



Semester-long Internship Report

On

FOSSEE Optimization Toolbox

Submitted by

Ilma Siddiqui

Under the guidance of

Prof. Ashutosh Mahajan

Industrial Engineering and Operations Research,
IIT Bombay

Mentor

Mr. Rupak Rokade

FOSSEE, IIT Bombay

July 2021

Acknowledgment

I would like to thank the FOSSEE team at IIT Bombay for giving me the opportunity to work on and contribute to the development of FOT version 0.4.1. Through this internship, I was able to connect with so many wonderful people, which I am thankful for.

I would like to convey my heartfelt gratitude to Mr. Rupak Rokade Sir for being a great mentor and guiding us throughout this internship. I'd also like to express my gratitude to Prof. Ashutosh Mahajan for his unwavering support and assistance. I am eternally thankful to my teammates for constantly supporting and helping me, as well as for making this experience more enjoyable.

I am glad that I got this opportunity, which helped me to add new skills to my skillset. I hope that the skills I learnt through this internship will help me grow as a developer.

Contents

1	Introduction	3
1.1	About FOSSEE Optimisation Toolbox	3
1.2	Important Directories of FOT	3
1.3	CUTEst and CUTE set of PrincetonLib	5
2	Contribution	6
2.1	Task 1	6
2.1.1	Problem Statement	6
2.1.2	Changes Made	6
2.2	Task 2	7
2.2.1	Problem Statement	7
2.2.2	Changes Made	7
2.3	Task 3	8
2.3.1	Problem Statement	8
2.3.2	Changes Made	8
3	Significance of limited memory option	9
4	Future Work	12

Chapter 1

Introduction

1.1 About FOSSEE Optimisation Toolbox

FOSSEE Optimization toolbox is a toolbox in Scilab maintained and developed by FOSSEE (Free and Open Source Software in Education), IIT Bombay. It offers several optimization routines including, but not limited to, linear optimization, integer linear optimization, unconstrained optimization, bounded optimization and constrained optimization. The function calls and outputs are similar to those available in Matlab. These routines call optimization libraries in the backend, all of which are COIN-OR libraries. CLP is used for linear programming, CBC for integer linear programming, IPOPT (with MUMPS) for nonlinear optimization and Bonmin for integer nonlinear optimization. There are also routines for specific optimization problems like linear and nonlinear least squares, minimax, and goal programming using these solvers. The tool box is free to use and open-source. A user can install the toolbox using the ATOMS package manager available in Scilab. FOT includes Scilab help files, examples and tests to help the user quickly understand the commands and options required to solve different types of optimization problems.

1.2 Important Directories of FOT

Some of the important directories are :

- macros : Contains the scilab function files containing the functions that are to be called from Scilab and their respective builder files.
- sci_gateway : Contains gateway files which are to be called from the thirdparty directory
- thirdparty : Contains the header files and dynamic linking libraries(shared libraries) of the toolbox classified by OS and architecture
- etc : Contains the files needed at the time of building the toolbox
- help : Contains the required help files for the toolbox
- tests : Contains the test files for the toolbox
- demo : Contains demo files for the toolbox

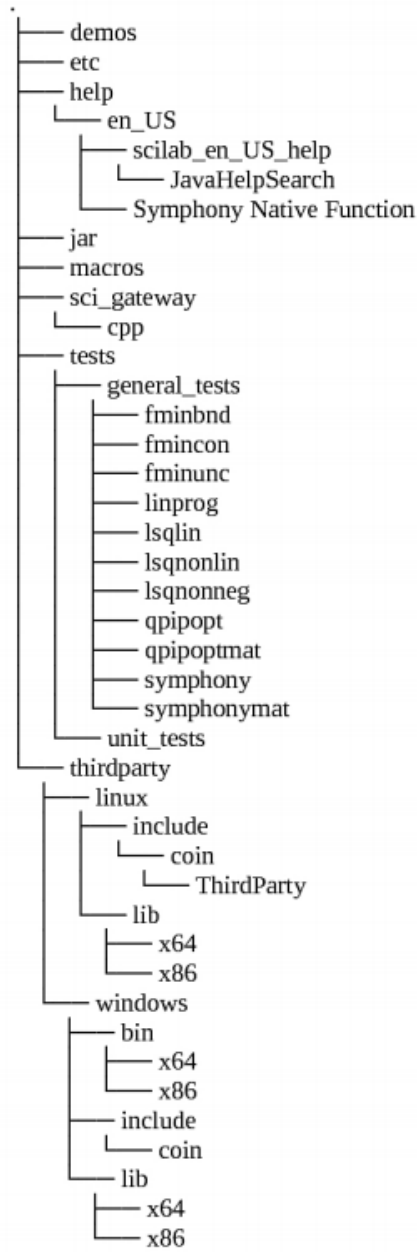


Figure 1.1: FOT Folder Structure Tree

1.3 CUTEst and CUTE set of PrincetonLib

CUTEst is the latest evolution of CUTE, the constrained and unconstrained testing environment for numerical optimization. It stands for constrained and unconstrained testing environment with safe threads. It is a versatile testing environment for optimization and linear algebra solvers. The test problems provided are written in so-called Standard Input Format (SIF). A decoder to convert from this format into well-defined Fortran 77 and data files is available as a separate package. Once translated, these files may be manipulated to provide tools suitable for testing optimization packages. Ready-to-use interfaces to existing packages, such as MINOS, SNOPT, filterSQP, Knitro, and more, are available. CUTEst is available on a variety of UNIX platforms, including Linux and is designed to be accessible and easily manageable on heterogeneous networks.

PrincetonLib is a collection of nonlinear programming (NLP) models. The purpose of the collection is to provide algorithm developers of nonlinear optimization codes with a large and varied set of both theoretical and practical test models. It also aids in the software quality assurance process by providing a set of tools to facilitate benchmarking and performance analysis. The original models are in AMPL format and collected by Robert Vanderbei and colleagues at Princeton University.

Chapter 2

Contribution

The French government space agency(CNES) is an active user of the FOSSEE Optimisation toolbox. They sent a report to the FOSSEE team, outlining a number of areas where improvements could be made. As a FOT Development intern, we were tasked with resolving some of the issues mentioned in the CNES report.

2.1 Task 1

2.1.1 Problem Statement

One of the limitations mentioned in the CNES report was the excess computation time required by the `fmincon` function to solve large optimization problems which include large number variables. Most of the computation time was spent on the function that computed the Hessian. So, they suggested including the "limited-memory" option from the `ipopt` library with the help of which the user could choose between exact or approximate hessian, which in turn will save a lot of computation time.

I was assigned the role of developer for this task. My responsibilities included adding the limited-memory option to the options structure for the `fmincon` function and updating the help document with an example demonstrating the limited-memory option's usage.

2.1.2 Changes Made

The following changes were made in the codebase to achieve the use of "limited memory" option of `Ipopt`:

- The option for choosing the Hessian Approximation has been added in the options struct in the file `sci_gateway/cpp/sci_ipoptfmincon.cpp`. The option is named as `HessianApproximation`, and takes a scalar, either 0 or 1 as an input, as follows:

```
options = struct("MaxIter", [3000], "CpuTime", [600],  
"HessianApproximation", 1)
```

Giving the value 0 passes the value of hessian approximation as "exact" in `Ipopt`, whereas a value of 1 passes the value of hessian approximation as "limited-memory". The line where this occurs (line 205) is as follows:


```
app->Options()->SetStringValue("hessian approximation", ha);  
where ha takes the value "exact" or "limited-memory" accordingly.
```

- The *fot_fmincon.sci* file has been updated to include the description for the newly added HessianApproximation option in the options section of the *fot_fmincon* help document.
- The error message has been added in the *fot_fmincon.sci* file, so that if the user enters any other value for HessianApproximation except 0 and 1, the following message shows up on the Scilab console:

```
fot_fmincon: Value for Hessian Approximation should be either 0  
or 1
```

- A minor error in the previous implementation has been corrected. This error caused the values in the options structure to be re-initialized and set to default if the options parameter was passed as the 5th, 7th or 9th argument in the calling sequence.
- The CUTEst test cases have been added in a separate folder in the general tests folder. These problems have been translated from AMPL to Scilab and such problems have been selected which have a large number of variables and equations.

2.2 Task 2

2.2.1 Problem Statement

In this task we were asked to solve a github issue raised by one of the users of the optimization toolbox. There was a problem with the integer constraint of the *fot_intlinprog* function. The *intcon* parameter of the *fot_intlinprog* function was not working as it was intended to. As a result, the user was not able to get the output in accordance with the passed integer constraints. This was mainly because *numintcons* variable in *matintlinprog/sci_intlinprog_matrixcpp.cpp* was always 0 and never set according to the input passed. We were also instructed to check other functions, such as *fot_intfminbnd*, *fot_intfmincon*, *fot_intfminimax*, *fot_intfminunc*, *fot_intlinprog*, and *fot_intquadprog*, for the same issue.

I was assigned the role of Manager for this task. My responsibilities included:

- Assigning roles to the team members- Developer, Tester, Documenter
- Managing and directing the team
- Organizing meetings
- Reviewing the work of all the team members

2.2.2 Changes Made

The following changes have been made in the codebase to fix the Integer Constraints issue:

- Modifications made in *macros/cbcmatrixintlinprog.sci* file to set the value of numintcons to the size of the intcon vector and pass it to the matintlinprog function:

```
numintcons = int32(size(intcon,1)); //line 229
[xopt, fopt, status, nodes, nfpnts, L, U, niter] =
matintlinprog(int32(nbVar), nbCon, c, intcon, conMatrix, conLB, conUB,
lb, ub, objSense, optval, numintcons); //line 262
```

- Changes made in *sci_gateway/cpp/sci_intlinprog_matrixcpp.cpp* file:

```
if (nin != 12) //Checking the input arguments
{
Scierror(999, "%s:Wrong number of input arguments:%d expected.",
fname, 12);
return STATUS_ERROR;
} //line 49-53

scilab_getInteger32(env, in[11], &numintcons); //line 99
```

2.3 Task 3

2.3.1 Problem Statement

In this task we were asked to verify the code files ,scripts, slides and fix errors (if any) for the spoken tutorials. The Spoken tutorial is a screencast with running commentary, a recording of a computer session created for self-learning. These tutorials are made to demonstrate how to use the various optimization routines offered by the FOSSEE Optimization Toolbox. We were also required to solve and add assignment problems for each of the optimization functions.

For this task, I was assigned the *Quadratic_Optimization_using_quadprog* directory which consisted of all the files for fot_quadprog and fot_intquadprog functions.

2.3.2 Changes Made

The following changes were made:

- Corrected the code files for fot_intquadprog and fot_quadprog functions
- Added an assignment problem for fot_quadprog function
- Solved the assignment problems for fot_quadprog and fot_intquadprog functions
- Updated the slides
- Updated the script

Chapter 3

Significance of limited memory option

As mentioned in the CNES report, `fmincon` function used a substantial amount of computation time in calculating the Hessian of Lagrangean. Also since the Hessian requires second order derivatives, sometimes it may take up an excessive amount of computer memory for storage ($O(n^2)$ in the worst case, where n is the number of variables). For this reason, nonlinear solvers like IPOPT provide a limited memory hessian approximation scheme in which Hessian information is estimated from previous gradients and step lengths rather than using Hessian information.

The following example demonstrates the importance of this option:

```
N=100;
x0 = zeros(2*N+1, 1);
//Linear equality constraints
beq = zeros(N+1, 1);
beq(1) = 1;
Aeq = ones(N+1, 2*N+1);
Aeq(1, 2:2*N+1) = 0;
for i=2:N+1
    if i>=3 then
        Aeq(i, 1:i-2) = 0;
        Aeq(i, N+2:N+i-1) = 0;
    end
    if i>=2 then
        Aeq(i, i-1) = -N-0.5;
    end
    Aeq(i, i) = N-0.5;
    Aeq(i, i+1:N+1) = 0;
    Aeq(i, N+i) = -1;
    Aeq(i, N+i+1:2*N+1) = 0;
end
function y=f(x)
N=100;
sumt = 0 ;
```

```

for i=1:N
sumt = sumt + ((x(N+1+i) ^ 2)/(2*N));
end
y = 0.5*x(N+1) ^ 2 + sumt;
endfunction
function y=fGrad(x)
N=100;
y = zeros(1,2*N+1);
y(1,1:N) = 0;
y(1,N+1) = x(N+1);
for i=1:N
y(1,N+1+i) = x(N+1+i)/N;
end
endfunction
//Hessian approximation turned on
options = struct("MaxIter", [3000], "CpuTime", [600],
"HessianApproximation", 0, "GradObj", fGrad, "Hessian","off",
"GradCon","off" );
[x,fval, exitflag, output] = fmincon(f, x0, [], [], Aeq, beq,
options)

```

The above mentioned is an example of a nonlinear optimization problem with large number of variables($N=100$). As we can see, the HessianApproximation option is set to 0, which implies that it is off. In this case, the function will calculate the hessian and we get the following output:

```

Optimal Solution Found.
--> fval
fval =
0.8807988
--> exitflag
exitflag =
2
0.
--> output
output =
Iterations = 1
Cpu_Time = 356.106
Objective_Evaluation = 2
Dual_Infeasibility = 6.661D-16
Message = "Optimal Solution Found"

```

From the above output, it can be observed that the Cpu_Time is very large. The following output can be obtained by setting the HessianApproximation option to 1 and enabling the HessianApproximation option. The Cpu_Time has been reduced to a great extent while giving the same optimal results.

```
Optimal Solution Found.  
--> fval  
fval =  
0.8807988  
--> exitflag  
exitflag =  
0.  
--> output  
output =  
Iterations = 6  
Cpu_Time = 0.124  
Objective_Evaluation = 7  
Dual_Infeasibility = 0.0000002  
Message = "Optimal Solution Found"
```

Chapter 4

Future Work

The current version of FOT does not support the use of sparse matrices for linear and nonlinear optimization. As a result several of the CUTEst instances cannot even be loaded into FOT. This feature could be added in future versions of FOT.

As mentioned in the CNES report, some important options are missing in the `fmincon` function of FOSSEE Optimization toolbox like:

FiniteDifferenceStepSize: to compute the gradients by finite difference when the user does not provide the analytical gradients.

ConstraintTolerance: specifies the accuracy in which the problem constraints have to be satisfied.

FiniteDifferenceType:: kind of finite difference used when the user does not provide the gradients (forward, centered, etc).

OptimalityTolerance: termination tolerance for the satisfaction of the first-order optimality conditions.

StepTolerance: Termination tolerance on x , a positive scalar

These can be added in the future versions of FOT.

Reference

- <https://github.com/FOSSEE/FOSSEE-Optimization-toolbox>
- <https://scilab.in/fossee-scilab-toolbox/optimization-toolbox/functions/fmincon>
- <https://coin-or.github.io/Ipopt/OPTIONS.html>