README-FILE for *NumOpt* v1.12

a *Numerical Optimization Framework*

The artifact is a prototype program (NumOpt v1.12) for reproducing the experiment results of the paper "Global Optimization of Numerical Programs via Prioritized Stochastic Algebraic Transformations". It is licensed under the GNU General Public License version 2 (GPL v2). Other details of the license information are in the document LICENSE.txt.

We make the artifact publicly accessible at: http://seg.nju.edu.cn/eytang/numopt/

The artifact is mainly written in Python 3.6, and fully tested and evaluated under a brand new Ubuntu 16.04.5-LTS-AMD64 Desktop System, we strongly recommend reviewers use the same system and environment. Other x64 Linux systems may also work well, but we are not sure about it. The tutorial detailedly describes the steps of reproducing the results presented in the paper. The installation (of running `make.sh`) needs less than 20 minutes on an Intel 3.0GHz workstation with 8GB memory, whereas the evaluation (of running `bench_eval.sh`) needs a couple of hours to reproduce the experiment results. Executing `bench_eval.sh` takes more time because it needs to optimize 28 subjects and test every subject with a lot of numerical inputs.

Tutorial

1. We assume a brand new AMDx64 Linux system (We strongly recommend users to use Ubuntu 16.04.5-LTS-AMD64 Desktop version, which is fully tested and evaluated. The installer is at http://releases.ubuntu.com/16.04/ubuntu-16.04.5-desktop-amd64.iso.)

2. Install required packages with the ***root*** account:

```
apt-get install build-essential
apt-get install m4
apt-get install autoconf
apt-get install libtool
add-apt-repository ppa:jonathonf/python-3.6
apt-get update
apt-get install python3.6
apt-get install curl
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python3.6 get-pip.py
pip3 install antlr4-python3-runtime==4.7.1
pip3 install sympy==1.1.1
apt-get install clang
```

3. Download the optimization framework:

```
mkdir ~/NumOpt
cd ~/NumOpt
wget -c http://seg.nju.edu.cn/eytang/numopt/numopt_v1.12.tar.gz
```

4. Build the library and dependences:

```
cd ~/NumOpt
tar -zxvf numopt_v1.12.tar.gz
cd ~/NumOpt/NumoptFramework
bash make.sh
```

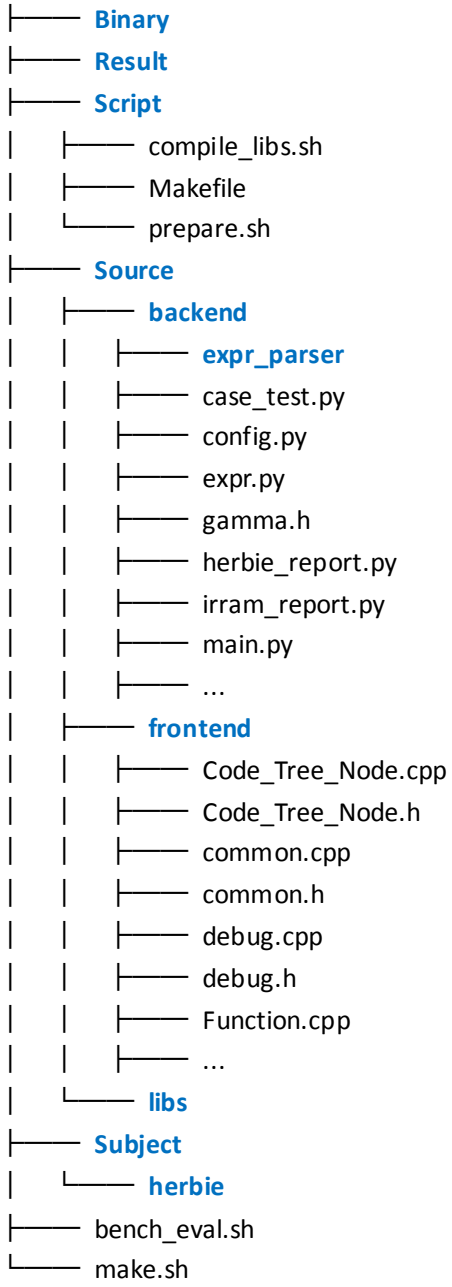
5. Optimize the 28 classic benchmarks with our pre-composed scripts:

```
bash bench_eval.sh
```

After executing the evaluation scripts, the optimized program of every case in the benchmark is located at `Binary/OptimizedCode/<case_group_name>/<case_name>`. The corresponding numerical error of every optimized program is evaluated and stored in the file `Result/<case_group_name>/<case_name>/error.rd`.

Directory Structure

The directory structure of numerical optimization framework is as follows: (We only list the main skeleton in 3 levels and mark the folders with the blue color.)



The source code of the framework and the related libraries are in the folder **Source**. The compiled binaries are in the folder **Binary**. The folder **Subject** stores the source code of our testing subjects, whereas the folder **Script** stores scripts in the evaluation. Finally the folder **Result** contains the evaluation outputs.

Screenshots

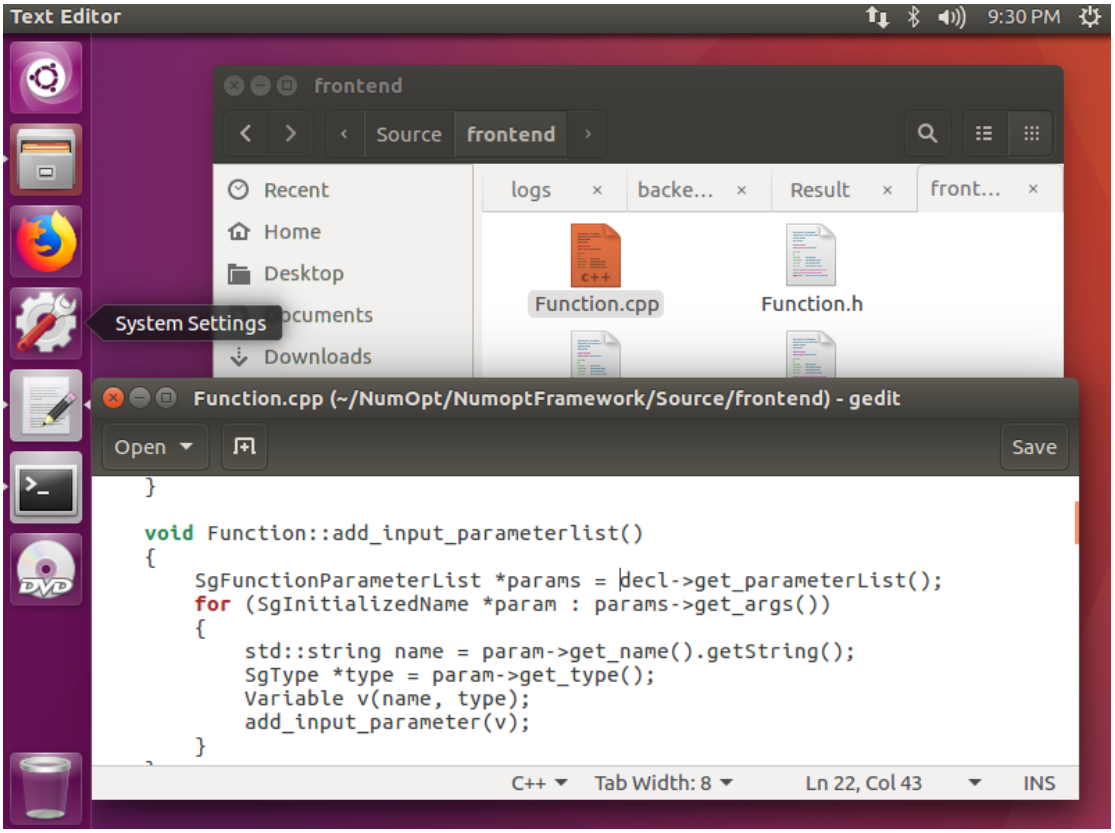


Figure 1: Frontend Source Code

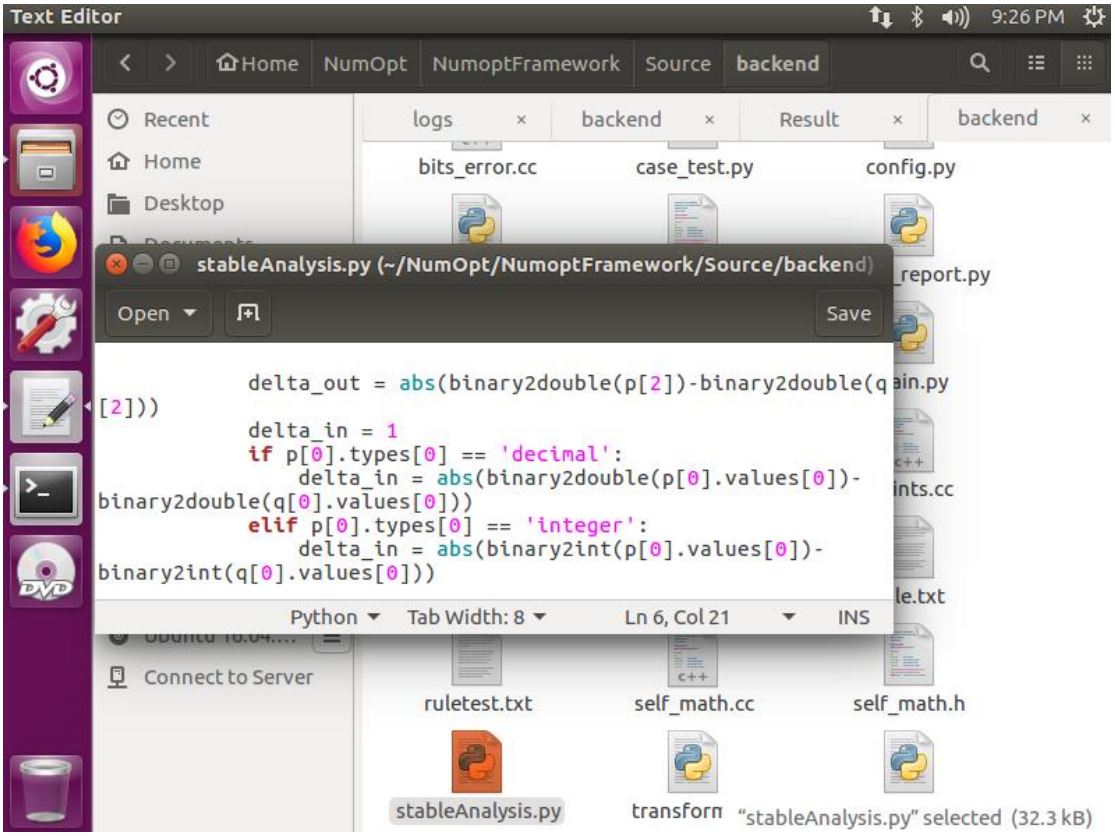


Figure 2: Backend Source Code

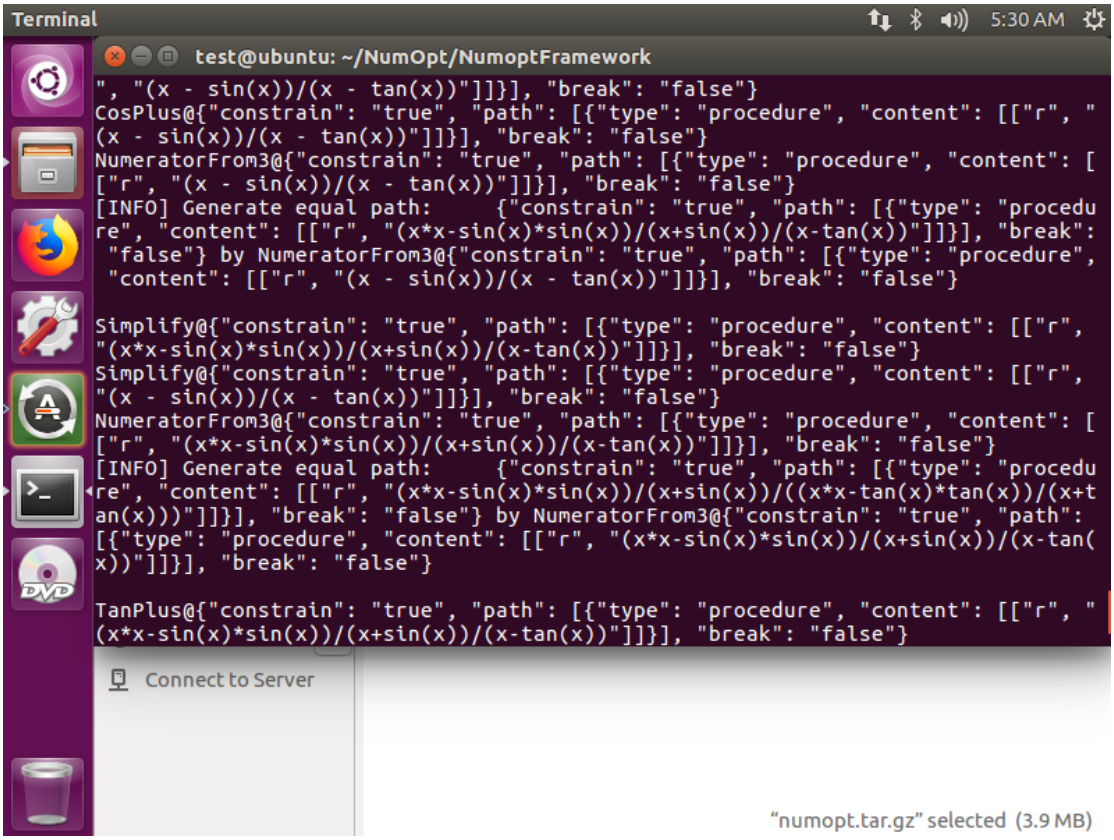


Figure 3: Runtime Output in Optimizing a Numerical Program

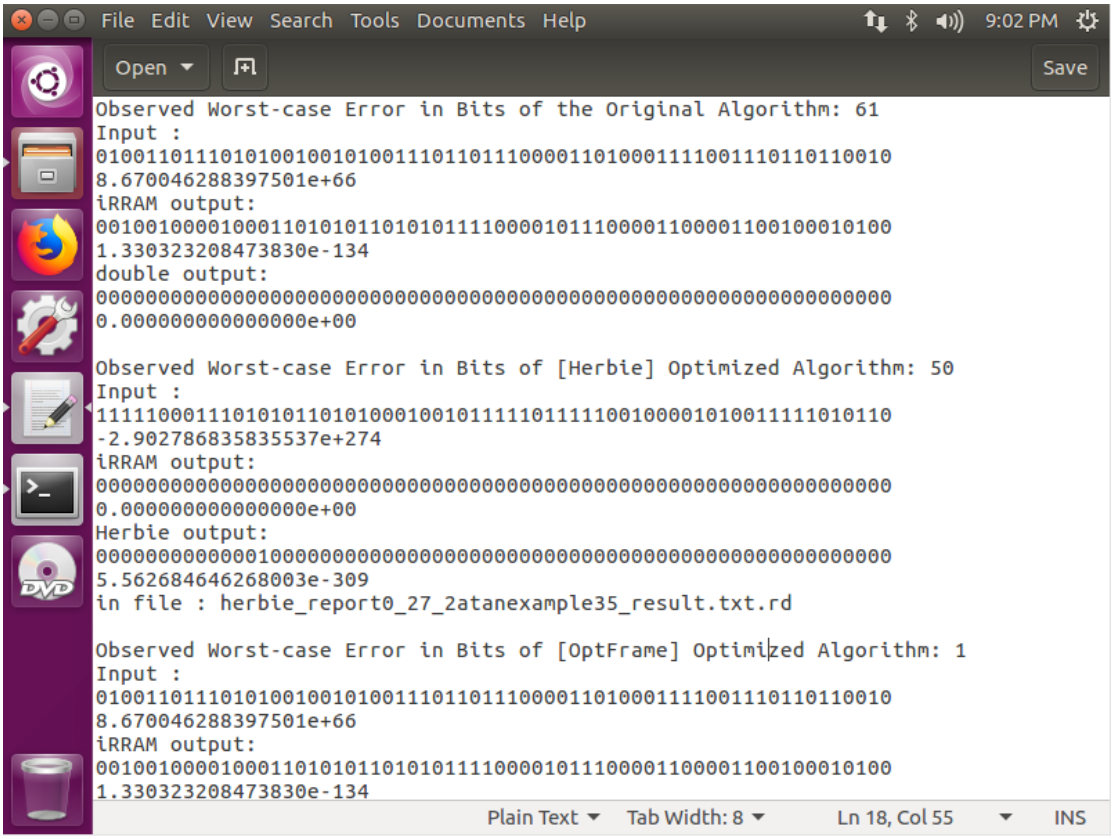


Figure 4: Observed worst-case Error of an Optimized Program