IM2010: Operations Research
More about the Simplex Method
(Chapter 4)

Ling-Chieh Kung

Department of Information Management
National Taiwan University

March 28, 2013

Road map

- **Interpretations of simplex tableau**.
- Unboundedness and multiple optimal solutions.
- Degeneracy vs. efficiency.

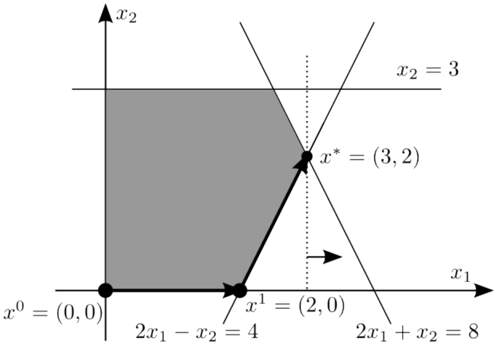
Initialization

- Let's revisit this example:

$$
(P) \quad \begin{array}{rrcrcrl}
\max & x_1 & & & & & \\
\text{s.t.} & 2x_1 & - & x_2 & \leq & 4 & \text{(Constraint 1)} \\
& 2x_1 & + & x_2 & \leq & 8 & \text{(Constraint 2)} \\
& & & x_2 & \leq & 3 & \text{(Constraint 3)} \\
& & & x_i & \geq & 0 & \forall\ i = 1, 2.
\end{array}
$$

Initialization

- Looking at the graphical solution for (P), we may see that its optimal solution is $x^* = (3, 2)$. The dotted line is the isoprofit line. The short arrow indicates the direction we push the isoprofit line.



Initialization

- The standard form of problem (P) is

$$
(S) \quad \begin{array}{rrcrcrcrcrcl}
\max & x_1 & & & & & & & & & & \\
\text{s.t.} & 2x_1 & - & x_2 & + & x_3 & & & & & = & 4 \\
 & 2x_1 & + & x_2 & & & + & x_4 & & & = & 8 \\
 & & & x_2 & & & & & + & x_5 & = & 3 \\
 & \multicolumn{11}{l}{x_i \geq 0 \quad \forall\, i = 1, ..., 5.}
\end{array}
$$

The first iteration

- For problem (S), we form the initial tableau

$$
\begin{array}{ccccc|c}
-1 & 0 & 0 & 0 & 0 & 0 \\
\hline
2 & -1 & 1 & 0 & 0 & x_3 = 4 \\
2 & 1 & 0 & 1 & 0 & x_4 = 8 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
$$

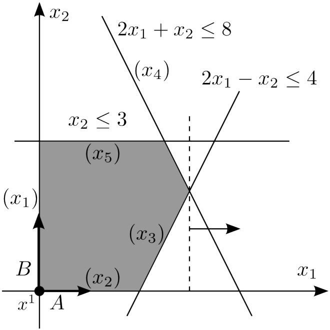
  - The initial basic feasible solution (bfs) is $x^0 = (0, 0, 4, 8, 3)$.
  - The current objective value $z_0 = 0$.
  - Basic variables are x_3, x_4, and x_5.
  - Nonbasic variables are x_1 and x_2.
- In the graph of (P), we may see that x^0 is the origin.

The objective row: Reduced costs

- The 0th row $[-1\ 0\ 0\ 0\ 0]$ have 0s for basic variables.
- For nonbasic ones, the 0th row contains their **reduced costs**.
- We will denote the reduced cost for variable x_j as $\bar{c}_j$ for $x_j \in N$.
- In this example, we know $\bar{c}_1 = -1 < 0$ and $\bar{c}_2 = 0$, which tells us that entering x_1 **improves** the objective while entering x_2 **does not change** the objective.

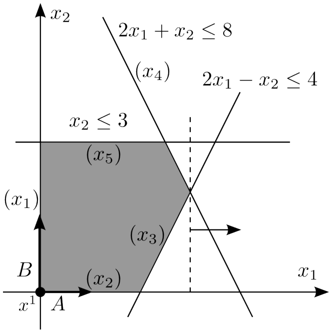
The objective row: Reduced costs

- By entering x_1, we will increase its value from 0 (while keeping $x_2 = 0$) to a positive number.
- This is direction A, an **improving direction**, which corresponds to the fact that $\bar{c}_1 < 0$.



The objective row: Reduced costs

- Suppose we enter x_2, we will increase its value from 0 (while keeping $x_1 = 0$) to a positive number.
- This is direction B, which is not an improving direction. Note that $\bar{c}_2 = 0$.



The objective row: Reduced costs

- What does $\bar{c}_1 = -1$ tell us about the current bfs x^0?
- If we increase x_1 by 1, we will improve our objective by 1!
  - We may recognize this by looking at the objective in (S).
- Similarly, $\bar{c}_2 = 0$ means if we increase x_2 by 1, we will improve our objective by 0, which means no improvement.
  - This may also be verified with the objective in (S).

The entering and RHS columns: Ratio test

- We should enter x_1 to improve our objective.
- With the **entering column** $d = [2\ 2\ 0]^T$ and the RHS $\bar{b} = [4\ 8\ 3]^T$, we apply the ratio test

$$\min\left\{\frac{\bar{b}_i}{d_i} : d_i > 0\right\} = \min\left\{\frac{4}{2}, \frac{8}{2}\right\} = 2$$

and conclude that x_3 should leave.

The entering and RHS columns: Ratio test

- The next tableau is found by pivoting at 2:

$$
\begin{array}{ccccc|c}
-1 & 0 & 0 & 0 & 0 & 0 \\
\hline
\boxed{2} & -1 & 1 & 0 & 0 & x_3 = 4 \\
2 & 1 & 0 & 1 & 0 & x_4 = 8 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
\quad \rightarrow \quad
\begin{array}{ccccc|c}
0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & 2 \\
\hline
1 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & x_1 = 2 \\
0 & 2 & -1 & 1 & 0 & x_4 = 4 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
$$

- The current bfs becomes $x^1 = (2, 0, 0, 4, 3)$ and the current objective value becomes $z_1 = 2$.

The entering and RHS columns: Ratio test

- Consider the ratio test which finds the leaving variable.
- By leaving the basis, the basic variable (in this case, x_3) becomes nonbasic with its value becoming 0.
- Since x_3 is a **slack** variable for constraint 1, it measures the **difference** between the RHS and the left-hand side (LHS) of constraint 1: $x_3 = 4 - (2x_1 - x_2)$.
- When we are at x^0, we have $x_3 = 4$. When we move along direction A, we stop at x^1 with $x_3 = 0$ because constraint 1 prevents us from moving farther.
- Since constraint 1 is nonbinding at x^0 and binding at x^1, we may also say that we move along the improving direction until one constraint changes **from nonbinding to binding**.

The entering and RHS columns: Ratio test

- Along direction A we may "hit" constraint 1 and constraint 2 after moving for some distances.
- We will never hit constraint 3 along direction A.
- Since we must satisfy all the constraints, we want to find the one that we will **hit first**.
- Consider $d_1 = 2$ and $\bar{b}_1 = 4$, the first element of the entering column and RHS, respectively.
- Intuitively and informally, we say that
  - The "**distance**" between the current bfs x^0 and constraint 1 is 4.
  - The "**speed**" we move along direction A is 2.
- Therefore,
  - The ratio $\frac{4}{2} = 2$ is the "**time**" we need to hit constraint 1.

The entering and RHS columns: Ratio test

- To understand this, we may look at the original constraint 1 in (P), $2x_1 - x_2 \leq 4$.
- At x^0, the two variables x_1 and x_2 are 0 and thus the LHS of constraint 1 a value of 0.
- We can say the distance between the constraint and the current bfs is 4.
- When we increase x_1 by 1, we increase the LHS by 2, and thus we say that the speed of approaching the constraint is 2.
- The ratio measures the time we need to hit constraint 1.

The entering and RHS columns: Ratio test

- $d_2 = 2$ and $\bar{b}_2 = 8$ means that the distance between x^0 and constraint 2 is 8 and the speed of approaching constraint 2 is 2.
- The ratio, 4, is the time we need to touch constraint 2.
- Starting at point $x^0 = (0, 0)$ and moving to the right, as ratio test finds $2 < 4$, we will hit constraint 1 before constraint 2.
  - "distance"?
  - $x^0 = (0, 0)$ and along direction A we touch constraint 1 at $x^1 = (2, 0)$, so it seems that the distance should be 2 rather than 4.
  - 4 is actually the **algebraic distance** between x^0 and constraint 1 (the difference between the RHS and the LHS of constraint 1).
  - 2 is the **geometric distance**.
  - We will still use "speed", "distance", and "time" for the entering column, the RHS column, and the ratio because they have an intuitive physical meaning.

The entering and RHS columns: Ratio test

- We summarize our result as below. This is a general result for **any** linear programs.

Proposition 1

When we decide to enter a nonbasic variable x_j, let d be the entering column and $\bar{b}$ be the RHS column. If for row i we have $d_i > 0$, then along the direction we are going to move:

- *$\bar{b}_i$ is the distance between the bfs and the constraint for row i,*
- *d_i is the speed approaching the constraint, and*
- *the ratio $\bar{b}_i / d_i$ is the time we need to hit the constraint.*

Sign of an element in the entering column

- How about constraint 3?
- Recall that we ignored constraint 3 when doing the ratio test because $d_3 = 0$.
- If we say $\bar{b}_3 = 3$ is the distance between constraint 1 and x^0 and $d_3 = 0$ is the speed, then the time we need to touch constraint 3 is infinity!
- This is true, according to the graph. Since constraint 3 is parallel to direction A, no matter how long we move along direction A, we will never touch constraint 3.

Sign of an element in the entering column

- Now we have investigated the meaning of a positive or zero element in the entering column. How about a **negative** one?
- Moving along direction B means entering x_2, and in this case we have $d = [-1\ 1\ 1]^T$.
- We observe that $d_1 < 0$, which means constraint 1 is "**behind**" x^0 if moving along direction B!
- We may ignore row 1 when doing the ratio test because along direction B we will **never** hit constraint 1.
- On the other hand, constraint 2 and 3 are both "**in front of**" x^0 along direction B because d_2 and d_3 are both positive.

Sign of an element in the entering column

- **Proposition 2**

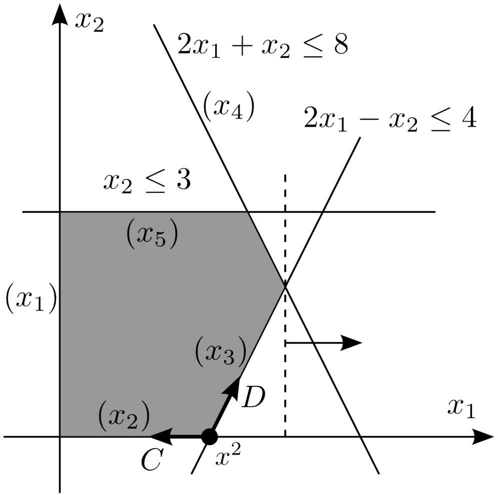
  When we decide to enter a nonbasic variable x_j, let d be the entering column. Then along the direction we are going to move, one of the following holds for each constraint of row i:

  - *If $d_i > 0$, then the constraint is in front of the current bfs. We will touch it after increasing x_j by $\bar{b}_i/d_i$.*
  - *If $d_i = 0$, then constraint i is parallel to the current bfs. We will never touch it.*
  - *If $d_i < 0$, then constraint i is behind the current bfs. We will never touch it.*

The second iteration

- At x^1, we again look at the reduced cost of nonbasic variables x_2 and x_3 to decide an entering variable.
- Now $\bar{c}_2 = -\frac{1}{2} < 0$ and $\bar{c}_3 = \frac{1}{2} > 0$ tell us that entering x_2 improves our objective but entering x_3 does not.
- Therefore, we choose x_2 to be the entering variable.
- If we only want to solve the problem, then we just need to do a ratio test and find the leaving variable.
- However, here we are interested in the direction we are going to move along.

The direction to move along



The direction to move along

- When we were at bfs x^0, we increase x_1 by moving on the x_1-axis or increase x_2 on the x_2-axis.
- At bfs x^1, as we want to increase the value of x_2, it seems that we should move parallel to the x_2-axis, which is along vector $(0, 1)$.
- This is not true in the simplex method, because it moves only **along edges** of the feasible region!
- So we may expect to move along direction D. This is correct, but why?

The direction to move along

- Using the simplex method, we switch from one bfs to one of its **adjacent bfs**.
  - Two bfs are adjacent if they share $n-1$ binding constraints.
- To move to a neighboring bfs, we must move along **one of the binding constraints**, so at x^1, we must move along either $2x_1 - x_2 = 4$ or $x_2 = 0$, that is, direction C or D.
  - Entering x_2: The constraint $x_2 = 0$ is no longer binding. We move along the other binding constraint $2x_1 - x_2 = 4$ (direction D).
  - Entering x_3: The constraint $2x_1 - x_2 \leq 4$ is no longer binding. We move along the other binding constraint $x_2 = 0$ (direction C).

The objective row: Reduced costs

- The second iteration is

$$
\begin{array}{ccccc|c}
0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & 2 \\
\hline
1 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & x_1 = 2 \\
0 & \boxed{2} & -1 & 1 & 0 & x_4 = 4 \\
0 & 1 & 0 & 0 & 1 & x_5 = 3
\end{array}
\quad \rightarrow \quad
\begin{array}{ccccc|c}
0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 3 \\
\hline
1 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & x_1 = 3 \\
0 & 1 & \frac{-1}{2} & \frac{1}{2} & 0 & x_2 = 2 \\
0 & 0 & \frac{1}{2} & \frac{-1}{2} & 1 & x_5 = 1
\end{array}
$$

  and we get the third bfs $x^* = (3, 2, 0, 0, 1)$, which is optimal, and the optimal objective value $z^* = 3$.
- In the second tableau (the left one above), we have $\bar{c}_2 = -\frac{1}{2} < 0$ and $\bar{c}_3 = \frac{1}{2} > 0$. Do they really indicate the unit improvements we have by entering x_2 and x_3?

The objective row: Reduced costs

- To increase the value of x_2, we know that we must move along direction D, which is along the equation $2x_1 - x_2 = 4$.
  - Increasing x_2 by 1 requires us to increase x_1 by $\frac{1}{2}$ **at the same time** so that the constraint is still binding.
  - Therefore, increasing x_2 by 1 improves the objective by $\frac{1}{2}$.
  - This is an indirect effect: increasing x_2 makes us increase x_1, and increasing x_1 makes the objective increase.
- Now consider entering x_3 and moving along direction C, the equation $x_2 = 0$. The effect is again indirect:
  - If we want to increase x_3 by 1 while keeping $x_2 = 0$, we must have x_1 to decrease by $\frac{1}{2}$ so that the constraint $2x_1 - x_2 + x_3 = 4$ is still satisfied.
  - That's why the objective decreases by $\frac{1}{2}$.

Sign of an element in the entering column

- At bfs x^1 we have $d = \left[\frac{-1}{2}\ 2\ 1\right]^T$ if we enter x_2.
- We want to show that Proposition 2 is correct in this example.
- The first row is now representing the constraint $x_1 \geq 0$.
- Recall that two neighboring bfs have exactly one different binding constraint. For example, $x_2 \geq 0$ is binding at both x^0 and x^1, but $x_1 \geq 0$ is binding only at x^0 and $2x_1 - x_2 \leq 4$ is only binding at x^1.
- Since the rows of a simplex tableau are for the **nonbinding** constraints, two simplex tableau associating to two adjacent bfs will have **one row** representing **different constraints**.
- In iteration 1, x_3 leaves in row 1, so row 1 becomes the representation of the nonbinding constraint $x_1 \geq 0$ of x^1.

Sign of an element in the entering column

- Now we can interpret the entering column by Proposition 2.
- Along direction D:
  - $d_1 < 0$ and constraint 4 ($x_1 \geq 0$) is behind the current bfs,
  - $d_2 > 0$ and constraint 2 is in front of the current bfs, and
  - $d_3 > 0$ and constraint 3 is in front of the current bfs.
- We may do the same interpretation for direction C. If we enter x_3, then $d = \begin{bmatrix} \frac{1}{2} & -1 & 0 \end{bmatrix}^T$. Along direction E:
  - $d_1 > 0$ and constraint 4 ($x_1 \geq 0$) is in front of the current bfs,
  - $d_2 < 0$ and constraint 2 is behind the current bfs, and
  - $d_3 = 0$ and constraint 3 is parallel to the current bfs.

The entering and RHS columns: Ratio test

- Here we only check the case of entering x_2 with $d = \left[\frac{-1}{2}\ 2\ 1\right]^T$ and $\bar{b} = [2\ 4\ 3]^T$.
- For constraint 2, the distance is 4 and the speed is 2.
- This may be verified by looking at constraint 2 in (P):

$$2x_1 + x_2 \leq 8.$$

  - At $x^1 = (2, 0)$, the LHS is 4 and the RHS is 8, so the distance is 4.
  - Along direction C (the equation $2x_1 - x_2 = 4$), if we increase x_2 by 1, then we must increase x_1 by $\frac{1}{2}$, and they together increase the LHS of $2x_1 + x_2 \leq 8$ by $2(\frac{1}{2}) + 1 = 2$.
  - Therefore, the speed approaching constraint 2 is 2.

The entering and RHS columns: Ratio test

- For constraint 3, the distance is 3 and the speed is 1.
- This may be verified by looking at constraint 3 in (P):

$$x_2 \leq 3.$$

  - At $x^1 = (2, 0)$, the LHS is 0 and the RHS is 3, so the distance is 3.
  - Along direction C (the equation $2x_1 - x_2 = 4$), if we increase x_2 by 1, then we must increase x_1 by $\frac{1}{2}$, and they together increase the LHS of $x_2 \leq 3$ by 1 (x_1 actually has no effect here).
  - Therefore, the speed approaching constraint 2 is 1.
- The ratios $\frac{4}{2} = 2$ and $\frac{3}{1} = 3$ tells us that we will touch constraint 2 first.

Conclusion

- There is an interpretation of the reduced costs in the objective row, the entering column, and the RHS column.
- Their physical meanings are given, though not very rigorously.
- Understanding the concepts listed in this note is not very easy, but it should help you understand the elegant idea of the simplex method more.
- It will also help you solve problems like Problem 4.Review.17 and 4.Review.18 in the textbook.
- Even if you can not understand every detail in this note, it will still be good to understand the conclusion and intuition in the two propositions.

Road map

- Interpretations of simplex tableau.
- **Unboundedness and multiple optimal solutions**.
- Degeneracy vs. efficiency.

Unbounded linear programs

- So far all the linear programs we encountered have exactly one unique optimal solution.
- What if a linear program is **unbounded**? Can the simplex method detect the unboundedness? If so, how?
- Consider the following example:

$$\begin{array}{rrcrcr} \max & x_1 & & & & \\ \text{s.t.} & x_1 & - & x_2 & \leq & 1 \\ & 2x_1 & - & x_2 & \leq & 4 \\ & & & \multicolumn{3}{l}{x_i \geq 0 \quad \forall i = 1, 2.} \end{array}$$

Unbounded linear programs

- The standard form is:

$$\begin{array}{rrcrcrcrcr} \max & x_1 & & & & & & & & \\ \text{s.t.} & x_1 & - & x_2 & + & x_3 & & & = & 1 \\ & 2x_1 & - & x_2 & & & + & x_4 & = & 4 \\ & \multicolumn{9}{l}{x_i \geq 0 \quad \forall i = 1, ..., 4.} \end{array}$$

- The first iteration:

$$\begin{array}{cccc|c} -1 & 0 & 0 & 0 & 0 \\ \hline \boxed{1} & -1 & 1 & 0 & x_3 = 1 \\ 2 & -1 & 0 & 1 & x_4 = 4 \end{array} \quad \rightarrow \quad \begin{array}{cccc|c} 0 & -1 & 1 & 0 & 1 \\ \hline 1 & -1 & 1 & 0 & x_1 = 1 \\ 0 & 1 & -2 & 1 & x_4 = 2 \end{array}$$

Unbounded linear programs

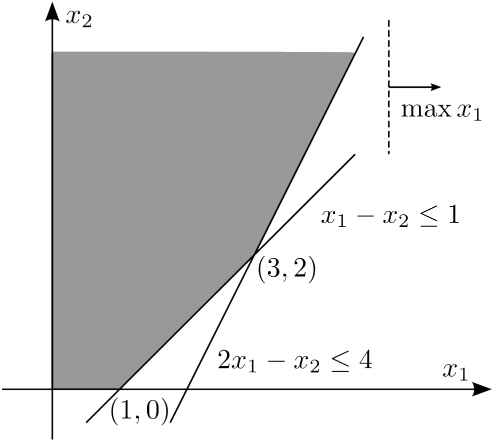
- The second iteration:

$$\begin{array}{cccc|c} 0 & -1 & 1 & 0 & 1 \\ \hline 1 & -1 & 1 & 0 & x_1 = 1 \\ 0 & \boxed{1} & -2 & 1 & x_4 = 2 \end{array} \quad \rightarrow \quad \begin{array}{cccc|c} 0 & 0 & -1 & 1 & 3 \\ \hline 1 & 0 & -1 & 1 & x_1 = 3 \\ 0 & 1 & -2 & 1 & x_2 = 2 \end{array}$$

- Wait... how may we do the third iteration? The ratio test fails!
  - All the denominators are nonpositive! Which variable to leave?
- No variable should leave: Along the improving direction (by entering x_3), both the two nonbinding constraints are **behind** us.
- The improving direction is thus an **unbounded improving direction**.

Unbounded improving directions

- At $(3, 2)$, when we enter x_3, we move along the rightmost edge. Both nonbinding constraints $x_1 \geq 0$ and $x_2 \geq 0$ are behind us.



Detecting unbounded linear programs

- For a **maximization** problem, whenever we see **any** column in **any** tableau

$\bar{c}_j$	
A_{1j}	
$\vdots$	
A_{mj}	

  such that $c_j < 0$ and $A_{ij} \leq 0$ for all $i = 1, ..., m$:
  - $\bar{c}_j < 0$: This is an improving direction.
  - $A_{ij} \leq 0$ for all $i = 1, ..., m$: This is an unbounded direction.
- In this case, we may stop and conclude that this linear program is unbounded.
- What is the unbounded condition for a **minimization** problem?

Multiple optimal solutions

- Consider another example (in standard form directly):

$$
\begin{array}{rrcrcrcrcrcl}
\max & x_1 & + & x_2 & & & & & & & & \\
\text{s.t.} & x_1 & + & 2x_2 & + & x_3 & & & & & = & 12 \\
& 2x_1 & + & x_2 & & & + & x_4 & & & = & 12 \\
& x_1 & + & x_2 & & & & & + & x_5 & = & 7 \\
& \multicolumn{11}{l}{x_i \geq 0 \quad \forall i = 1, ..., 5.}
\end{array}
$$

Multiple optimal solutions

- In two iterations, we find an optimal solution. What is it?

-1	-1	0	0	0	0
1	2	1	0	0	$x_3 = 12$
$\boxed{2}$	1	0	1	0	$x_4 = 12$
1	1	0	0	1	$x_5 = 7$

$\rightarrow$

0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	6
0	$\frac{3}{2}$	1	$-\frac{1}{2}$	0	$x_3 = 6$
1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$x_1 = 6$
0	$\boxed{\frac{1}{2}}$	0	$-\frac{1}{2}$	1	$x_5 = 1$

$\rightarrow$

0	0	0	0	1	7
0	0	1	1	-2	$x_3 = 3$
1	0	0	1	-2	$x_1 = 5$
0	1	0	-1	2	$x_2 = 2$

Multiple optimal solutions

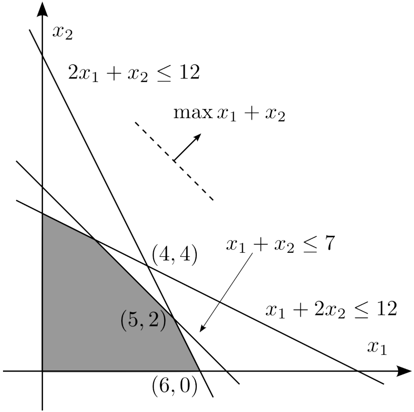
- In practice, we will simply stop and report the optimal solution.
- Here to illustrate the power of the simplex method, let's focus on the optimal tableau:

0	0	0	0	1	7
0	0	1	1	-2	$x_3 = 3$
1	0	0	1	-2	$x_1 = 5$
0	1	0	-1	2	$x_2 = 2$

- What does a zero reduced cost ($\bar{c}_4 = 0$) mean?
  - If we increase this variable by 1, the objective value will be decreased by **zero**.
- As the current solution is optimal, if there is a direction such that moving along it does not change the objective value, **all points** on that direction are optimal.

Multiple optimal solutions

- At **an** optimal solution $(5, 2)$, by entering x_4, we move along $x_1 + x_2 = 7$ and all points on this direction are optimal.



Detecting multiple optimal solutions

- At the **optimal** (not any!) tableau, if
  - x_j's reduced cost $\bar{c}_j = 0$ **and**
  - along the direction of entering x_j, we may move a **positive distance**,

  then the linear program has multiple optimal solution.
- What does the second condition mean?
- Is "there is a constraint parallel to the isoprofit line" necessary, sufficient, both, or none?

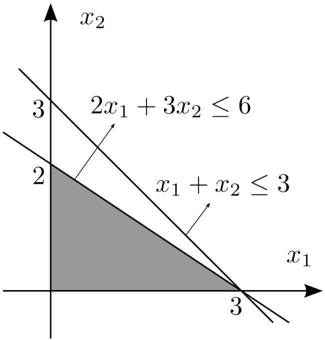
Road map

- Interpretations of simplex tableau.
- Unboundedness and multiple optimal solutions.
- **Degeneracy vs. efficiency**.

Solving degenerate linear programs

- Recall that an LP is **degenerate** if multiple bases correspond to a single basic solution.
- For the simplex method, in each iteration we move to an adjacent basis.
- If the LP is degenerate, it is possible to move to **another** basis but still at the **same** basic feasible solution.
- Running an iteration may have **no improvement**!

$$
\begin{array}{rrcrcl}
\max & x_1 & + & 3x_2 & & \\
\text{s.t.} & x_1 & + & x_2 & \leq & 3 \\
 & 2x_1 & + & 3x_2 & \leq & 6 \\
 & & & x_i \geq 0 & & \forall i = 1, 2.
\end{array}
$$



Solving degenerate linear programs

- In three iterations, we may find an optimal solution:

$$
\begin{array}{cccc|c}
-1 & -3 & 0 & 0 & 0 \\
\hline
\boxed{1} & 1 & 1 & 0 & x_3 = 3 \\
2 & 3 & 0 & 1 & x_4 = 6
\end{array}
\rightarrow
\begin{array}{cccc|c}
0 & -2 & 1 & 0 & 3 \\
\hline
1 & 1 & 1 & 0 & x_1 = 3 \\
0 & \boxed{1} & -2 & 1 & x_4 = 0
\end{array}
$$

$$
\rightarrow
\begin{array}{cccc|c}
0 & 0 & -3 & 2 & 3 \\
\hline
1 & 0 & \boxed{3} & -1 & x_1 = 3 \\
0 & 1 & -2 & 1 & x_2 = 0
\end{array}
\rightarrow
\begin{array}{cccc|c}
1 & 0 & 0 & 1 & 6 \\
\hline
\frac{1}{3} & 0 & 1 & -\frac{1}{3} & x_3 = 1 \\
\frac{2}{3} & 1 & 0 & \frac{1}{3} & x_2 = 2
\end{array}
$$

- Note that in the second iteration, there is no improvement!
- The basis changes but the basic feasible solution does not change.

Computational efficiency of the simplex method

- In general, when we use the simplex method to solve a degenerate LP, there may be some iterations that have no improvements.
  - We think we can have improvements (with a positive reduced cost for a minimization problem), but we hit a constraint before we move for any positive distance.
- For some (very strange) instances, the simplex method needs to travel through all the bases before it can make a conclusion.
- Therefore, the simplex method is, in the worst case, an **exponential-time** algorithm:

$$O\left(\binom{n}{m} f(n,m)\right),$$

where $f(n,m)$ is the time of completing one iteration.

Polynomial-time algorithms for LP

- There are polynomial-time algorithms for Linear Programming.
  - Beyond the scope of this course.
- Interestingly, some of them are very complicated and run **slower** than the simplex method for most **practical** problems.
- With its simplicity and extendability, The simplex method is still the most widely adopted method for Linear Programming in practice.
- However, there is a big problem ...

Cycling

- At a basic feasible solution, the simplex method may enter an infinite loop! This is called **cycling**.
  - Basis 1 $\rightarrow$ basis 2 $\rightarrow$ basis 3 $\rightarrow \cdots \rightarrow$ basis 1.
- This may happen when we use a "not so good" way of selecting entering and leaving variables.
- There are at least two ways to avoid cycling:
  - Randomize the selection of variables.
  - Apply the **smallest index rule**.
- By using the smallest index rule:
  - When there are multiple variables having positive reduced cost for a minimization problem, select the one with the smallest index.
  - When there are multiple variables whose ratio are all the smallest ratio, select the one with the smallest index.
  - Smallest indexing: choose x_i rather than x_j if $i < j$.

The smallest index rule

- The smallest index rule may not generate the **least iterations** toward an optimal solution.
  - Why don't we choose the variable with the reduced cost with the largest magnitude?
  - No variable selection rule can guarantee to be the most efficient!
- The smallest index rule can guarantee **no cycling**!
  - The "most significant reduced cost" rule, however, may result in cycling in some cases.