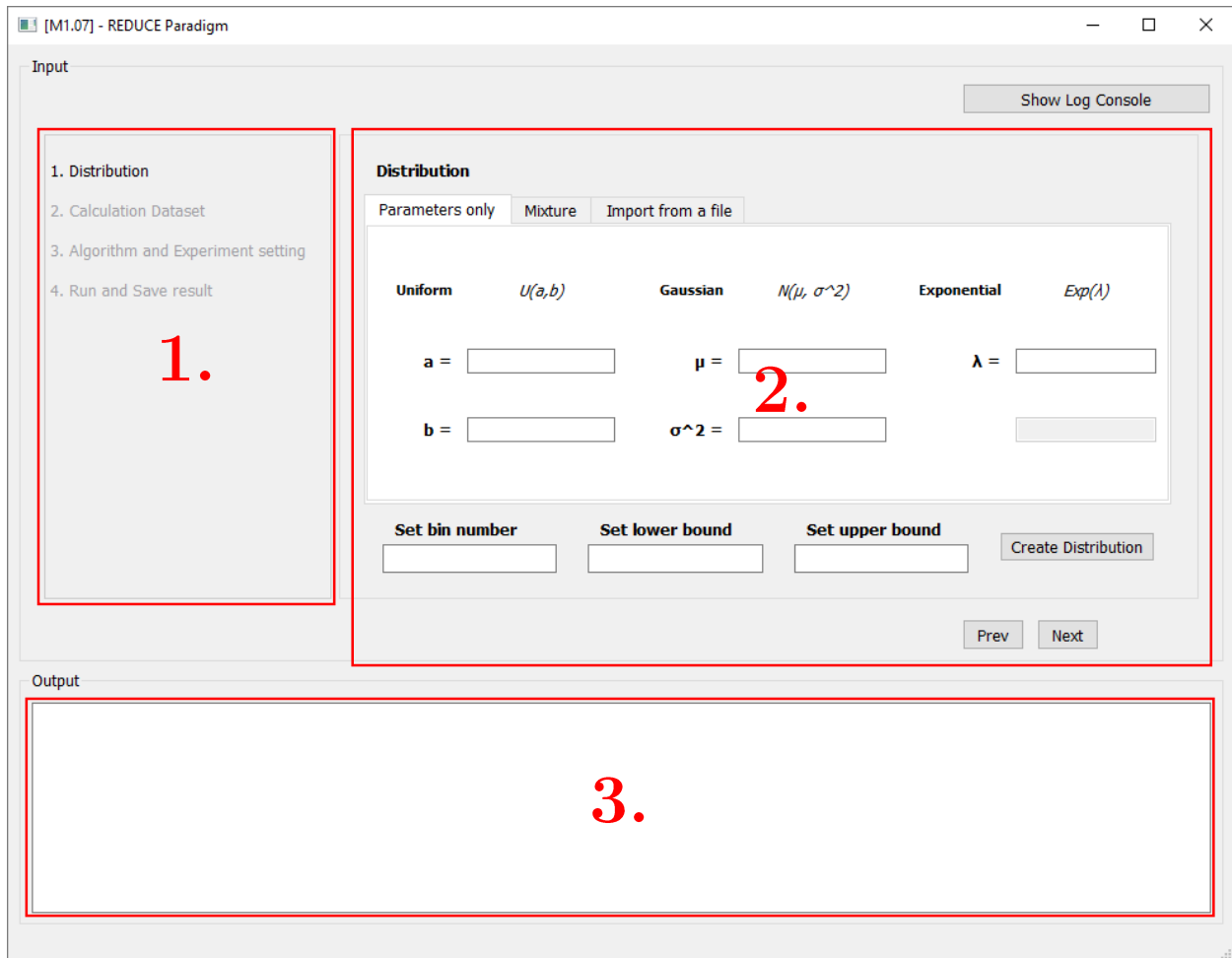


Figure 1: Project GUI – Overview

2.2 Detail features

Region 1: Progress tracking

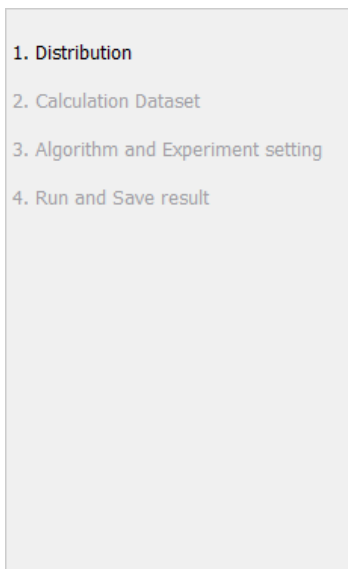


Figure 2: Project GUI – Progress tracking

This region will automatically highlight the current task in the workflow. It ensures that the user know what is going on.

Region 2: Main input interface

Region 2.1: Set Distribution

1. Input distribution options

Distribution

Parameters only Mixture Import from a file

Uniform $U(a, b)$ **Gaussian** $N(\mu, \sigma^2)$ **Exponential** $E(\lambda)$

a = **μ** = **λ** =

b = **σ^2** =

2. Set Parameters

Set bin number **Set lower bound** **Set upper bound** **Create Distribution**

3. General distribution settings

4. Create distribution

Prev Next

Figure 3: Project GUI – Set Distribution area

- The very first step is to select the most proper input options (see **1. Input distribution options** above). There are 3 ways to setup distribution:

+ Parameters only: The user needs to type in the parameters. The system then parse those parameters to form a distribution.

Choose this option for a distribution such as $U(a, b) + N(\mu, \sigma^2) + E(\lambda)$.

+ Mixture: The user can type in an equation of distribution. The system then parse the string to form a distribution.

Choose this option for a complex mixture of distribution such as

$$(U(a, b) + N(\mu_1, \sigma_1^2)) * (N(\mu_2, \sigma_2^2) + E(\lambda))$$

+ Note: only parentheses “(“ and “)” are supported

+ Import from a file: The user can select a text file which contains strings of distribution.

Choose this option to import complex mixture of distributions from a text file such as $((U(a, b) + N(\mu_1, \sigma_1^2)) * N(\mu_2, \sigma_2^2)) + E(\lambda_1) + E(\lambda_2)$

- After setup distribution, it's time to establish bin number (the higher bin number, the higher precision), then Lower and Upper bound (measurement range).
- Click Next button to go to the next task.

* **Note: If uniform distribution is used, then a, b must be inside lower bound \rightarrow upper bound range.**

Region 2.2: Set Calculation Dataset

1. Choose Data Type 2. Set Number of data

Data Type
Array

Number of data

Calculation dataset
Generated randomly Import from a file

3. Get dataset options

Save file setting

Generate! Save to Browse Save

4. Generate dataset 5. Browse Save Dir 6. Save dataset

Prev Next

Figure 4: Project GUI – Set Calculation Dataset area

- Click Next button to go to the next task.
- Click Prev button to go to the previous task.

Region 2.3: Set Experiment setting and Choose Algorithm

1. Choose Operation 2. Set Number of test

Experiment setting

Operation
Sum

Number of test

Shuffle/Generate
☐ Generate new

Precision 3. Get dataset options

4. Set Precision type PDDOUBLE

Algorithm

Algorithm List
LINEAR

5. Add a current algorithm in
Algorithm List or Remove
one in Selected Algorithm

Selected Algorithm

Prev Next

Figure 5: Project GUI – Set Experiment setting and Choose Algorithm area

- Click Next button to go to the next task.
- Click Prev button to go to the previous task.

Region 2.4: Run and Save Experiment

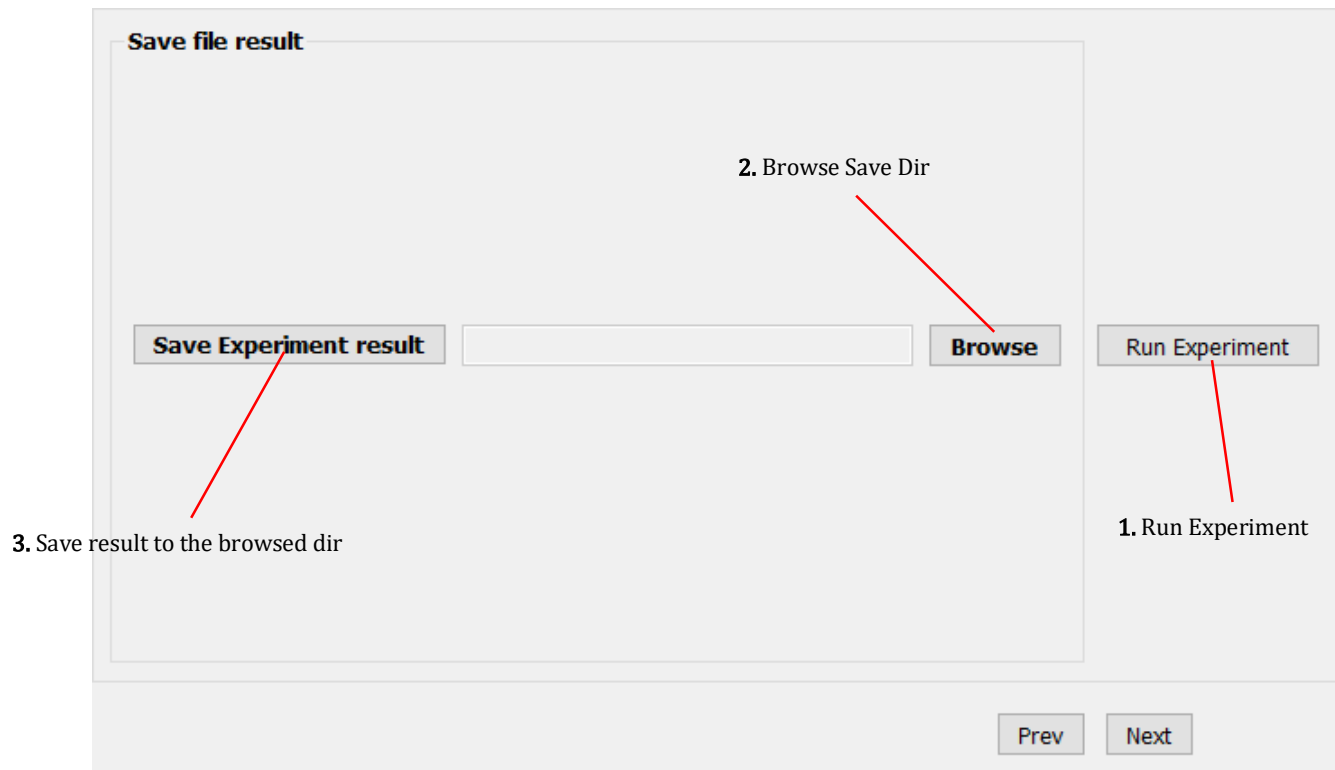


Figure 6: Project GUI – Run and Save Experiment area

- Click Prev button to go to the previous task.

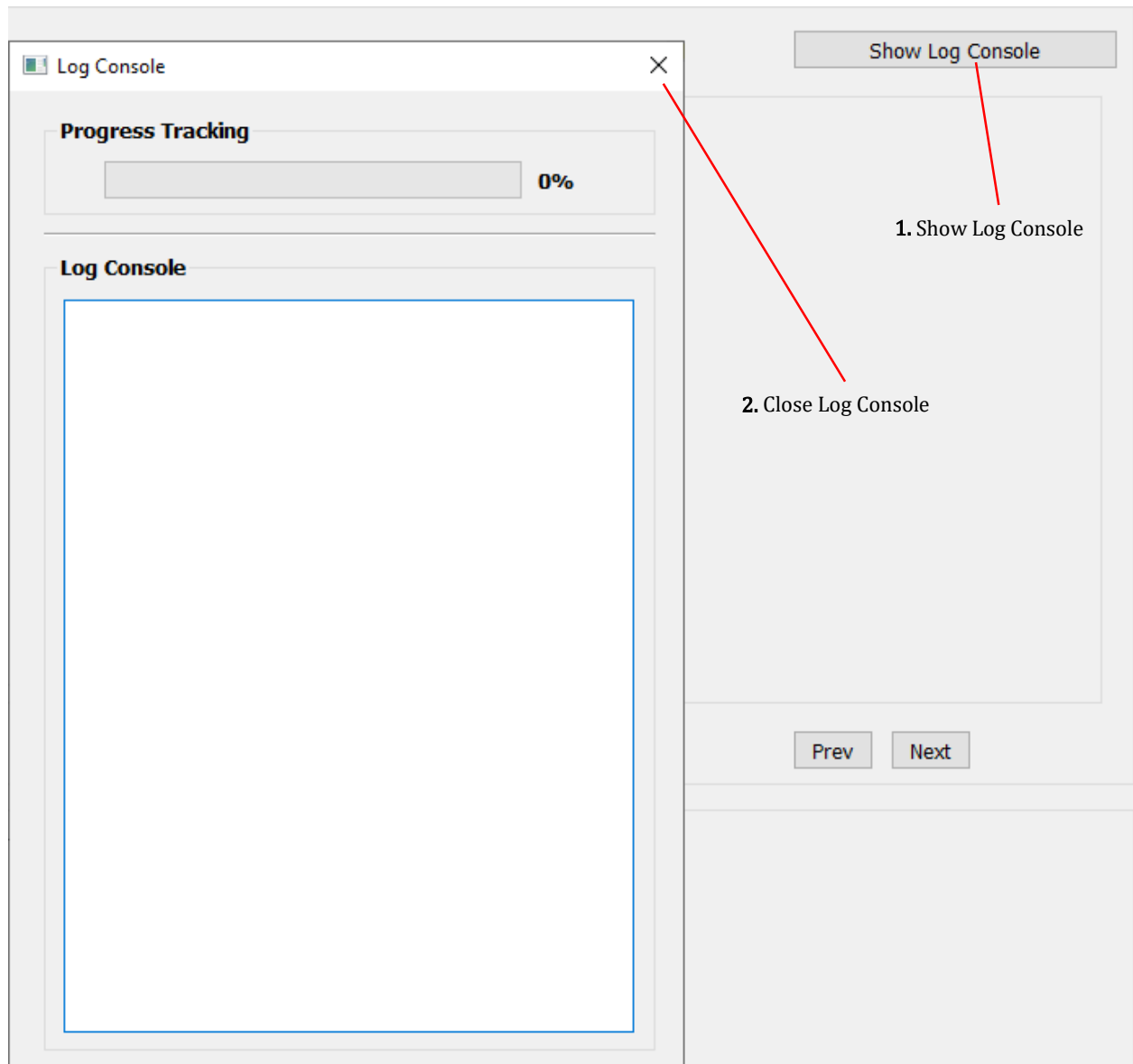


Figure 7: Project GUI – Show Log Console option

Region 3: Output box



Figure 8: Project GUI – Output box

This region displays the result of the finished experiment(s).