Constrained Optimization:
Linear Programming

AE 6310: Optimization for the Design of Engineered Systems

Spring 2017

Dr. Glenn Lightsey

Lecture Notes Developed By Dr. Brian German

Georgia Institute of Technology

DANIEL GUGGENHEIM SCHOOL OF AEROSPACE ENGINEERING

"Standard Form" of a Linear Programming Problem

Minimize:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} c_i x_i$$

Subject to:

$$\sum_{i=1}^{n} a_{ji} x_i = b_j, \qquad j = 1, \ldots, m$$

$$x_i \geq 0, \qquad i = 1, \ldots, n$$

Conversion of a Linear Problem to Standard Form

Standard form may appear to be very restrictive because it would appear that it does not allow for:

- Negative design variables
- Inequality constraints

However, it turns out that we can convert arbitrary linear problems to standard form rather easily.

Conversion of a Linear Problem to Standard Form

Let's say that our original design variables *can* be negative. What can we do?

To write the problem in standard form that obeys the nonnegativity constraints, we have (at least) two options:

- Write each original variable as difference between two new nonnegative variables
- Add a constant to each design variable such that the new variables never become negative

Conversion of a Linear Problem to Standard Form

Note that any finite real number can be written as a difference between two nonnegative finite real numbers.

Let's say our original variables x_i have no restrictions on sign. We can write each x_i as,

$$x_i = x_i' - x_i'', \qquad i = 1, \ldots, n$$

with $x_i' \geq 0$ and $x_i'' \geq 0$. The problem can then be written in standard form in terms of the new variables x_i' and x_i''.

The drawback of this approach is that it doubles the number of design variables in the problem.

Conversion of a Linear Problem to Standard Form

The alternate approach of adding a constant keeps the dimensionality of the problem the same.

The approach is to select an appropriate value of $Q_i > 0$ and to introduce new variables x_i' as,

$$x_i' = x_i + Q_i$$

We could just pick a <u>*very*</u> large Q_i, i.e. $Q_i \to \infty$; however, this approach results in poor numerical behavior in LP algorithms.

A better approach is to select $Q_i = -x_{i,L}$ where $x_{i,L} < 0$ is an estimate of the lower bound of $x_{i,L}$ that we would expect to occur in typical problems.

Conversion of a Linear Problem to Standard Form

What if our original problem has inequality constraints?

We can address this problem by introducing additional variables to the problem called *slack variables.*

Slack variables are so named because they “take up the slack” of an inactive inequality constraint and create an equivalent active equality constraint.

Conversion of a Linear Problem to Standard Form

Presume that the original inequality constraints are of the form,

$$\sum_{i=1}^{n} a_{ji} x_i \leq b_j$$

We then write this as,

$$\sum_{i=1}^{n} a_{ji} x_i + x_{n+j} = b_j$$

where x_{n+j} is the slack variable. Note that there are as many slack variables as there are inequality constraints, with just one slack variable appearing in each constraint equation.

Standard Matrix Form of a Linear Program

Linear programming problems can be written in standard matrix form as,

Minimize:

$$f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x}$$

Subject to:

$$A\boldsymbol{x} = \boldsymbol{b}$$

$$\boldsymbol{x} \geq \boldsymbol{0}$$

where $\boldsymbol{x}$ is the $n \times 1$ vector of design variables, $\boldsymbol{c}$ is an $n \times 1$ vector of "cost coefficients" , $\boldsymbol{b}$ is a $m \times 1$ vector of equality constraints and A is an $m \times n$ matrix.

Standard Matrix Form of a Linear Program

Let's look a bit more at the equality constraint equation,

$$A\boldsymbol{x} = \boldsymbol{b}$$

where $\boldsymbol{x}$ is $n \times 1$, $\boldsymbol{c}$ is $n \times 1$, $\boldsymbol{b}$ is a $m \times 1$, and A is $m \times n$.

We cannot invert A to find $\boldsymbol{x}$ unless $m = n$ and A is nonsingular.

But, if these conditions held and we *could* invert A to find a unique $\boldsymbol{x}$. There would then be no need for optimization; the problem would be just to solve the linear system.

Standard Matrix Form of a Linear Program

Optimization is useful for cases in which $m < n$, i.e. when we have "extra degrees of freedom" in setting the design variables.

By asking to minimize the objective function subject to the m constraints, we are specifying a unique way to lock down these extra degrees of freedom.

This strong analogy of linear programs with linear systems of equations implies that many LP algorithms apply similar linear algebra techniques to those used for solving linear systems of equations.

Word of Caution about the “Standard Form”

The form on the previous slide is Vanderplaats description of standard form.

Other authors call Vanderplaats’ “standard form” the *augmented form* or *slack form* when the inequality constraints are converted to equality constraints with slack variables.

Others authors consider a form involving inequality constraints written out directly as the standard form.

Possible Solutions to Linear Programming Problems

There are four types of possible outcomes to solving LP problems:

1. Unique solution
2. Non-unique solution
3. Unbounded solution
4. No solution

Possible Solutions to Linear Programming Problems

As a basis for discussing these types of solutions, consider the following example problem:

Minimize:

$$f(x_1, x_2) = -4x_1 - x_2 + 50$$

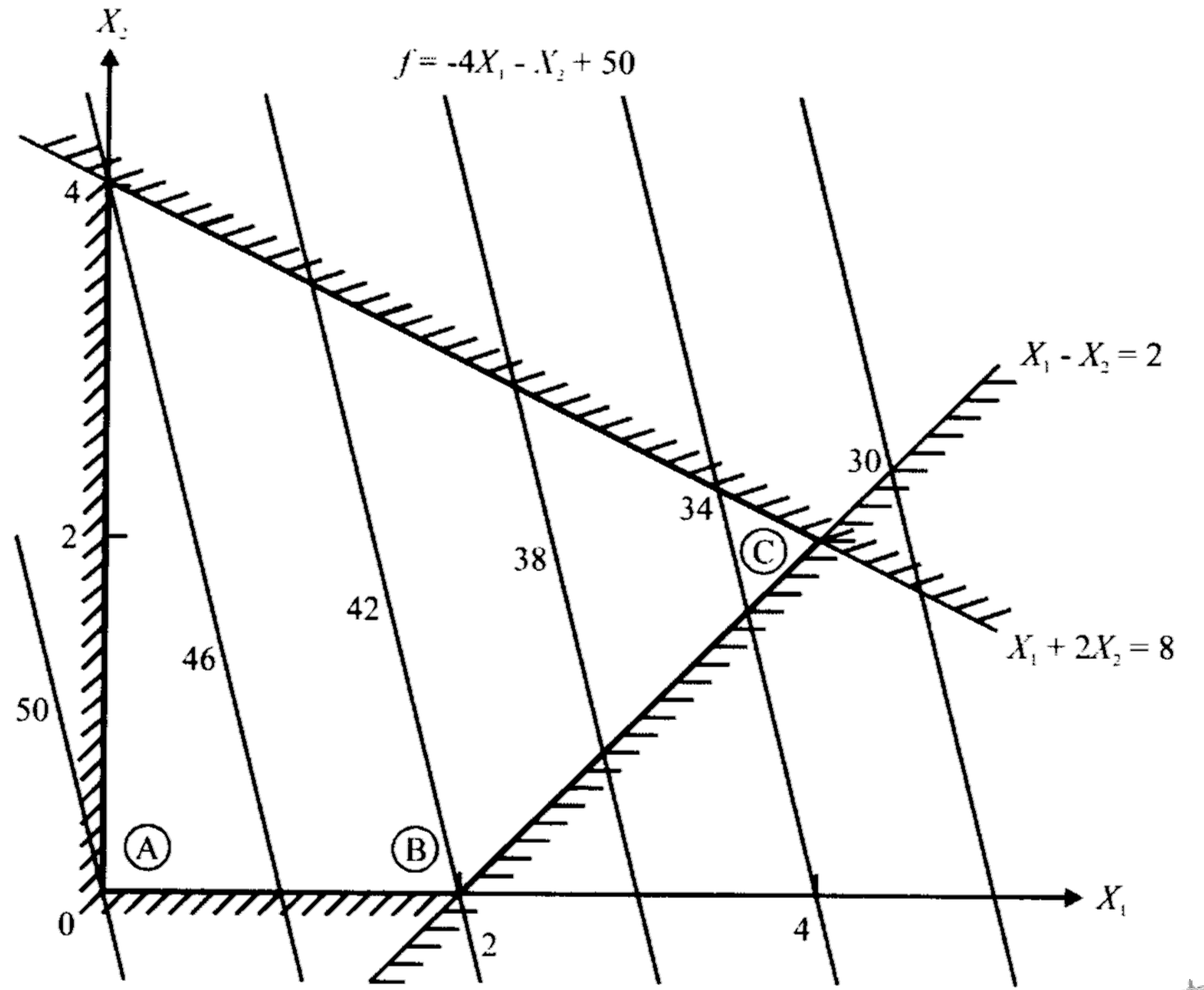
Subject to:

$$x_1 - x_2 \leq 2$$
$$x_1 + 2x_2 \leq 8$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

Vanderplaats, p. 127

Possible Solutions to Linear Programming Problems

Unique solution:

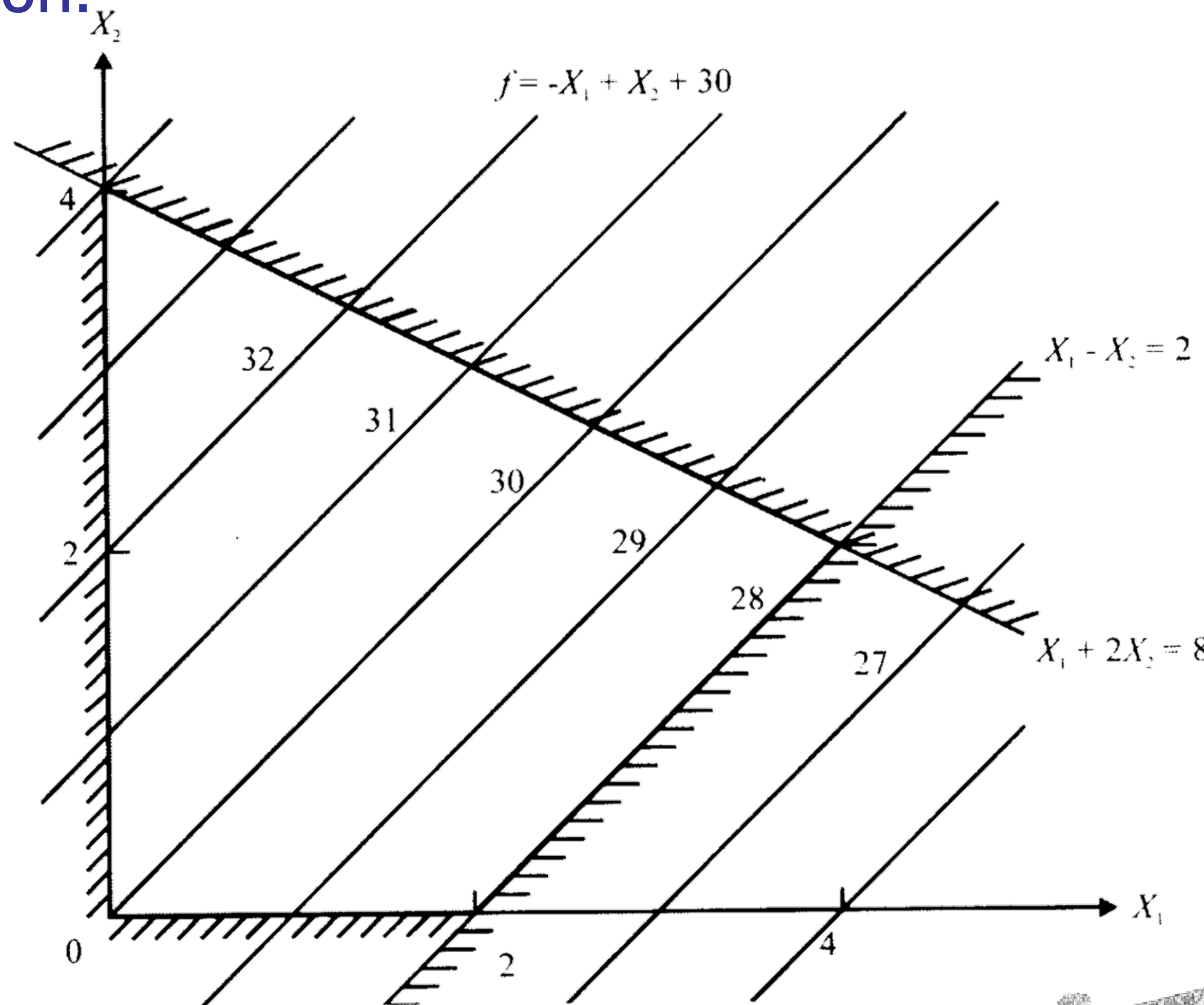


Vanderplaats, Fig. 4.1

Possible Solutions to Linear Programming Problems

Non-unique solution:

Imagine we replace f with $f = -x_1 + x_2 + 30$

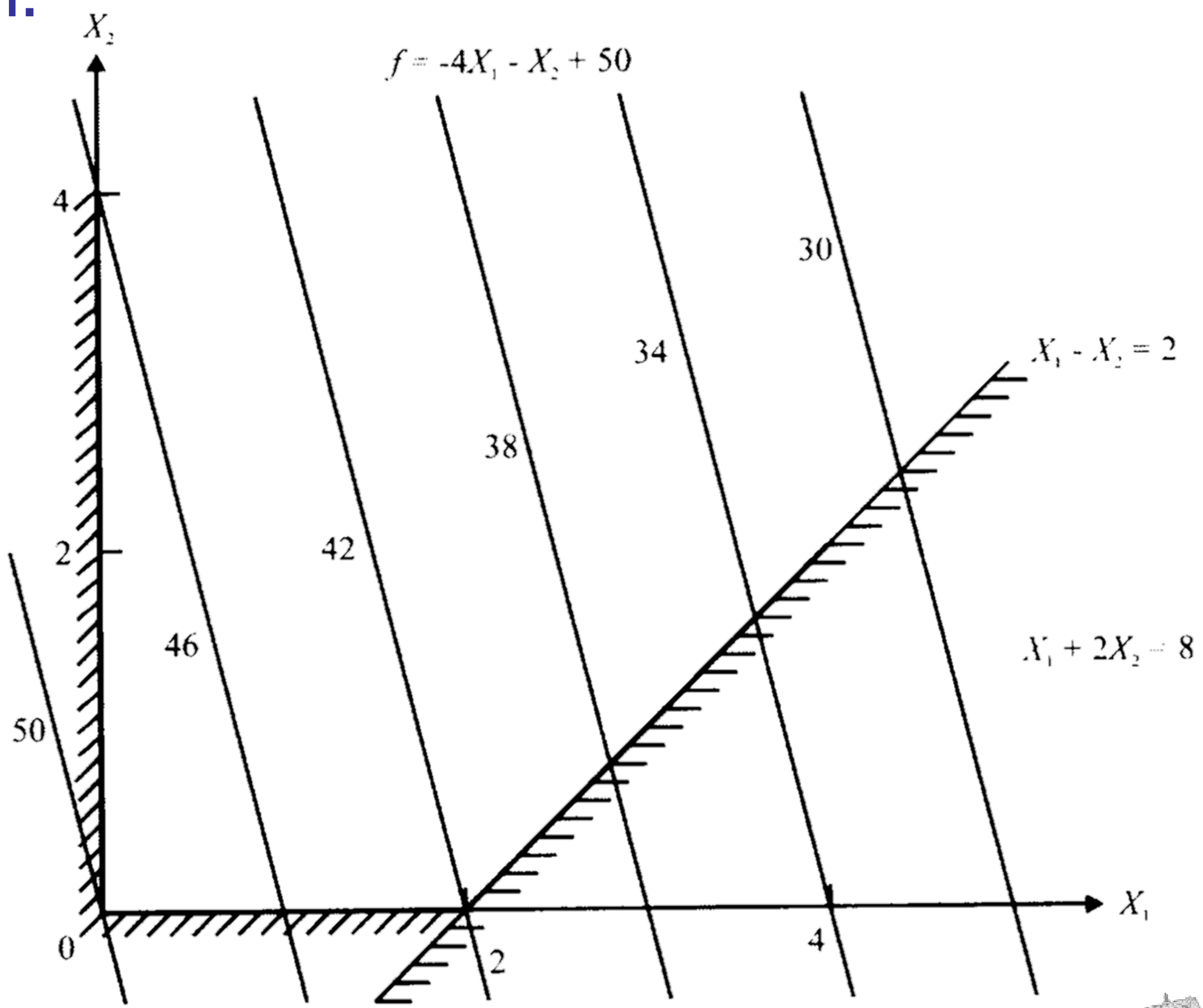


Vanderplaats, Fig. 4.2

Possible Solutions to Linear Programming Problems

Unbounded solution:

Imagine we omit the constraint
$x_1 + 2x_2 \leq 8$



Vanderplaats, Fig. 4.3

Possible Solutions to Linear Programming Problems

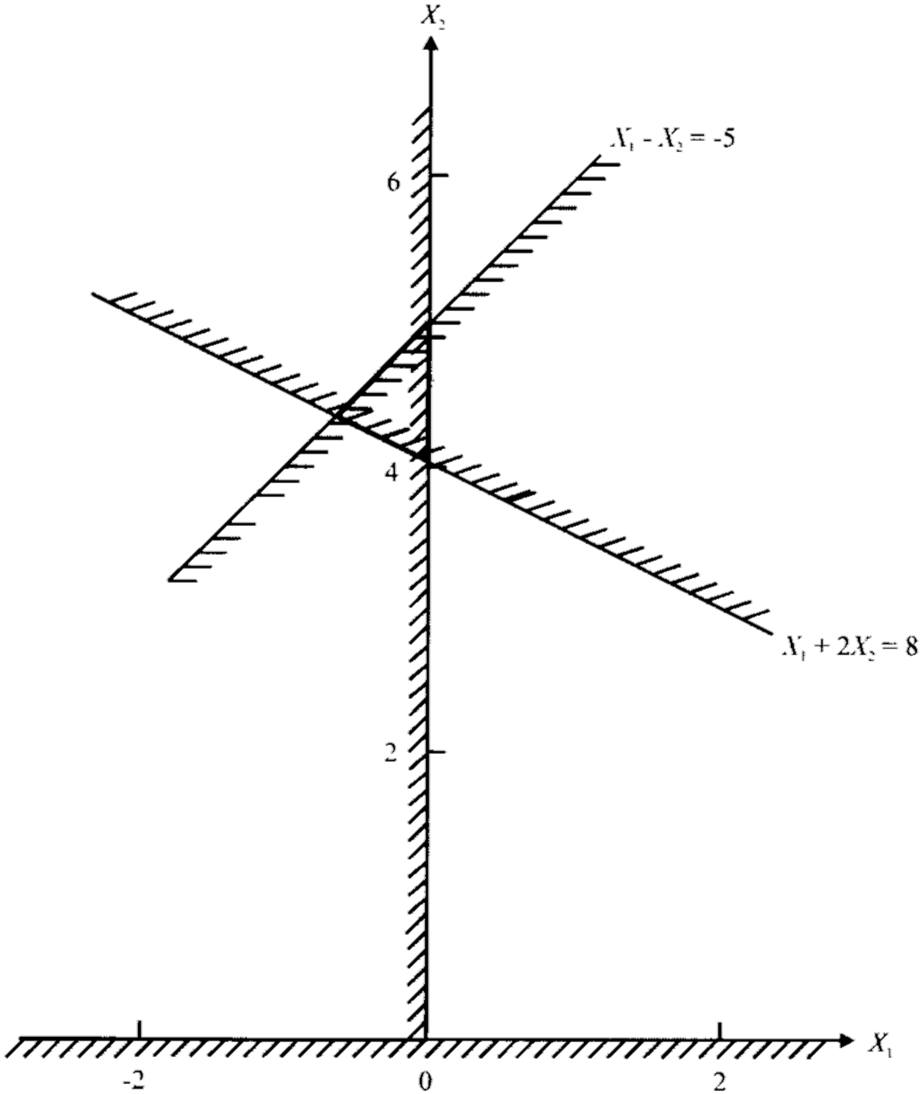
No feasible solution:

Imagine we replace the constraint,

$$x_1 - x_2 \leq 2$$

with,

$$x_1 - x_2 \leq -5$$



Vanderplaats, Fig. 4.4

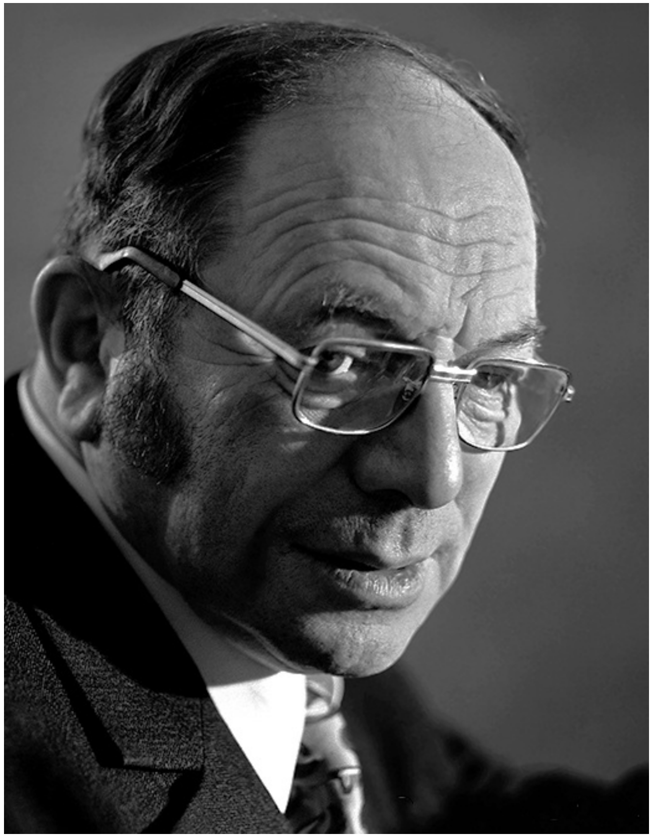
History of Linear Programming

Linear programming has roots in the fields of logistics, operations research, and economics.

The original methods and applications for linear programming were developed by the Soviet economist Leonid Kantorovich in 1939 for military planning problems in WWII.

Kantorovich later won the Nobel Prize in Economics in 1975.

(http://en.wikipedia.org/wiki/Leonid_Kantorovich)



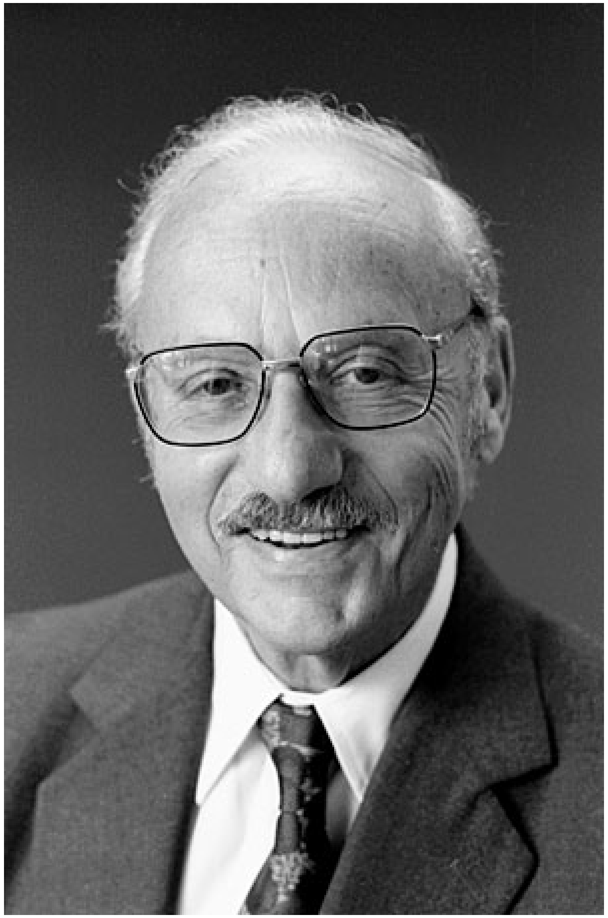
http://en.wikipedia.org/wiki/File:Leonid_Kantorovich_1975.jpg

History of Linear Programming

George Dantzig developed and published the *simplex method* for solving linear programs in 1947.

We will examine this method in detail.

Dantzig's first application was to find the best way to assign 70 people to 70 jobs. The combinatorics of this problem are vast, yet the simplex method arrives at a solution very efficiently.



http://news.stanford.edu/news/2006/june7/memldant-060706.html

Methods for Solving LP Problems

Two major classes of algorithms for solving LPs are as follows:

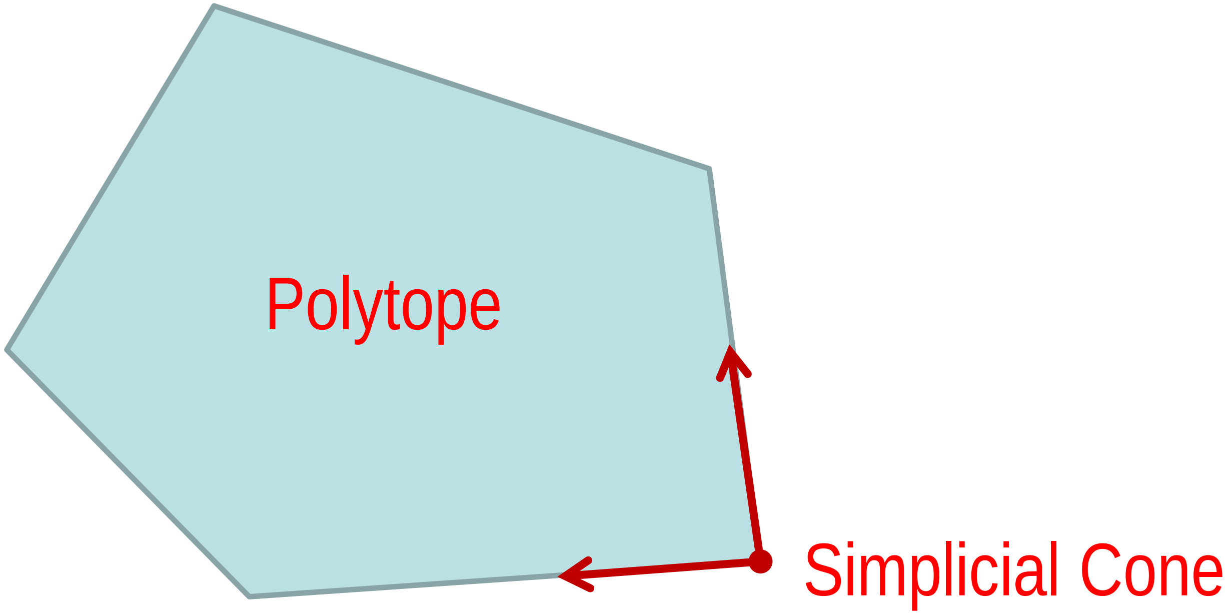
- Basis exchange methods, e.g. the simplex method
- Interior point methods, e.g. Karmarkar's algorithm

Each of these classes of methods, and particular methods within each class, have advantages and disadvantages based on the features and scale of particular problems.

We will examine only the simplex method in detail.

The Simplex Method

The simplex method takes its name from the geometry of the linear programming problem. The set of inequality constraints form a *polytope* and vectors drawn from each vertex of the polytope along the constraint lines form a *simplicial cone.*



The Simplex Method

The simplex method works in two phases:

- Phase I identifies an initial *basic feasible solution*. This solution has m nonzero design variables (the same as the number of equality constraints).

- Phase II is a process to move from one basic feasible solution to another, from vertex to vertex of the polytope, until the optimal solution is found.

Constrained Optimization: The Simplex Method

AE 6310: Optimization for the Design of Engineered Systems

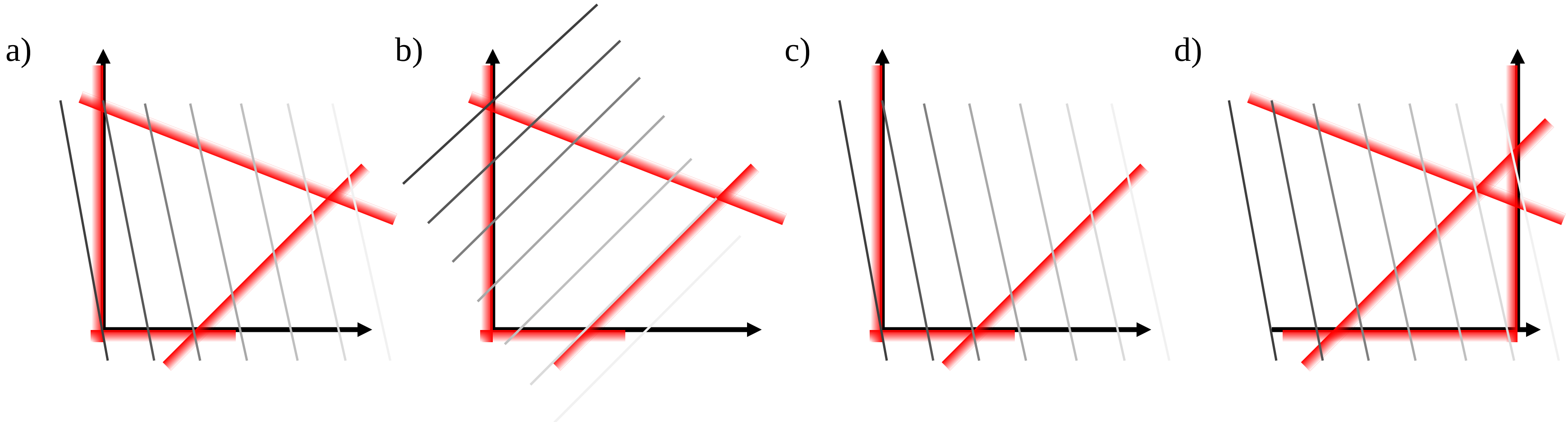
Spring 2017

Dr. Glenn Lightsey

Lecture Notes Developed By Dr. Brian German

Solutions to LP Problems

- There are four possible solutions to a linear programming problem: a) Unique, b) Non-unique, c) Unbounded, and d) No feasible solution.



- A finite solution, if it exists, will always lie on a vertex of the feasible space or on an edge between two vertices.

The Simplex Method

- The simplex method finds the optimum of a linear programming problem by systematically examining the vertices of the feasible space.
- There are two phases to the Simplex method:
  - Phase I: find an initial feasible solution by achieving *canonical form.*
  - Phase II: move from one feasible solution to another until the optimum is found.
- Before going into detail on each phase, we must first understand the simplex tableau and canonical form.

Creating the Simplex Tableau

- We will use a simple problem to demonstrate:

Minimize: $f = -4X_1 - X_2 + 50$

Subject to:

$$X_1 - X_2 \leq 2$$
$$X_1 + 2X_2 \leq 8$$
$$X_1 \geq 0$$
$$X_2 \geq 0$$

- Step 1: convert the problem formulation into LP standard form. In this case, slack variables X_3 and X_4 are added to form equality constraints. Side constraints remain.

Minimize: $f = -4X_1 - X_2 + 50$

Subject to:

$$X_1 - X_2 + X_3 = 2$$
$$X_1 + 2X_2 + X_4 = 8$$
$$X_1 \geq 0$$
$$X_2 \geq 0$$

Question: what would be different if there were $\geq$ constraints?

Creating the Simplex Tableau

- Step 2: create a tableau with:
  - one column for each variable
  - one column for the constants
  - one row for each non-side constraint
  - one row for the objective function

Headers and ='s added for convenience

X_1	X_2	X_3	X_4		b
				=	
				=	
				=	

Minimize: $f = -4X_1 - X_2 + 50$

Subject to:

$X_1 - X_2 + X_3 = 2$

$X_1 + 2X_2 + X_4 = 8$

$X_1 \geq 0$

$X_2 \geq 0$

Creating the Simplex Tableau

- Step 3a: fill in the constraint coefficients ($a_{i,j}$).
- Step 3b: fill in the constants (b_i).
- Step 3c: fill in the cost coefficients (c_j).
- Step 3d: fill in f-f_0.

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
1	2	0	1	=	8
-4	-1	0	0	=	f-50

Minimize: $f - 50 = -4X_1 + -1X_2$

Subject to:
$1X_1 + -1X_2 + 1X_3 = 2$
$1X_1 + 2X_2 + 1X_4 = 8$
$X_1 \geq 0$
$X_2 \geq 0$

Here, *i* subscripts represent the row and *j* subscripts represent the column.

What is Canonical Form?

- Canonical form is achieved when all constants are nonnegative and there exist *m* basic variables, where *m* is the number of non-side constraints.
- A basic variable is one whose column consists of one 1 with the rest being 0 including the cost coefficient. We denote these with black arrows.

X_1	X_2	X_3 ↓	X_4 ↓		b
1	-1	1	0	=	2
1	2	0	1	=	8
-4	-1	0	0	=	f-50

Note: while the columns of basic variables do not need to be side-by-side, when concatenated together, their constraint coefficients should be able to form a $m \times m$ identity matrix.

What is Canonical Form?

- In this case, the initial tableau is already in canonical form. Being in canonical form means a feasible (not necessarily optimal) point has been found.
- **When in canonical form, it is implied that the values of non-basic variables are zero.** Thus, the feasible point found here is $f = 50$ at $(X_1, X_2, X_3, X_4) = (0, 0, 2, 8)$.

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
1	2	0	1	=	8
-4	-1	0	0	=	f-50

$$1X_1 - 1X_2 + 1X_3 + 0X_4 = 2$$
$$1X_1 + 2X_2 + 0X_3 + 1X_4 = 8$$
$$-4X_1 - 1X_2 + 0X_3 + 0X_4 = f - 50$$

Phase II

- Phase I ended with the achievement of canonical form. Phase II now begins with determining whether the feasible point is optimal.
- The existence of a negative cost coefficient signifies that the current point is not optimal. In this case, *f* can be reduced by increasing either X_1 or X_2.
- This requires making one a basic variable.

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
1	2	0	1	=	8
-4	-1	0	0	=	f-50

$$-4X_1 - 1X_2 + 0X_3 + 0X_4 = f - 50$$

Phase II

- Make X_1 a basic variable because it has the largest negative cost coefficient. This choice is denoted by the square.
- Which basic variable should X_1 replace? That is, which $a_{i,1}$ should be the "1"? The rule is: **choose the row with the lowest value $b_i/a_{i,j}$ for all positive $a_{i,j}$** (here, $j=1$).
- X_3 is replaced. This choice is denoted by the circle.

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
1	2	0	1	=	8
$\boxed{-4}$	-1	0	0	=	$f-50$

In zeroing out these two elements, these elements will no longer be 0, so X_3 will no longer be a basic variable.

Phase II

- Change in basic variables is accomplished via elementary row operations:
  - [Row II] = [Row II] – [Row I]
  - [Row III] = [Row III] + 4×[Row I]
- The result is already in canonical form again. The current point is $f = 42$ at $(X_1, X_2, X_3, X_4) = (2, 0, 0, 6)$.

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
1	2	0	1	=	8
$\boxed{-4}$	-1	0	0	=	f-50

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
0	3	-1	1	=	6
0	-5	4	0	=	f-42

Question: what would happen if we chose to replace X_4 instead of X_3?

Phase II

- X_2 still has a negative cost coefficient. Following the same principles, we choose X_2 to replace X_4 as a basic variable.
- Row operations performed:
  - [Row II] = [Row II] / 3
  - [Row I] = [Row I] + [Row II]
  - [Row III] = [Row III] + 5×[Row II]
- The current point is $f = 32$ at $(X_1, X_2, X_3, X_4) = (4, 2, 0, 0)$.

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
0	3	-1	1	=	6
0	$\boxed{-5}$	4	0	=	f-42

X_1	X_2	X_3	X_4		b
1	0	2/3	1/3	=	4
0	1	-1/3	1/3	=	2
0	0	7/3	5/3	=	f-32

Question: X_2 cannot replace X_1 as a basic variable because $a_{2,1}$ is negative. Why?

Phase II

- The tableau is in canonical form and all cost coefficients outside the basic variables are nonzero and positive. This means the current point is a unique optimum.
- This concludes Phase II and the Simplex method.

X_1	X_2	X_3	X_4		b
1	0	2/3	1/3	=	4
0	1	-1/3	1/3	=	2
0	0	7/3	5/3	=	f-32

- But what did all that math really do…?

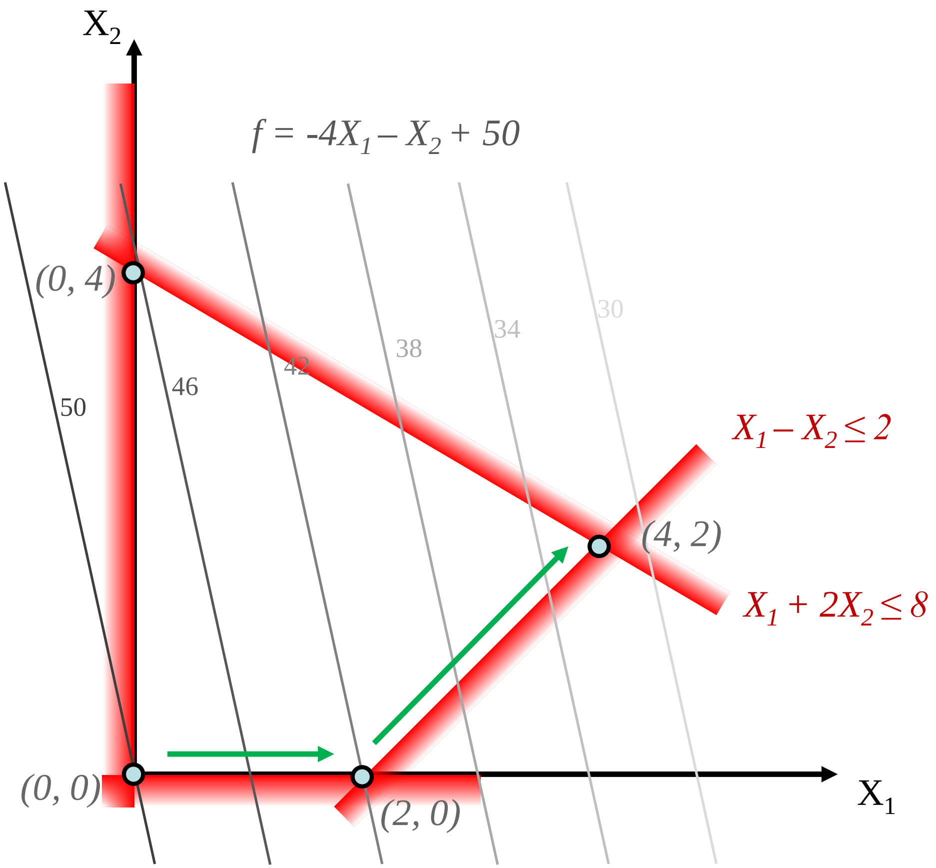
Phase II

- List of all points visited:
  1. $(X_1, X_2, X_3, X_4) = (0, 0, 2, 8)$
     $f = 50$
  2. $(X_1, X_2, X_3, X_4) = (2, 0, 0, 6)$
     $f = 42$
  3. $(X_1, X_2, X_3, X_4) = (4, 2, 0, 0)$
     $f = 32$

Minimize: $f = -4X_1 - X_2 + 50$

Subject to:

$$X_1 - X_2 \leq 2$$
$$X_1 + 2X_2 \leq 8$$
$$X_1 \geq 0$$
$$X_2 \geq 0$$



Nonunique Solution Example

- The contours of the objective function are now parallel with one of the constraints.

Minimize: $f = -X_1 + X_2 + 30$

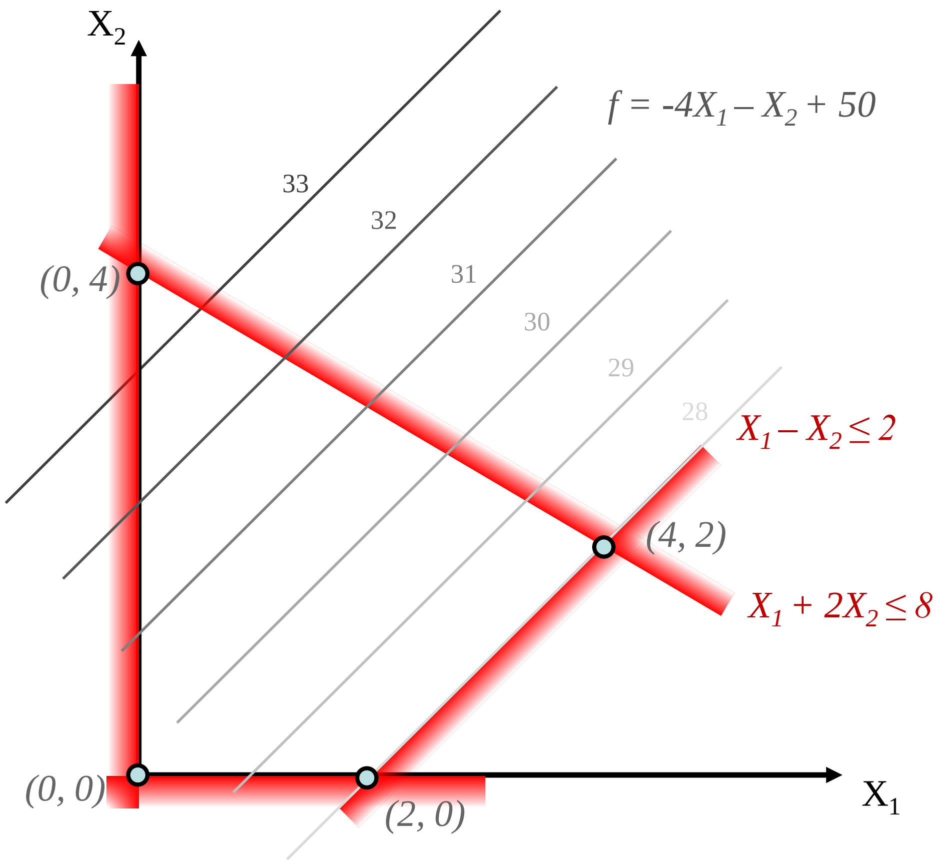
Subject to:

$X_1 - X_2 \leq 2$

$X_1 + 2X_2 \leq 8$

$X_1 \geq 0$

$X_2 \geq 0$



Nonunique Solution Example

Minimize: $f = -X_1 + X_2 + 30$

Subject to:

$$X_1 - X_2 \leq 2$$
$$X_1 + 2X_2 \leq 8$$
$$X_1 \geq 0$$
$$X_2 \geq 0$$

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
1	2	0	1	=	8
$\boxed{-1}$	1	0	0	=	f-30

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
0	3	-1	1	=	6
0	0	1	0	=	f-28

- Initial tableau is in canonical form so Phase I is complete.
- Row operations:
  - [Row II] = [Row II] – [Row I]
  - [Row III] = [Row III] + [Row I]

Nonunique Solution Example

- No more negative cost coefficients and tableau is in canonical form so Phase II is complete.
- However, there is a 0 in the cost coefficients (excluding those in the basic variables).
- This means the solution is nonunique as there is a direction in which you can travel that would not change the value of *f*.

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	2
0	3	-1	1	=	6
0	0	1	0	=	f-28

Question: What happens if you tried to make X_2 a basic variable next?

Unbounded Solution Example

- Remove the second constraint to create an unbounded design space.

Minimize: $f = -4X_1 - X_2 + 50$

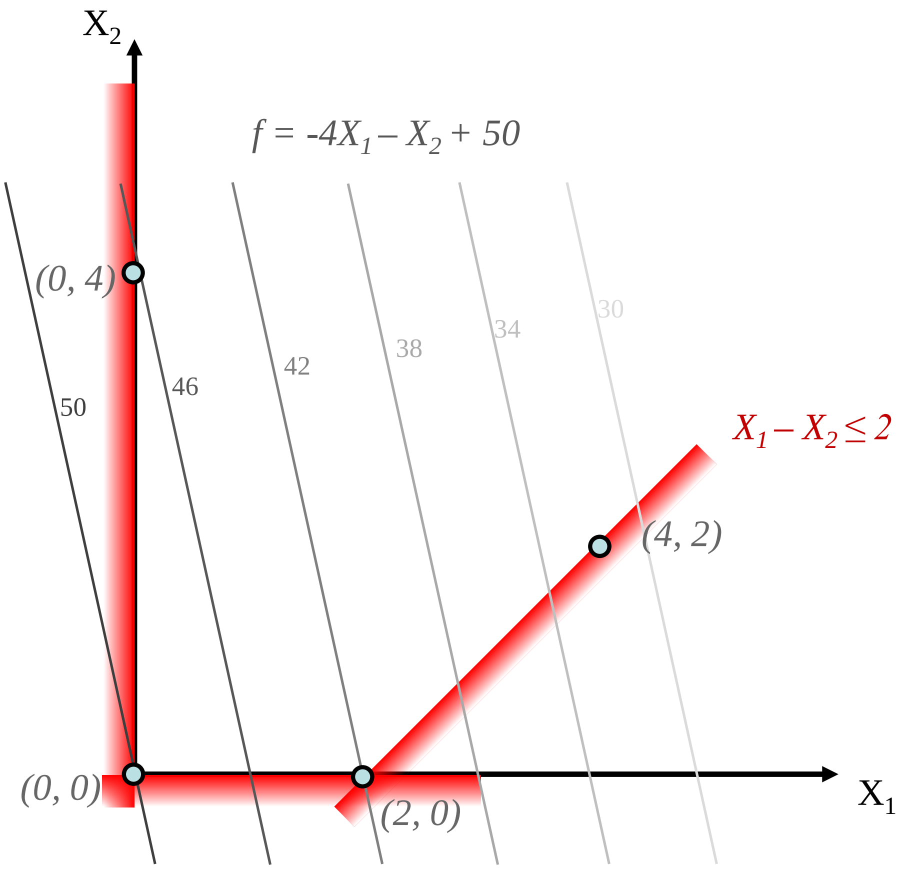
Subject to:

$X_1 - X_2 \leq 2$

$\cancel{X_1 + 2X_2 \leq 8}$

$X_1 \geq 0$

$X_2 \geq 0$



Georgia Institute of Technology
Daniel Guggenheim School of Aerospace Engineering

Unbounded Solution Example

Minimize: $f = -4X_1 - X_2 + 50$

Subject to: $X_1 - X_2 \leq 2$
$X_1 \geq 0$
$X_2 \geq 0$

X_1	X_2	X_3		b
1	-1	1	$=$	2
$\boxed{-4}$	-1	0	$=$	$f-50$

X_1	X_2	X_3		b
1	-1	1	$=$	2
0	-5	4	$=$	$f-42$

- Initial tableau is in canonical form so Phase I is complete.
- Row operations:
  - [Row II] = [Row II] + 4×[Row I]

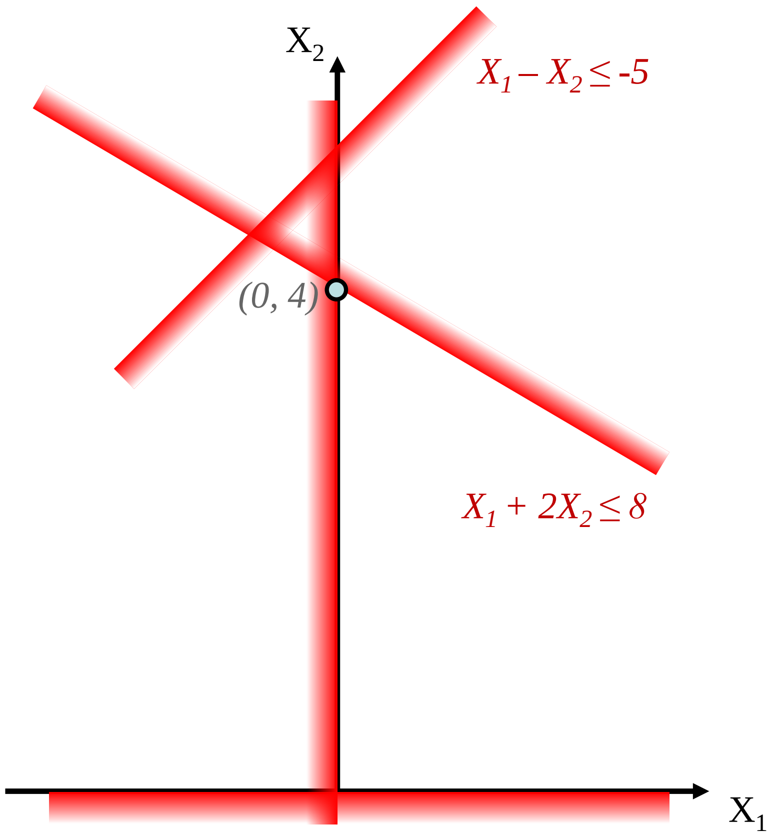
Unbounded Solution Example

- Tableau is in canonical form but there is still a negative cost coefficient.
- It will be impossible to make X2 a basic variable as all of the constraint coefficients in that column are negative or zero.
- This means the solution is unbounded as there is a direction in which you can travel that will continuously decrease *f*.

X_1	X_2	X_3		b
1	-1	1	=	2
0	-5	4	=	f-42

Phase I & No Feasible Solutions Example

- What if the initial tableau cannot be made into canonical form?
- We changed the first constraint to be ≤ -5.



Minimize: $f = -4X_1 - X_2 + 50$

Subject to:
$X_1 - X_2 \leq -5$
$X_1 + 2X_2 \leq 8$
$X_1 \geq 0$
$X_2 \geq 0$

Phase I & No Feasible Solutions Example

Minimize: $f = -4X_1 - X_2 + 50$

Subject to:

$X_1 - X_2 \leq -5$
$X_1 + 2X_2 \leq 8$
$X_1 \geq 0$
$X_2 \geq 0$

X_1	X_2	X_3	X_4		b
1	-1	1	0	=	-5
1	2	0	1	=	8
-4	-1	0	0	=	f-50

X_1	X_2	X_3	X_4		b
-1	1	-1	0	=	5
1	2	0	1	=	8
-4	-1	0	0	=	f-50

- Initial tableau has a -5 in the *b* column but multiplying [Row I] by -1 removes X_3 as a basic variable. No easy way to achieve canonical form.

Phase I & No Feasible Solutions Example

- The more complicated version of Phase I: Add an artificial variable X_5 and the function $w = X_5$ to create the possibility for a second basic variable.
- Row operation:
  - [Row IV] = [Row IV] – [Row I]

X_1	X_2	X_3	X_4	X_5		b
-1	1	-1	0	1	=	5
1	2	0	1	0	=	8
-4	-1	0	0	0	=	f-50
0	0	0	0	1	=	w

X_1	X_2	X_3	X_4	X_5		b
-1	1	-1	0	1	=	5
1	2	0	1	0	=	8
-4	-1	0	0	0	=	f-50
1	-1	1	0	0	=	w-5

Phase I & No Feasible Solutions Example

- Now the goal is to zero out the artificial variable X_5 and function w. Use the same techniques as before except with w as the objective function.
- Row operations:
  - [Row II] = [Row II]/2
  - [Row I] = [Row I] – [Row II]
  - [Row III] = [Row III] + [Row II]
  - [Row IV] = [Row IV] + [Row II]

X_1	X_2	X_3	X_4	X_5		b
-1	1	-1	0	1	=	5
1	2	0	1	0	=	8
-4	-1	0	0	0	=	f-50
1	$\boxed{-1}$	1	0	0	=	w-5

X_1	X_2	X_3	X_4	X_5		b
-3/2	0	-1	-1/2	1	=	1
1/2	1	0	1/2	0	=	4
-7/2	0	0	1/2	0	=	f-46
3/2	0	1	1/2	0	=	w-1

Phase I & No Feasible Solutions Example

- The cost coefficients of *w* (excluding basic variables) are now all positive which means w cannot be reduced any further.
- X_5 is still a basic variable and $w = X_5 = 1$. Because the artificial X_5 and *w* could not be zeroed out, there is no feasible solution to the original problem.

X_1	X_2	X_3	X_4	X_5		b
-3/2	0	-1	-1/2	1	=	1
1/2	1	0	1/2	0	=	4
-7/2	0	0	1/2	0	=	f-46
3/2	0	1	1/2	0	=	w-1

Phase I & No Feasible Solutions Example

- Had it been possible to zero out X_5 and w, another variable would have replaced X_5 as a basic variable.
- Then, simply remove the X_5 column and w row, resulting in a tableau of canonical form for the original problem and completing Phase I.
- Note that having to use artificial variables does not imply there are no feasible solutions.