System Verification and Validation Test Report: A Library of Simplex Method Solvers (LoSMS)

Hanane Zlitni

December 18, 2018

1 Revision History

Date	**Version**	**Notes**
December 15, 2018	1.0	First draft

2 Symbols, Abbreviations and Acronyms

See System V&V Report found at: https://github.com/hananezlitni/HZ-CAS741-Project/tree/master/docs/VnVReport/SystVnVReport/SystVnVReport.pdf.

Contents

List of Tables

This document reports all results obtained from testing the Library of Simplex Method Solvers (LoSMS) tool. The test cases under evaluation, along with the full documentation of the library, can be found at: https://github.com/hananezlitni/HZ-CAS741-Project

The document starts by outlining the results of the test cases related to the tool's functional requirements in Section 3. Then, the results of the test cases related to the non-functional requirements are reported in Section 4. This is followed by Sections 5, 6 and 7, which discuss comparing any previous implementations with the existing one, unit testing of the library and changes which occured after testing, respectively. Section 8 discusses the tests that were executed automatically, while sections 9 and 10 provide traceability tables for the tests with the requirements and modules. Sections 11 concludes the document by discussing the metrics used to achieve code coverage.

3 Functional Requirements Evaluation

See Section 6.

4 Nonfunctional Requirements Evaluation

This section will report the accuracy of the outputs obtained from executing the test cases in the Unit V&V Plan.

The evaluation of the rest of the qualities were detailed in the System V&V Report found at: https://github.com/hananezlitni/HZ-CAS741-Project/tree/master/docs/VnVReport/SystemVnVReport/SystemVnVReport.pdf.

4.1 Accuracy

T9 in the System V&V Plan tests the accuracy of the solutions produced by LoSMS.

The following table reports the relative errors of the expected outputs for each test case detailed in the Unit V&V Plan that produces numbers.

T	Expected	Actual	Relative Error
T1	Z= 1052000	Z= 1052000.0	$\epsilon_Z = 0$
	x_1= 4	x_1= 4.0	$\epsilon_{x_1} = 0$
	x_2= 10	x_2= 10.0	$\epsilon_{x_2} = 0$
	x_3= 14	x_3= 14.0	$\epsilon_{x_3} = 0$
T2	Z= 180	Z= 180.0	$\epsilon_Z = 0$
	x_1= 40	x_1= 40.0	$\epsilon_{x_1} = 0$
	x_2= 30	x_2= 30.0	$\epsilon_{x_2} = 0$

Table 1: Relative Errors of Expected Outputs

5 Comparison to Existing Implementation

This section is not applicable for LoSMS.

6 Unit Testing

T1 & T2 in the Unit V&V Plan verify the *solveLP()* function for 2 maximization linear programs. The addition of these 2 test cases was intended to have a thorough coverage of the cases. Both test have passed.

T3 in the Unit V&V Plan verifies the *solveLP()* function for a minimization linear program that doesn't have an optimal solution. The test has passed.

The Unit V&V Plan can be found at: `https://github.com/hananezlitni/HZ-CAS741-Project/tree/master/docs/VnVPlan/UnitVnVPlan/UnitVnVPlan.pdf`.

7 Changes Due to Testing

No changes besides what was detailed in the System V&V Report were made.

8 Automated Testing

All for functional requirements were automated using *unittest* provided by Python. The implemented tests are executed by entering a command in the command line tool.

9 Trace to Requirements

Table 2 shows the traceability between the test cases in the System V&V Plan and the requirements.

T	**Requirements**
T1	R1, R4, R5, R6
T2	R1, R4, R5, R6
T3	R1, R4, R5, R6

Table 2: Traceability Between Test Cases and Requirements

10 Trace to Modules

Table 3 shows the traceability between the test cases and the modules.

T	**Modules**
T1	Simplex Method Solver Module
T2	Simplex Method Solver Module
T3	Simplex Method Solver Module, Exceptions Module

Table 3: Traceability Between Test Cases and Modules

11 Code Coverage Metrics

It was important to ensure that the implemented test cases were achieving the highest possible code coverage. To verify this, Python's *Coverage.py* tool was used.

The tested linear program was:

$$\begin{aligned} &\min Z = -2x_1 + 3x_2 \\ &s.\,t. \quad 3x_1 + 4x_2 = 24 \\ &\qquad\; 7x_1 + 4x_2 = 16 \end{aligned}$$

Table 4 reports the code coverage that the test cases achieved:

Module	Statements	Missing	Coverage
Exceptions.py	6	0	100%
SimplexSolverADT.py	131	1	99%
SimplexSolverADT_Test.py	14	0	100%
Total	151	1	99%

Table 4: Code Coverage Results by Coverage.py

References