CSL 356: Analysis and Design of Algorithms

Ragesh Jaiswal

CSE, IIT Delhi

Linear Programming

Solving an LP

Linear Programming: Solving LP

- To be able to design an algorithm for solving LP problems, it will be useful if we define problems more precisely in some standard format.
- **Standard form**: A Linear Program is said to be *standard form* if the following holds:
  1. The linear objective function should be *maximized*.
  2. All variables have non-negativity constraint.
     i.e., for all i, $x_i \geq 0$.
  3. All the remaining linear constraints are of the following form:
     $\sum_{j=1}^{n} a_j \cdot x_j \leq b_j$

Linear Programming: Solving LP

- **Standard form**: A Linear Program is said to be *standard form* if the following holds:
  1. The linear objective function should be *maximized*.
  2. All variables have non-negativity constraint.
     i.e., for all i, $x_i \geq 0$.
  3. All the remaining linear constraints are of the following form:
     $\sum_{j=1}^{n} a_j \cdot x_j \leq b_j$

- Question: Is there a way to convert any LP problem to an *equivalent* standard form?
  - Equivalence of LP's: Two LP problems P1 and P2 are said to be equivalent if for any feasible solution for P1 with objective value z, there is a feasible solution of P2 with the same objective value and vice versa.

Linear Programming: Solving LP

- **Standard form**: A Linear Program is said to be *standard form* if the following holds:
  1. The linear objective function should be *maximized*.
  2. All variables have non-negativity constraint.
     i.e., for all i, $x_i \geq 0$.
  3. All the remaining linear constraints are of the following form:
     $\sum_{j=1}^{n} a_j \cdot x_j \leq b_j$

- A general LP problem might not be in standard for because it might have:
  1. Equality constraints (=) rather than inequality (≤).
  2. ≥ instead of ≤.
  3. Variables without non-negativity constraints.
  4. Minimization rather than maximization.

Linear Programming: Solving LP

- A general LP problem might not be in standard form because it might have:
  1. Equality constraints ($=$) rather than inequality ($\leq$).
     - Idea: $a = b$ can be expresses as $a \leq b$ and $a \geq b$.
  2. $\geq$ instead of $\leq$.
  3. Variables without non-negativity constraints.
  4. Minimization rather than maximization.

Linear Programming: Solving LP

- A general LP problem might not be in standard form because it might have:
  1. Equality constraints ($=$) rather than inequality ($\leq$).
     - Idea: $a = b$ can be written as $a \leq b$ and $a \geq b$.
  2. $\geq$ instead of $\leq$.
     - Idea: $a \geq b$ can be written as $-a \leq -b$.
  3. Variables without non-negativity constraints.
  4. Minimization rather than maximization.

Linear Programming: Solving LP

- A general LP problem might not be in standard form because it might have:
  1. Equality constraints ($=$) rather than inequality ($\leq$).
     - Idea: $a = b$ can be written as $a \leq b$ and $a \geq b$.
  2. $\geq$ instead of $\leq$.
     - Idea: $a \geq b$ can be written as $-a \leq -b$.
  3. Variables without non-negativity constraints.
     - Idea: Replace a variable x (that has no non-negativity constraint) with $(x' - x'')$ everywhere and put $x' \geq 0$ and $x'' \geq 0$.
  4. Minimization rather than maximization.

Linear Programming: Solving LP

- A general LP problem might not be in standard form because it might have:
  1. Equality constraints (=) rather than inequality (≤).
     - Idea: $a = b$ can be written as $a \leq b$ and $a \geq b$.
  2. ≥ instead of ≤.
     - Idea: $a \geq b$ can be written as $-a \leq -b$.
  3. Variables without non-negativity constraints.
     - Idea: Replace a variable x (that has no non-negativity constraint) with $(x' - x'')$ everywhere and put $x' \geq 0$ and $x'' \geq 0$.
  4. Minimization rather than maximization.
     - Idea: Replace "Minimize $\sum c_i \cdot x_i$" with "Maximize $\sum(-c_i) \cdot x_i$".

Linear Programming: Solving LP

- Example:
  - Minimize $-2x_1 + 3x_2$
  - subject to
    - $x_1 + x_2 = 7$
    - $x_1 - 2x_2 \leq 4$
    - $x_1 \geq 0$

Linear Programming: Solving LP

- Example: Minimize to Maximize
  - Maximize $2x_1 - 3x_2$
  - subject to
    - $x_1 + x_2 = 7$
    - $x_1 - 2x_2 \leq 4$
    - $x_1 \geq 0$

Linear Programming: Solving LP

- Example: non-negativity constraint for x_2
  - Maximize $2x_1 - 3(x_2' - x_2'')$
  - subject to
    - $x_1 + (x_2' - x_2'') = 7$
    - $x_1 - 2(x_2' - x_2'') \leq 4$
    - $x_1 \geq 0, x_2' \geq 0, x_2'' \geq 0$

Linear Programming: Solving LP

- Example: non-negativity constraint for x_2
  - Maximize $2x_1 - 3x_2' + 3x_2''$
  - subject to
    - $x_1 + x_2' - x_2'' = 7$
    - $x_1 - 2x_2' + 2x_2'' \leq 4$
    - $x_1 \geq 0, x_2' \geq 0, x_2'' \geq 0$

Linear Programming: Solving LP

- Example: renaming variables
  - Maximize $2x_1 - 3x_2 + 3x_3$
  - subject to
    - $x_1 + x_2 - x_3 = 7$
    - $x_1 - 2x_2 + 2x_3 \leq 4$
    - $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

Linear Programming: Solving LP

- Example: Equality to inequality
  - Maximize $2x_1 - 3x_2 + 3x_3$
  - subject to
    - $x_1 + x_2 - x_3 \leq 7$
    - $-x_1 - x_2 + x_3 \leq -7$
    - $x_1 - 2x_2 + 2x_3 \leq 4$
    - $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

Linear Programming: Solving LP

- **<u>Standard form</u>**: A Linear Program is said to be *standard form* if the following holds:
  1. The linear objective function should be *maximized*.
  2. All variables have non-negativity constraint.
     i.e., for all i, $x_i \geq 0$.
  3. All the remaining linear constraints are of the following form:
     $\sum_{j=1}^{n} a_j \cdot x_j \leq b_j$.

- It will be useful to further convert an LP in standard for to an equivalent LP in *Slack form*.
  - <u>Slack form</u>: For every inequality $\sum_j a_j x_j \leq b_j$, we introduce a *slack* variable s_j and replace$\sum_j a_j x_j \leq b_j$ with $s_j = b_j - \sum_j a_j x_j$ and $s_j \geq 0$.

Linear Programming: Solving LP

- Example:
  - Maximize $2x_1 - 3x_2 + 3x_3$
  - subject to
    - $x_1 + x_2 - x_3 \leq 7$
    - $-x_1 - x_2 + x_3 \leq -7$
    - $x_1 - 2x_2 + 2x_3 \leq 4$
    - $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

Linear Programming: Solving LP

- Example: Standard form to slack form.
  - $z \quad = \qquad\qquad 2x_1 - 3x_2 + 3x_3$
  - $x_4 = 7 - x_1 - x_2 + x_3$
  - $x_5 = -7 + x_1 + x_2 - x_3$
  - $x_6 = 4 - x_1 + 2x_2 - 2x_3$
  - $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0.$
- The variables on the LHS are called *basic variables* and those on the RHS are called *non-basic variables*.
- Basic solution: Set all non-basic variables to 0 and compute the value of the basic variables.

Linear Programming: Solving LP

- The variables on the LHS are called *basic variables* and those on the RHS are called *non-basic variables*.
- <u>Basic solution</u>: Set all non-basic variables to 0 and compute the value of the basic variables.
- <u>Simplex algorithm</u>:
  - Repeat:
    - **Pivot**: Rewrite the LP in an equivalent slack form such that the objective value of the basic solution increases.

Linear Programming:

The Simplex Algorithm

Linear Programming: Solving LP

- <u>Simplex algorithm</u>:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- Example:
  - $z = 3x_1 + x_2 + 2x_3$
  - $x_4 = 30 - x_1 - x_2 - 3x_3$
  - $x_5 = 24 - 2x_1 - 2x_2 - 5x_3$
  - $x_6 = 36 - 4x_1 - x_2 - 2x_3$
- Use $x_1 = (9 - x_6/4 - x_2/4 - x_3/2)$

Linear Programming: Solving LP

- Simplex algorithm:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- Example:
  - $z = 3(9 - x_6/4 - x_2/4 - x_3/2) + x_2 + 2x_3$
  - $x_4 = 30 - (9 - x_6/4 - x_2/4 - x_3/2) - x_2 - 3x_3$
  - $x_5 = 24 - 2(9 - x_6/4 - x_2/4 - x_3/2) - 2x_2 - 5x_3$
  - $x_1 = (9 - x_6/4 - x_2/4 - x_3/2)$

Linear Programming: Solving LP

- <u>Simplex algorithm</u>:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- Example:
  - $z = 27 + x_2/4 + x_3/2 - 3x_6/4$
  - $x_4 = 21 - 3x_2/4 - 5x_3/2 + x_6/4$
  - $x_5 = 6 - 3x_2/2 - 4x_3 + x_6/2$
  - $x_1 = 9 - x_2/4 - x_3/2 - x_6/4$
- Now x_2, x_3, and x_6 are the non-basic variables and x_1, x_4, and x_5 are the basic variables.
- The objective value of the *basic solution* is now 27.
- <u>Claim</u>: If the basic solution is feasible for the LP before pivoting, then the basic solution for the LP after pivoting is also feasible.

Linear Programming: Solving LP

- <u>Simplex algorithm</u>:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- Example:
  - $z = 27 + x_2/4 + x_3/2 - 3x_6/4$
  - $x_4 = 21 - 3x_2/4 - 5x_3/2 + x_6/4$
  - $x_5 = 6 - 3x_2/2 - 4x_3 + x_6/2$
  - $x_1 = 9 - x_2/4 - x_3/2 - x_6/4$
- Use $x_3 = 3/2 - 3x_2/8 - x_5/4 + x_6/8$

Linear Programming: Solving LP

- Simplex algorithm:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- Example:
  - $z = 111/4 + x_2/16 - x_5/8 - 11x_6/16$
  - $x_4 = 69/4 + 3x_2/16 + 5x_5/8 - x_6/16$
  - $x_1 = 33/4 - x_2/16 + x_5/8 - 5x_6/16$
  - $x_3 = 3/2 - 3x_2/8 - x_5/4 + x_6/8$
- Now x_2, x_5, and x_6 are the non-basic variables and x_1, x_3, and x_4 are the basic variables.
- The objective value of the *basic solution* is now $111/4$.

Linear Programming: Solving LP

- Simplex algorithm:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- Example:
  - $z = 111/4 + x_2/16 - x_5/8 - 11x_6/16$
  - $x_4 = 69/4 + 3x_2/16 + 5x_5/8 - x_6/16$
  - $x_1 = 33/4 - x_2/16 + x_5/8 - 5x_6/16$
  - $x_3 = 3/2 - 3x_2/8 - x_5/4 + x_6/8$
- Use $x_2 = 4 - 8x_3/3 - 2x_5/3 + x_6/3$

Linear Programming: Solving LP

- <u>Simplex algorithm</u>:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- Example:
  - $z = 28 - x_3/6 - x_5/6 - 2x_6/3$
  - $x_1 = 8 + x_3/6 + x_5/6 - x_6/3$
  - $x_2 = 4 - 8x_3/3 - 2x_5/3 + x_6/3$
  - $x_4 = 18 - x_3/2 + x_5/2$
- Now the basic solution is the optimal solution.
- The optimal objective value for the initial LP is 28 and the value of the variables are $x_1 = 8$, $x_2 = 4$, and $x_3 = 0$.

Linear Programming: Solving LP

- <u>Simplex algorithm</u>:
  - Repeat:
    - **Pivot**: Rewrite the LP in slack form such that the objective value of the basic solution increases.
- We looked at a contrived example devoid of any complications. Here are some of the complications that could arise:
  1. What if the initial basic solution is not a feasible solution?
  2. What if the LP is unbounded? How and where do we detect this?
  3. What if after a pivoting step the objective value of the basic solution does not increase? What is the running time of the Simplex algorithm?

Linear Programming: Solving LP

- Complications:
  1. What if the initial basic solution is not a feasible solution?
     - We will determine this in a preprocessing step. If the LP has a feasible solution, then we will rewrite it in a form where the basic solution is feasible.
  2. What if the LP is unbounded? How and where do we detect this?
     - We will check this while pivoting.
  3. What if after a pivoting step the objective value of the basic solution does not increase? What is the running time of the Simplex algorithm?
     - This is indeed a problem with Simplex. The algorithm might cycle without increasing the objective value. Simplex is actually not a polynomial time algorithm but it is still used in practice because it works very well on real world instances.

Linear Programming: Solving LP

- (*Complication 2*) What if the LP is unbounded? How and where do we detect this?
- Consider the following general slack LP that we obtain while running Simplex:
- $z \quad = \ v + \ c_1x_1 \ + \ c_2x_2 + \ldots \ + \ c_nx_n$
- $x_{n+1} = \ b_1 - a_{11}x_1 - \ a_{12}x_2 - \ldots - \ a_{1nxn}$
- $x_{n+2} = \ b_2 - a_{21}x_1 - \ a_{22}x_2 - \ldots - \ a_{2nxn}$
- .
- $x_{n+m} = \ b_m - a_{m1}x_1 - a_{m2}x_2 - \ldots - a_{mn}x_n$
- <u>Claim</u>: Suppose $c_i > 0$ and $a_{1i}, a_{2i}, a_{3i}, \ldots, a_{mi} \ \leq \ 0$. Then the LP is unbounded.

Linear Programming: Solving LP

- (*Complication 3*) What if after a pivoting step the objective value of the basic solution does not increase? What is the running time of the Simplex algorithm?
- Consider the following example:
- $z \;=\; 8 \;+ x_3 - x_4$
- $x_1 = \;8 - \;x_2 \;- \;x_4$
- $x_5 = \;x_2 - \;x_3$
- We have to pivot using $x_3 = x_2 - x_5$ but that gives us
- $z \;=\; 8 \;+ x_2 - x_4 - x_5$
- $x_1 = \;8 - \;x_2 \;- \;x_4$
- $x_3 = \;x_2 - \;x_5$
- The objective value of the basic solution does not change.

Linear Programming: Solving LP

- (*Complication 3*) What if after a pivoting step the objective value of the basic solution does not increase? What is the running time of the Simplex algorithm?
- So, the Simplex may cycle between slack forms without increasing the objective value of the basic solution.
- Claim: Each slack form is uniquely determined by the set of basic and non-basic variables.
- Question: What is the upper bound on the number of slack forms that the Simplex cycles without increasing the objective value of the basic solution?

Linear Programming: Solving LP

- (*Complication 3*) What if after a pivoting step the objective value of the basic solution does not increase? What is the running time of the Simplex algorithm?
- So, the Simplex may cycle between slack forms without increasing the objective value of the basic solution.
- <u>Claim</u>: Each slack form is uniquely determined by the set of basic and non-basic variables.
- <u>Question:</u> What is the upper bound on the number of slack forms that the Simplex cycles without increasing the objective value of the basic solution?
  - ${}^{n+m}C_m$. This is the upper bound on the number of different slack forms.

Linear Programming: Solving LP

- (*Complication 3*) What if after a pivoting step the objective value of the basic solution does not increase? What is the running time of the Simplex algorithm?
- So, the Simplex may cycle between slack forms without increasing the objective value of the basic solution.
- Claim: Each slack form is uniquely determined by the set of basic and non-basic variables.
- Claim: If the Simplex fails to terminate in ${}^{n+m}C_m$ steps, then it cycles.
- There is a way (*Bland's rule*) to choose the pivoting variables so that Simplex always terminates.

Linear Programming: Solving LP

- (*Complication 1*) What if the initial basic solution is not a feasible solution?
- We construct the following LP, L' in slack form:
- $z = -x_0$
- $x_{n+1} = b_1 - a_{11}x_1 - a_{12}x_2 - \ldots - a_{1nxn} + x_0$
- $x_{n+2} = b_2 - a_{21}x_1 - a_{22}x_2 - \ldots - a_{2n}x_n + x_0$
- .
- $x_{n+m} = b_m - a_{m1}x_1 - a_{m2}x_2 - \ldots - a_{mn}x_n + x_0$
- <u>Claim</u>: The given LP has a feasible solution if and only if the optimal objective value of L' is 0.
- So, all we need to do is to solve L'. This seems to bring us back to the original problem. However, we see that L' is a *simple* LP.

Linear Programming: Solving LP

- (*Complication 1*) What if the initial basic solution is not a feasible solution?
- We construct the following LP, L' in slack form:
- $z = -x_0$
- $x_{n+1} = b_1 - a_{11}x_1 - a_{12}x_2 - \ldots - a_{1nxn} + x_0$
- $x_{n+2} = b_2 - a_{21}x_1 - a_{22}x_2 - \ldots - a_{2nxn} + x_0$
- .
- $x_{n+m} = b_m - a_{m1}x_1 - a_{m2}x_2 - \ldots - a_{mn}x_n + x_0$
- Claim: The given LP has a feasible solution if and only if the optimal objective value of L' is 0.
- Claim: L' is feasible.
- The basic solution might not be a feasible solution since some $b_i < 0$.

Linear Programming: Solving LP

- (*Complication 1*) What if the initial basic solution is not a feasible solution?
- *L'*:
- $z = -x_0$
- $x_{n+1} = b_1 - a_{11}x_1 - a_{12}x_2 - \ldots - a_{1nxn} + x_0$
- $x_{n+2} = b_2 - a_{21}x_1 - a_{22}x_2 - \ldots - a_{2nxn} + x_0$
- .
- $x_{n+m} = b_m - a_{m1}x_1 - a_{m2}x_2 - \ldots - a_{mn}x_n + x_0$
- The basic solution might not be a feasible solution since some $b_i < 0$.
- Let b_i be the smallest among $b_1, \ldots, b_m$. We will pivot using

$$\mathrm{x}_{n+i} = b_i - a_{i1}x_1 - \ldots + x_0$$

Linear Programming: Solving LP

- (*Complication 1*) What if the initial basic solution is not a feasible solution?
- *L'*:
- $z = -x_0$
- $x_{n+1} = b_1 - a_{11}x_1 - a_{12}x_2 - \ldots - a_{1nxn} + x_0$
- $x_{n+2} = b_2 - a_{21}x_1 - a_{22}x_2 - \ldots - a_{2nxn} + x_0$
- .
- $x_{n+m} = b_m - a_{m1}x_1 - a_{m2}x_2 - \ldots - a_{mn}x_n + x_0$
- Let b_i be the smallest among $b_1, \ldots, b_m$. We will pivot using
  $$x_{n+i} = b_i - a_{i1}x_1 - \ldots + x_0$$
- <u>Claim</u>: The basic solution of the LP obtained after the above pivoting is a feasible solution.

Linear Programming: Solving LP

- (*Complication 1*) What if the initial basic solution is not a feasible solution?
- Pre-processing algorithm:
  - Given L, check if all b_i's are positive. In that case return L.
  - Consider L'. Perform the pivoting using the equation with smallest b_i to obtain L''.
  - Solve L'' using Simplex and find the optimal objective value Opt.
  - If ($Opt \neq 0$), then output "LP is infeasible".
  - Otherwise, let L_S be the LP obtained at the end of the simplex. Do the following:
    - If x_0 is a basic variable in L_S, then perform a pivoting step to obtain L_S'.
    - Remove all instances of x_0 and rewrite the objective function of L in terms of non-basic variables of L_S'.

Linear Programming: Solving LP

- (*Complication 1*) What if the initial basic solution is not a feasible solution?
- Pre-processing algorithm: Example
- L:
  - $z = 2x_1 - x_2$
  - $x_3 = 2 - 2x_1 + x_2$
  - $x_4 = -4 - x_1 + 5x_2$
- L':
  - $z = -x_0$
  - $x_3 = 2 - 2x_1 + x_2 + x_0$
  - $x_4 = -4 - x_1 + 5x_2 + x_0$
- L'': After Pivot using ($x_4 = \ldots$)
  - $z = -4 - x_1 + 5x_2 - x_4$
  - $x_3 = 6 - x_1 - 4x_2 + x_4$
  - $x_0 = 4 + x_1 - 5x_2 + x_4$

Linear Programming: Solving LP

- (*Complication 1*) What if the initial basic solution is not a feasible solution?
- Pre-processing algorithm: Example
- L:
  - $z = 2x_1 - x_2$
  - $x_3 = 2 - 2x_1 + x_2$
  - $x_4 = -4 - x_1 + 5x_2$
- L_S:
  - $z = -x_0$
  - $x_2 = 4/5 - x_0/5 + x_1/5 + x_4/5$
  - $x_3 = 14/5 + 4x_0/5 - 9x_1/5 + x_4/5$
- L_S:
  - $z = 2x_1 - x_2 = 2x_1 - (4/5 + x_1/5 + x_4/5) = -4/5 + 9x_1/5 - x_4/5$
  - $x_2 = 4/5 + x_1/5 + x_4/5$
  - $x_3 = 14/5 - 9x_1/5 + x_4/5$

End