Commonality Analysis of a Library of Simplex Method Solvers [Put the name of your library in the title —SS]

Hanane Zlitni

October 4, 2018

1 Revision History

Date	Version	Notes
October 13, 2018	1.1	Applied most of the comments obtained from Jennifer Garner's review
October 4, 2018	1.0	First Draft

2 Reference Material

This section records information for easy reference.

2.1 Table of Units

This section is not applicable for LoSMS.

2.2 Table of Symbols

The table that follows summarizes the symbols used in this document.

symbol	unit	description
Z	-	Optimal solution of the objective function
Z'	-	The negation of the objective function
R	-	The set of real numbers
N	-	The set of natural numbers

2.3 Abbreviations and Acronyms

symbol	description
A	Assumption
C	Calculation
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
CA	Commonality Analysis
LoSMS	Library of Simplex Method Solvers
T	Theoretical Model
$s.\ t.$	Subject to

Contents

3 Introduction

The simplex method, a linear programming algorithm, is considered one of the most popular algorithms that has significant influence in the fields of science and engineering (Dongarra and Sullivan (2000)).

[In your introduction it would be nice to say what linear programming actually is. —SS]

The algorithm can be used in a variety of fields and its goal is to make the most of the available resources to achieve the optimal solution. For example, the simplex method is used in the sand casting process to optimize the sand casting parameters to produce the best results (Nadar (2016)). Moreover, the simplex method was used in chemistry to maximize the yield of a chemical reaction (Rozycki (1993)).

Since the simplex method has various applications in different fields, a software that facilitates solving objective functions using the simplex method for different purposes can be useful.

This commonality analysis (CA) provides detailed documentation of a general-purpose program family, called LoSMS (Library of Simplex Method Solvers), that solves linear programming problems using the simplex method. The reason for choosing this specific algorithm is because of its high efficiency, its numerous applications and its influence in various fields including science and engineering. The CA template is based on Smith (2006).

3.1 Purpose of Document

The purpose of this document is to formally describe the requirements for the development of the LoSMS tool which simplifies obtaining the optimal solution of linear programs. Having a thorough and comprehensive documentation of this tool would be useful for future use of LoSMS, possible enhancements and maintenance.

3.2 Scope of the Family

The scope of LoSMS is limited to solving linear equations using the simplex method. The tool supports both maximization and minimization linear programs.

3.3 Characteristics of Intended Reader

The intended reader of this document must have basic knowledge of linear programming which is typically given to year 3 undergraduate students. The intended reader must also have basic knowledge of linear algebra and calculus. No technical background is required.

3.4 Organization of Document

The document begins by providing a general description of the system, which includes potential system contexts, user characteristics and system constraints, in Section 4. Then, Section 5 describes the commonalities in the LoSMS program family by giving a background overview of the tool, terminology definitions, data definitions that are used to build instance models, goal statements and theoretical models. Next, Section 6 details the variabilities in the tool and consists of instance models and assumptions. This is followed by the functional and nonfunctional requirements of the LoSMS tool in Section 7 and likely changes in variabilities in Section 8. Finally, Section 9 visualises the way different sections of this document can be traced to one another.

4 General System Description

This section identifies the interfaces between the system and its environment, describes the potential user characteristics and lists the potential system constraints.

4.1 Potential System Contexts

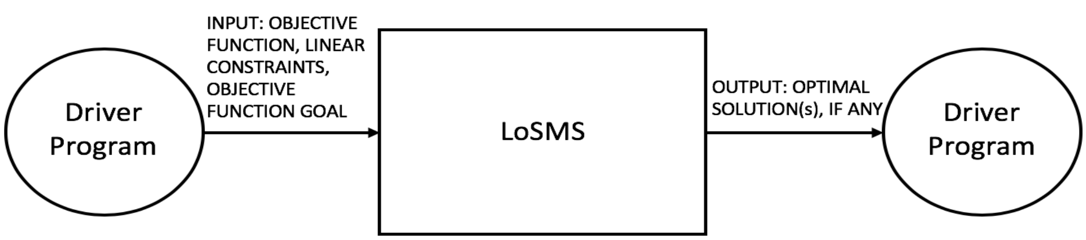

Figure 1: System Context

[Nice to see the drive program is interacting with your library, and not a user. —SS]

Figure 1 describes the system context of LoSMS. The user inputs the correct data and LoSMS displays the output, if any.

- User Responsibilities: [Rather than user responsibilities, you should probably discuss the calling programs responsibilities —SS]
  - Input the objective function, linear constraints and the objective function goal.
- LoSMS Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number.
  - Handle any errors occuring when the user enters the inputs, such as entering inputs in an incorrect format.

- Find and display the linear program's optimal solution(s) (if any).

[You say optimal solution, but it is unclear whether you mean the value of the optimal value of the objective function, or the optimal value and the corresponding values of the decision variables. —SS]

4.2 Potential User Characteristics

The end user of LoSMS should have basic knowledge of linear programming which is typically given to year 3 undergraduate students. [A little more information would be good here. Undergraduate students in what programs? What course subject would typically cover linear programming - operations research? systems? —SS]

4.3 Potential System Constraints

LoSMS does not have any system constraints.

5 Commonalities

This section begins by providing a general idea about the LoSMS tool, followed by terminology and data definitions, goal statements and theoretical models.

5.1 Background Overview

LoSMS is a program family that facilitates obtaining the optimal solution of linear programs given the objective function and linear constraints. The tool can be beneficial for users coming from various fields, including physics and chemistry.

[The background section would be a great place to show the "classic" 2D picture of the simplex method. —SS]

5.2 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Linear program: An optimization problem where the goal is to minimize or maximize the objective function.
- Objective function: The function to be minimized or maximized.
- Objective function goal: The choice of minimizing or maximizing the objective function.

- Linear constraints: Linear constraints are some of the main inputs needed to solve a linear programming problem. They can be equalities or inequalities. There are two types of linear constraints: *main constraints* (e.g. $2x_1 + 3x_2 \leq 10$) and *non-negativity constraints* (e.g. $x_1, x_2 \geq 0$).
- Decision variables: They are variables that represent the parameters that the user wishes to optimize, and they are written as: x_1, x_2, ..., x_k, where $k \in N$.
- Feasible solution: The point that satisfies all constraints and sign restrictions.
- Feasible region: The set of all feasible points.
- Optimal solution/The optimum: A feasible solution with the maximum value in maximization objective functions or the minimum value in minimization objective functions.

5.3 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**The Negation of a Minimization Linear Program**
Symbol	Z'
Equation	$Z' = -Z$
Description	To convert the linear program goal from minimization to maximization, negate the minimization function by multiplying it by -1.
Sources	-
Ref. By	IM2

[It is nice to see that IM2 actually does reference DD1. Good! —SS]

Number	DD2
Label	**The Simplex Tableau**
Symbol	-
Equation	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \dots & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & \dots & b_2 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & \dots & b_n \end{bmatrix}$, where: $a, b \in R$ and $n =$ the row number
Description	The simplex tableau is the augmented matrix form that LoSMS creates from the linear program's equations.
Sources	-
Ref. By	IM1

[I feel like there should be a symbol for the tableau and a source. —SS]

Number	DD3
Label	**The Slack Variable**
Symbol	S_n
Equation	$a_1x_1 + a_2x_2 \leq b_n$ becomes $a_1x_1 + a_2x_2 + S_n = b_n$
Description	The slack variable is a variable that represents zero or a positive real number. It is added to less than or equal to inequalities so they become equalities.
Sources	Stacho (2014)
Ref. By	IM1

5.4 Goal Statements

Given the objective function, linear constraints and the objective function goal, the goal statement of LoSMS is:

GS1: Use the simplex method to find and display the objective function's optimal solution(s) satisfying all linear constraints and sign restrictions.

[Rather than display the optimal solution, say "output" the optimal solution. This is a more abstract way to say it. —SS] [As mentioned previously, you should be clear on what you mean by the optimal solution. —SS]

5.5 Theoretical Models

This section focuses on the general equations and laws that LoSMS is based on.

Number	T1
Label	**The Standard Form of a Linear Program**
Equation	$$\begin{cases} \quad max\ Z = \quad c_1x_1 + c_2x_2 + ... + c_kx_k \\ s.\ t. \quad a_{11}x_1 + a_{12}x_2 + ... + a_{1m}x_k \leq b_1 \\ \quad \vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots \\ \quad a_{n1}x_1 + a_{n2}x_2 + ... + a_{nm}x_k \leq b_n \\ \quad x_1, x_2 + ... + x_k \geq 0 \end{cases}$$, where: $c, a, b \in R$; $k \in N$; $n =$ the row number ; $m =$ the column number [a and b should have the subscripts. Without the subscripts, they would not have the type real, but multidimensional sequences of real. —SS]
Description	A linear program is in its standard form when it satisfies the following conditions: • The objective function is a maximization function • All constraints are inequalities • All decision variables are greater than or equal to zero.
Source	Stacho (2014)
Ref. By	IM1, R4

[The names of functions, like max, should not be in italic font. You can use the following LaTeXcode to change fonts inside an equation: max $Z = ...$ —SS] [Isn't Z usually in lower case in this formula? —SS]

[Do you want to consider using matrix notation for these equations? I guess it depends on how many times you need the equation. If you are using it frequently, the more succinct matrix notation would help. —SS]

Number	T2
Label	**The Canonical Form of a Standard Linear Program**
Equation	$$\begin{cases} max\ Z = & c_1x_1 + c_2x_2 + ... + c_kx_k \\ s.\ t. & a_{11}x_1 + a_{12}x_2 + ... + a_{1m}x_k \ = \ b_1 \\ & \vdots \quad \vdots \quad \vdots \quad \vdots \\ & a_{n1}x_1 + a_{n2}x_2 + ... + a_{nm}x_k \ = \ b_n \\ & x_1, x_2 + ... + x_k \geq 0 \end{cases}$$, where: $c, a, b \in R$; $k \in N$; $n =$ the row number ; $m =$ the column number
Description	A standard linear program is in its canonical form when it satisfies the conditions in T1, except: • The constraints must be equality constraints and not inequalities
Source	Stacho (2014)
Ref. By	IM1, R5

[Isn't this where you want to mention slack variables? —SS]

Number	T3
Label	**Pivoting in an Augmented Matrix**
Equation	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & b_1 \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & b_n \end{bmatrix}$, where: $a \in R$ Let R_i and R_j be any two rows in the matrix, $i, j \in N$: Pivoting includes one or more of the following operations: • Row switching: $R_i \leftrightarrow R_j$ • Row addition: $R_i + R_j$ • Multiplying a row by a non-zero constant: kR_i ; $k \neq 0$
Description	Pivoting in an augmented matrix consists of a series of row operations that are done to clear a column in the matrix (set elements of the column to zero).
Source	-
Ref. By	IM1

[The information on T3 doesn't actually tell me what pivoting is. You have just defined elementary row operations, not why you would do them. —SS]

6 Variabilities

This section describes the variabilities in the LoSMS tool and details the instance models, assumptions and variabilities in the calculation and the output.

6.1 Instance Models

This section transforms the documented problem into one which is expressed in mathematical terms.

<table>
<tr><th>Number</th><th>IM1</th></tr>
<tr><td>Label</td><td>The Simplex Method for Solving Linear Programs: Maximization Functions</td></tr>
<tr><td>Input</td><td>1. Max objective function in the following form:
$max\ Z = c_1x_1 + c_2x_2 + ... + c_kx_k$

2. Linear constraint(s) in the following form:
$a_{11}x_1 + a_{12}x_2 + ... + a_{nm}x_k \leq b_n$
$x_1, ..., x_k \geq 0$

, where: $c, a, b \in R$; $k, n, m \in N$</td></tr>
<tr><td>Output</td><td>The optimal solution Z [I'm surprised you just want Z. Don't you also want the x_i values that give you Z? —SS]</td></tr>
<tr><td>Description</td><td>The purpose of this instance model is to provide details about solving linear programs that intend to maximize a parameter. The steps to solve a maximization problem are:

1. Convert linear program to its canonical form using slack variables.
2. Set up the simplex method tableau.
3. Perform pivoting.
4. Set up the new simplex tableau.
5. Repeat steps 2-4 until there are no negative numbers in the bottom row of the tableau.
6. The optimal solution Z is found in the basic feasible solution derived from the final tableau.</td></tr>
<tr><td>Sources</td><td>Stacho (2014)</td></tr>
<tr><td>Ref. By</td><td>IM2, C1, R6</td></tr>
</table>

Number	IM2
Label	**The Simplex Method for Solving Linear Programs: Minimization Functions**
Input	1. Min objective function in the following form: $min\ Z = \ c_1x_1 + c_2x_2 + ... + c_kx_k$ 2. Linear constraint(s) in the following form: $a_{11}x_1 + a_{12}x_2 + ... + a_{nm}x_k \leq b_n$ $x_1, ..., x_k \geq 0$, where: $c, a, b \in R$; $k, n, m \in N$
Output	The optimal solution Z
Description	The purpose of this instance model is to provide details about solving linear programs that intend to minimize a parameter. The steps to solve a minimization problem are: 1. Convert the minimization linear program to a maximization linear program by finding Z' defined in DD1. 2. Solve IM1.
Sources	Stacho (2014)
Ref. By	C1, R6

[You have missed the notion of binding time. You could have family members where the number of equations are fixed at design time. This information could be hard-coded into a family member. I don't think you need to add this now, but you could clarify that the size variabilities are all left to run-time. —SS]

[You could have mentioned the "covering" variation (see the Wikipedia page for Linear Programming. —SS]

Derivation of the Simplex Method

The origin of the simplex method is detailed in Dantzig (1987).

6.2 Assumptions

A1: The objective function $Z \in R$.

A2: If the linear constraints are inequalities, they are of type less than or equal to. This is to ensure that there are no negative variables.

6.3 Calculation

C1: Solve IM1 for maximization problems and IM2 for minimization problems.

[It would be better if you phrased these as variabilities. You could also have mentioned the variabilities I mentioned above. —SS]

6.4 Output

All variabilities have the same output. For example, the following inputs:

$$
\begin{aligned}
&max\ Z = 3x_1 + 4x_2 \\
&\quad s.\,t.\ \ x_1 + x_2 \ \leq\ 4 \\
&\qquad 2x_1 + x_2 \ \leq\ 5 \\
&\qquad x_1, x_2 \ \geq\ 0
\end{aligned}
$$

yield the output: $Z = 16$, which is the optimal solution.

[You have more output variabilities than this. You have the decision on whether the output includes the x values, or not. You could output to the screen, or to a file, or to memory. The format of the output could be a variability. —SS]

7 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

7.1 Functional Requirements

R1: The LoSMS tool shall read the objective function, linear constraints and the objective function goal from the user.

R2: The LoSMS tool shall verify that all inputs are valid and satisfy A1 and A2.

R3: If there are invalid inputs, the LoSMS tool shall display a corresponding message to the user.

R4: The LoSMS tool shall convert the objective function to its standard form (T1).

R5: The LoSMS tool shall convert the standard objective function to its canonical form (T2).

R6: The LoSMS tool shall find the optimal solution(s) of the linear program by solving IM1 or IM2, depending on the objective function goal.

R7: The LoSMS tool shall display the optimal solution(s) to the user, if any.

R8: If the given linear program does not have any optimal solutions, the LoSMS tool shall display a corresponding message to the user.

7.2 Nonfunctional Requirements

Usability

NFR1: It shall take at most 10 minutes for the user to learn to use the LoSMS tool.

Robustness

NFR2: If the LoSMS tool encounters an unexpected behavior, the tool shall display a corresponding message to the user in no longer than 2 minutes.

Portability

NFR3: The LoSMS tool shall be operable in at least Windows and Mac platforms.

8 Likely Changes

LC1: The support for greater than or equal to inequalities in the linear constraints.

LC2: The support for additional linear programming algorithms. [You could name some other algorithms, like the criss-cross algorithm, or interior point methods. —SS]

9 Traceability Matrices and Graphs

The purpose of a traceability matrix is to visualise the way different components of this CA are dependent on one another. Every time a component in a row is changed, the component in the corresponding column marked with an "X" may have to be changed as well. Table 1 shows the traceability between the data definitions, theoretical models, instance models, assumptions and the calculation.

	DD1	DD2	DD3	T1	T2	T3	IM1	IM2	A1	A2	C1
DD1								X			
DD2							X				
DD3							X				
T1							X				
T2							X				
T3							X				
IM1								X			X
IM2											X
A1											
A2							X				
C1											

Table 1: Traceability Matrix Showing the Dependencies between Components of this CA

References

George B. Dantzig. Origins of the simplex method. Technical report, Stanford University, 1987.

Jack Dongarra and Francis Sullivan. Guest editors introduction to the top 10 algorithms. *Computing in Science and Engineering*, 2(1):22, 2000.

Divya K. Nadar. Some applications of simplex method. *International Journal of Engineering Research and Reviews*, 4(1):60–63, 2016.

C. Rozycki. Application of the simplex method for optimization of the analytical methods. *Chem. Anal. (Warsaw)*, 1993.

W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

Juraj Stacho. Introduction to operations research- deterministic models, 2014. URL http://www.cs.toronto.edu/~stacho/public/IEOR4004-notes1.pdf.

10 Appendix

This section provides additional content related to this commonality analysis.

10.1 Symbolic Parameters

There are no symbolic parameters used in this document.