ORF 522

Linear Programming and Convex Analysis

Initial solution and particular cases

Marco Cuturi
Reminder: Tableaux

- At each iteration, a tableau for an LP in standard form keeps track of

\[
\begin{array}{cccccc}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
B^{-1}_I M & : & B^{-1}_I & : & B^{-1}_I b \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
c - z^T & \vdots & \vdots & \vdots & \vdots & \vdots \\
& & & & & -x_0
\end{array}
\]

- The pivot rule

\[
\begin{bmatrix}
\vdots & j & \vdots & e & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & a_{ij} & \vdots & a_{ie} & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
r & a_{rj} & \vdots & a_{re} & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
\vdots & j & \vdots & e & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & a_{ij} - \frac{a_{rj}a_{ie}}{a_{re}} & \vdots & 0 & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
r & a_{rj}/a_{re} & \vdots & 1 & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}
\]
Today

Some recipes

- Finding an initial BFS
  - M-method
  - Two phase algorithm

- Simplex Troubleshooting
  - no feasible solution,
  - unbounded feasible set and unbounded objective,
  - infinite number of solutions
  - degeneracy, cycling, and how to avoid it
    - Perturbation
    - Bland’s pivot Rule
Finding an initial BFS
Finding an initial BFS

- So far most examples had an initial feasible solution.
- For most programs given in mixed forms for instance, this is not the case.
- Again, imagine someone solves the problem \((c, A, b)\) before us and finds \(x^*\) as the optimal solution.
- Gives back the problem adding the constraint \(c^T x \geq c^T x^*\).
- Finding an initial feasible solution is equivalent to finding the optimal solution itself!

- The following two methods modify the LP or use an auxiliary LP to look for an initial BFS.
- No major theoretical/methodological change, just a trick.
1. The M-method

- given $A \in \mathbb{R}^{m \times n}$

  maximize $x_0 = c^T x$

  subject to

  \[
  \begin{align*}
  Ax & = b \\
  x & \geq 0
  \end{align*}
  \]

- turn it into the following standard form with $M \gg 0$ and $u \in \mathbb{R}^m$, and $b \geq 0$

  maximize $x_0 = c^T x - M 1^T u$

  subject to

  \[
  \begin{align*}
  Ax + I_m u & = b \\
  x, u & \geq 0
  \end{align*}
  \]

  - Note that this may require some sign-flipping since we need the $b$ on the right to be nonnegative.
  - Just multiply by -1 all lines which correspond to negative $b_i$. 


Princeton ORF-522
1. The M-method

- Using the standard form

\[
\text{maximize } \quad x_0 = c^T x - M \mathbf{1}^T u \\
\text{subject to } \quad \begin{cases} 
Ax + I_m u &= b \\
x, u &\geq 0 
\end{cases}
\]

\(x = 0\) and \(u = b\) is now a **feasible BFS** of the augmented problem.

- Any solution to \(Ax + I_m u = b\) such that \(Ax = b\) necessarily implies \(u = 0\).

- Thus \(u\) must be zero if possible at the end of the optimization.

- Conclusion: \(M\) helps us find an **initial starting point** to run the simplex.
  - If the problem was feasible, it will naturally drive \(u\) to 0 and give the right solution.
  - \(u\) will be quickly driven to 0 if \(M\) is big \(\Rightarrow\) big reduced costs.
  - If at the end the solution \(u\) is not zero then unfeasibility.
1. The M-method

- **Remark**: in some cases, canonical vectors already exist in $A$.
- **Example**: starting with the **non-feasible** canonical form

  \[
  \begin{align*}
  \text{maximize} & \quad x_0 = 2x_1 + x_2 \\
  \text{subject to} & \quad \begin{cases}
  -x_1 + x_2 & \geq 2 \\
  x_1 + x_2 & \leq 1 \\
  x_1, x_2 & \geq 0
  \end{cases}
  \end{align*}
  \]

  - yields the standard form

  \[
  \begin{align*}
  \text{maximize} & \quad x_0 = 2x_1 + x_2 \\
  \text{subject to} & \quad \begin{cases}
  -x_1 + x_2 - x_3 & = 2 \\
  x_1 + x_2 + x_4 & = 1 \\
  x_1, x_2, x_3, x_4 & \geq 0
  \end{cases}
  \end{align*}
  \]

  - The column of $x_4$ is already canonical.

  - We only need to add one artificial variable for the first equation, not two as stated in the general case.
2. The Two-Phase Method

- Instead of considering the **modified** objective
  \[ x_0 = c^T x - M 1^T u , \]
  of the M-method,
- we first maximize the **auxiliary** objective
  \[ x_0 = -1^T u \]
  using the same set of constraints, that is we follow the program

\[
\begin{align*}
\text{maximize} & \quad x_0 = -1^T u \\
\text{subject to} & \quad \begin{cases} \ Ax + I_m u = b \\ x, u \geq 0 \end{cases}
\end{align*}
\]
2. The Two-Phase Method

- **Run the simplex** with the last (artificial) canonical columns as the basis and the initial solution as $b$.

- As with the M-method, we may need to flip the sign of some lines in order to have $b \geq 0$.

- **Three possible scenarios** now:
  
  - The optimum $x_0^* < 0$... unfeasibility.
  - We reach a zero objective $x_0^* = 0$ with a base $I$ which only contains columns of the original $A$. We **keep this base** and run the simplex with that initial BFS.
  - We reach a zero objective but the basis has artificial variables in it set to zero...
2. The Two-Phase Method, driving artificial variables out

- Phase I gives an initial BFS which contains artificial variables set to zero.
- **degeneracy**... We want to use \( \mathbf{I} \) for the original problem. We have to **drive artificial variables out**.
- Suppose the \( l \)th index of the base \( \mathbf{I} \) corresponds to an artificial variable.
- Consider the row \( y_{l,j} \) for \( j \leq n \), that is the \( l \)th entry of all \( B_\mathbf{I}^{-1}A \) columns of the original problem.
- If all entries in the row are zero, the constraint is redundant, we can ignore this variable.
- Otherwise there is a possible pivot with a \( j \)th element. Remember that \( B_\mathbf{I}^{-1}\mathbf{b} \) is zero at \( l \). The solution won’t be changed and will still be feasible.
2. The Two-Phase Method, driving artificial variables out

- Example: our tableau is split between **original** variables, **artificial** variables and current solution $B_I^{-1}b$.

- In this example we have an artificial variable in the basis and a degenerate solution.

- We want to remove the artificial variable and replace it with an original one in order to move to phase 2.

- Do something we haven’t done often: pivot on $-3$ a negative number. That’s fine because the $B_I^{-1}b$ will not be modified, no unfeasibility problem.
1. **M-method**: use a big M to get an initial BFS directly to the modified LP

\[
\begin{align*}
\text{maximize} & \quad x_0 = c^Tx - M1^Tu \\
\text{subject to} & \quad \left\{ \begin{array}{c}
Ax + I_mu = b \\
x, u \geq 0
\end{array} \right.
\end{align*}
\]

after convergence either \( u = 0 \) and we have the solution, either \( u \neq 0 \) and there is unfeasibility.

2. **Two-phase method**:
   - bring \( x_0 \) to zero in phase 1:

\[
\begin{align*}
\text{maximize} & \quad x_0 = -1^Tu \\
\text{subject to} & \quad \left\{ \begin{array}{c}
Ax + I_mu = b \\
x, u \geq 0
\end{array} \right.
\end{align*}
\]

- Remove artificial variables if necessary.
- Use the I as an initial BFS for the original problem and run phase 2.
Simplex Troubleshooting
Unexpected problems when running the simplex

we assume we are using the M or 2-phase methods to look for an initial BFS.

- non-feasibility.
- unboundedness.
- degeneracy and cycling.
- infinite number of solutions.
Non-feasibility

- With an explicit initial BFS (think feasible canonical form) no need to test for feasibility.
- The issue only appears in phase I or using the M-method.
- In both cases there is always an initial BFS to a modified LP.
- The initial feasibility is checked when we have converged ($u = 0$)

- Feasibility is thus either
  - elucidated immediately (there is a feasible obvious point)
  - tested at the end of phase 1 or after the M-method has converged.
Unboundedness

- Remember the theorem

**Theorem** Let x be a basic feasible solution (BFS) to a LP with index set I. If \( \exists \) an index \( e \notin I \) such that \( y_e \leq 0 \) then the feasible region is **unbounded**. If moreover for \( e \) the reduced cost \( c_e - z_e > 0 \) then there exists a feasible solution with at most \( m + 1 \) nonzero variables and an **arbitrary large objective function**.

- Unboundedness of the feasible and of the objective can be easily tested at any time in the algorithm.
Degeneracy

- Degeneracy appears whenever a base solution $x_I$ has a zero.
- That is whenever the column $b$ is l.d. with $m - 1$ columns of $B_I$.
- Degeneracy thus depends on the specific index $I$ selected on that iteration.
- In practice this is rare and more of a theoretical curiosity.
- Intuition: non-degenerate problems are dense in the space of problems.
- In the same way that … matrices are dense in spaces of square matrices.

- Degeneracy implies that the objective on the next iteration may stall.
- No problem if it only happens from time to time.
- The real issue is when it happens repeatedly: cycling
Cycling

- Cycling occurs whenever the simplex loops through a set of index sets $I_1, I_2, I_3, \ldots, I_1, I_2, \ldots$

- Once we return to a previously seen $I$ we can only loop...

- This is even **more rare** than degeneracy.

- Most solvers take this into account (when they do) by **adding small perturbations**. We will see that later.

- Cycling examples in books are like freaks in a circus... you won't see them outside.

- The example with **minimal size** is with 4 variables and 2 constraints, that is dimension 6.
yet we have to consider cycling because

**Theorem 1.** *Given a pivot rule, if the simplex fails to terminate then it must cycle.*

*Proof.* finite number of states, if no termination then one state is visited twice, and since everything is deterministic we just loop. ■

- cycling is the most serious issue with the simplex. but it never happens.
We have the following problem in canonical form, which is an instance of the aforementioned pathological family.

\[
\begin{align*}
\text{maximize} & \quad x_0 = 20x_1 + \frac{1}{2}x_2 - 6x_3 + \frac{3}{4}x_4 \\
\text{subject to} & \quad \begin{cases} 
8x_1 - x_2 + 9x_3 + \frac{1}{4}x_4 & \leq 16 \\
12x_1 - \frac{1}{2}x_2 + 3x_3 + \frac{1}{2}x_4 & \leq 24 \\
x_2 & \leq 2 \\
x_1, x_2, x_3, x_4 & \geq 0
\end{cases}
\end{align*}
\]

We have not mentioned pivot rules. An intuitive pivot rule would be

- Select index \( e \) with highest reduced cost coefficient. If tie choose the lowest.
- Select index \( r \) which ensures minimum ratio. If tie choose the lowest.

We use it on this problem.
• ... ok. let’s check with matlab directly.
Remedies to Cycling
Remedies to Cycling: Perturbation

- The no-brainer approach: perturb the constraints objective $b$ to artificially ensure $b'$s independence from the columns.

- $b \leftarrow b \pm \varepsilon$.

- Matlab demo...

- Conclusion: an ugly hack that works.
Remedies to Cycling: Advanced Pivoting

- A more subtle approach: define alternative rules for pivoting.
- Cycling occurs when we do not know how to handle effectively ties.
- Some more clever rules ensure no cycling:
  - Lexicographic rule.
  - Bland’s rule.
Definition 1. A vector \( \mathbf{u} \in \mathbb{R}^n \) is said to be lexicographically larger (resp. smaller) than \( \mathbf{v} \), written \( \mathbf{u} \succ \mathbf{v} \) if \( \mathbf{u} \neq \mathbf{v} \) and the first nonzero component of \( \mathbf{u} - \mathbf{v} \) is positive (resp. negative).

- \[
\begin{bmatrix}
0 \\
2 \\
4
\end{bmatrix} \succ 
\begin{bmatrix}
0 \\
2 \\
2
\end{bmatrix},
\begin{bmatrix}
0 \\
4 \\
4
\end{bmatrix} \prec 
\begin{bmatrix}
1 \\
2 \\
2
\end{bmatrix}
\]

- When \( \mathbf{u} \succ 0 \), \( \mathbf{u} \) is said to be lexicographically positive.
Lexicographic Pivot Rule

1. Choose any entering index $e$ as long as $c_e - z_e > 0$. Consider $y_e = B^{-1}_I a_e$.

2. Select $I = \arg\min_{k=1,\ldots,m} \left\{ \frac{x_{ik}}{y_{k,e}} \mid y_{k,e} > 0 \right\}$.

   - If $I$ is a singleton pick that index.
   - Otherwise, for each tied index in $I$, divide the corresponding $i$th row of the tableau by $y_{i,e}$. Pick the index $r$ as the index of the row which is lexicographically smallest.

**Remark**: There is always a unique choice for $r$. If not that would mean that two rows of $B^{-1}A$ are proportional, hence rank is smaller than $m$, hence $A$ also has rank smaller than $m$. 
Lexicographic Pivot Rule

- Example: suppose we want to enter the third variable here

\[
\begin{array}{ccc|c}
0 & 5 & 3 & \cdots & 1 \\
4 & 6 & -1 & \cdots & 2 \\
0 & 7 & 9 & \cdots & 3 \\
\end{array}
\]

- There is a tie between the first and third elements, both \(\frac{1}{3} = \frac{3}{9}\). We divide the first and third row by \(y_{1,3} = 3\) and \(y_{3,3} = 9\) respectively.

\[
\begin{array}{ccc|c}
0 & \frac{5}{3} & 1 & \cdots & \frac{1}{3} \\
* & * & * & \cdots & * \\
0 & \frac{7}{9} & 1 & \cdots & \frac{1}{3} \\
\end{array}
\]

- Pick \(r = 3\) this time since the third row is lexicographically smaller.
Theorem 2. Suppose that the simplex algorithm starts with all rows in the upper tableau lexicographically positive. Suppose that the lexicographic pivot rule is observed. Then

- Every row of the upper simplex tableau remains lexicographically positive throughout the algorithm.
- The lower tableau, i.e. the reduced cost coefficient and the objective, strictly increase lexicographically at each iteration.
- The simplex method terminates after a finite number of iterations.
- In practice: too expensive computationally. Never used.
- cheaper alternative
Bland’s Pivot Rule

**Theorem 3.** Using the following pivot rule:

1. Set the index $e$ as the smallest index in \( \{ i \mid 1 \leq i \leq n, c_i - z_i > 0 \} \).

2. Define $I = \arg\min_{k=1,\ldots,m} \left\{ \frac{x_{ik}}{y_{k,e}} \mid y_{k,e} > 0 \right\}$ and pick $r$ as the smallest element of $I$.

The simplex method *terminates after a finite number of iterations.*

- Also called smallest subscript rule.
- More efficient and cheaper.
A theorem we proved tells us that (if it exists) the optimum is reached by at least one extreme point.

The simplex only checks for extreme points in the argmax, and guarantees to find one.

Yet, when the objective is colinear with one of the constraints, the optimal region might be a convex set.

Any combination of the different extreme points (possibly found through different pivot rules) is still a solution.
Short comment on infinite number of optimal solutions

- simple example in 2-dimensions.
Next Lecture

- Complexity and Efficiency.
- Klee-Minty Counterexample.
- Some words on more clever implementations of the simplex.