# 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

$$
\begin{array}{cccccc|c}
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\
\vdots & B_{\mathbf{I}}^{-1}M & \vdots & \vdots & B_{\mathbf{I}}^{-1} & \vdots & B_{\mathbf{I}}^{-1}\mathbf{b} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\
\hline
\cdots & & \mathbf{c}-\mathbf{z}^T & & & \cdots & -x_0
\end{array}
$$

- The pivot rule

$$
\begin{array}{c}
 \\ \vdots \\ i \\ \vdots \\ r \\ \vdots
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \cdots & j & \cdots & e & \cdots \end{array} \\
\begin{bmatrix}
\ddots & \vdots & \ddots & \vdots & \ddots \\
\cdots & a_{ij} & \cdots & \boldsymbol{a_{ie}} & \cdots \\
\ddots & \vdots & \ddots & \vdots & \ddots \\
\cdots & \boldsymbol{a_{rj}} & \cdots & \boldsymbol{a_{re}} & \cdots \\
\ddots & \vdots & \ddots & \vdots & \ddots
\end{bmatrix}
\end{array}
\Rightarrow
\begin{array}{c}
 \\ \vdots \\ i \\ \vdots \\ r \\ \vdots
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \cdots & j & \cdots & e & \cdots \end{array} \\
\begin{bmatrix}
\ddots & \vdots & \ddots & \vdots & \ddots \\
\cdots & a_{ij} - \frac{\boldsymbol{a_{rj}}\boldsymbol{a_{ie}}}{\boldsymbol{a_{re}}} & \cdots & 0 & \cdots \\
\ddots & \vdots & \ddots & \vdots & \ddots \\
\cdots & \boldsymbol{a_{rj}}/\boldsymbol{a_{re}} & \cdots & \mathbf{1} & \cdots \\
\ddots & \vdots & \ddots & \vdots & \ddots
\end{bmatrix}
\end{array}
$$

# 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 $(\mathbf{c}, A, \mathbf{b})$ before us and finds $\mathbf{x}^\star$ as the optimal solution.
- Gives back the problem adding the constraint $\mathbf{c}^T\mathbf{x} \geq \mathbf{c}^T\mathbf{x}^\star$.
- 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 $A \in \mathbf{R}^{m \times n}$

$$\begin{array}{ll} \text{maximize} & x_0 = \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \left\{ \begin{array}{rcl} A\mathbf{x} & = & \mathbf{b} \\ \mathbf{x} & \geq & 0 \end{array} \right. \end{array}$$

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

$$\begin{array}{ll} \text{maximize} & x_0 = \mathbf{c}^T \mathbf{x} - M \mathbb{1}^T \mathbf{u} \\ \text{subject to} & \left\{ \begin{array}{rcl} A\mathbf{x} + I_m \mathbf{u} & = & \mathbf{b} \\ \mathbf{x}, \mathbf{u} & \geq & 0 \end{array} \right. \end{array}$$

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

# 1. The M-method

- Using the standard form

$$\begin{array}{ll} \text{maximize} & x_0 = \mathbf{c}^T\mathbf{x} - M\mathbb{1}^T\mathbf{u} \\ \text{subject to} & \left\{ \begin{array}{rcl} A\mathbf{x} + I_m\mathbf{u} & = & \mathbf{b} \\ \mathbf{x}, \mathbf{u} & \geq & 0 \end{array} \right. \end{array}$$

  $\mathbf{x} = 0$ and $\mathbf{u} = \mathbf{b}$ is now a **feasible BFS** of the augmented problem.

- Any solution to $A\mathbf{x} + I_m\mathbf{u} = \mathbf{b}$ such that $A\mathbf{x} = \mathbf{b}$ necessarily implies $\mathbf{u} = \mathbf{0}$.
- Thus $\mathbf{u}$ 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 $\mathbf{u}$ to 0 and give the right solution.
  - $\mathbf{u}$ will be quickly driven to $\mathbf{0}$ if $M$ **is big** $\Rightarrow$ big reduced costs.
  - If at the end the solution $\mathbf{u}$ 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

$$\text{maximize} \quad x_0 = 2x_1 + x_2$$
$$\text{subject to} \quad \begin{cases} -x_1 + x_2 \geq 2 \\ x_1 + x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases}$$

- yields the standard form

$$\text{maximize} \quad x_0 = 2x_1 + x_2$$
$$\text{subject to} \quad \begin{cases} -x_1 + x_2 - x_3 = 2 \\ x_1 + x_2 + \boldsymbol{x_4} = 1 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

- The column of $\boldsymbol{x_4}$ 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

$$x_0 = \mathbf{c}^T\mathbf{x} - M\mathbb{1}^T\mathbf{u},$$

of the M-method,

- we first maximize the **auxiliary** objective

$$x_0 = -\mathbb{1}^T\mathbf{u}$$

using the same set of constraints, that is we follow the program

$$\begin{array}{ll} \text{maximize} & x_0 = -\mathbb{1}^T\mathbf{u} \\ \text{subject to} & \left\{ \begin{array}{rcl} A\mathbf{x} + I_m\mathbf{u} & = & \mathbf{b} \\ \mathbf{x}, \mathbf{u} & \geq & 0 \end{array} \right. \end{array}$$

# 2. The Two-Phase Method

- **Run the simplex** with the last (artificial) canonical columns as the basis and the initial solution as $\mathbf{b}$
- As with the M-method, we may need to flip the sign of some lines in order to have $\mathbf{b} \geq 0$.
- **Three possible scenarios** now:
  - The optimum $\mathbf{x}_0^\star < 0$... unfeasibility.
  - we reach a zero objective $\mathbf{x}_0^\star = 0$ with a base $\mathbf{I}$ 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 intial 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 $B_{\mathbf{I}}^{-1}\mathbf{b}$.
- In this example we have an artificial variable in the basis and a degenerate solution.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | $\cdots$ | $\cdots$ | **0** | $\cdots$ | 3 |
| 0 | $-3$ | $\cdots$ | $\cdots$ | **1** | $\cdots$ | 0 |
| 5 | 6 | $\cdots$ | $\cdots$ | **0** | $\cdots$ | 2 |

- 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 $B_{\mathbf{I}}^{-1}\mathrm{b}$ **will not be modified**, no unfeasibility problem.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | **0** | $\cdots$ | $\cdots$ | $\frac{2}{3}$ | $\cdots$ | **3** |
| 0 | **1** | $\cdots$ | $\cdots$ | $-\frac{\mathbf{1}}{\mathbf{3}}$ | $\cdots$ | **0** |
| 5 | **0** | $\cdots$ | $\cdots$ | 2 | $\cdots$ | **2** |

# Wrap up

1. **M-method**: use a big M to get an initial BFS directly to the **modified** LP

$$\begin{array}{ll} \text{maximize} & x_0 = \mathbf{c}^T\mathbf{x} - M\mathbb{1}^T\mathbf{u} \\ \text{subject to} & \left\{ \begin{array}{ccc} A\mathbf{x} + I_m\mathbf{u} & = & \mathbf{b} \\ \mathbf{x}, \mathbf{u} & \geq & 0 \end{array} \right. \end{array}$$

   after convergence either $\mathbf{u} = \mathbf{0}$ and we have the solution, either $\mathbf{u} \neq \mathbf{0}$ and there is unfeasibility.

2. **Two-phase method**:

   - bring $x_0$ to zero in phase 1:

$$\begin{array}{ll} \text{maximize} & x_0 = -\mathbb{1}^T\mathbf{u} \\ \text{subject to} & \left\{ \begin{array}{ccc} A\mathbf{x} + I_m\mathbf{u} & = & \mathbf{b} \\ \mathbf{x}, \mathbf{u} & \geq & 0 \end{array} \right. \end{array}$$

   - Remove artificial variables if necessary.
   - Use the $\mathbf{I}$ 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 $(\mathbf{u} = \mathbf{0})$

- 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* $\mathbf{x}$ *be a* $\boldsymbol{basic\ feasible\ solution\ (BFS)}$ *to a LP with index set* $I$*. If* $\exists$ *an index* $e \notin \mathbf{I}$ *such that* $\mathbf{y}_e \leq 0$ *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* $\mathbf{m+1}$ *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 $x_{\mathbf{I}}$ has a zero.
- That is whenever the column $b$ is l.d. with $m-1$ columns of $B_{\mathbf{I}}$.
- Degeneracy thus depends on the **specific index $\mathbf{I}$** 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 $\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3, \cdots, \mathbf{I}_1, \mathbf{I}_2, \cdots$
- Once we return to a previously seen $\mathbf{I}$ 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 wont see them outside.
- The example with **minimal size** is with 4 variables and 2 constraints, that is dimension 6.

# Cycling

- 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.

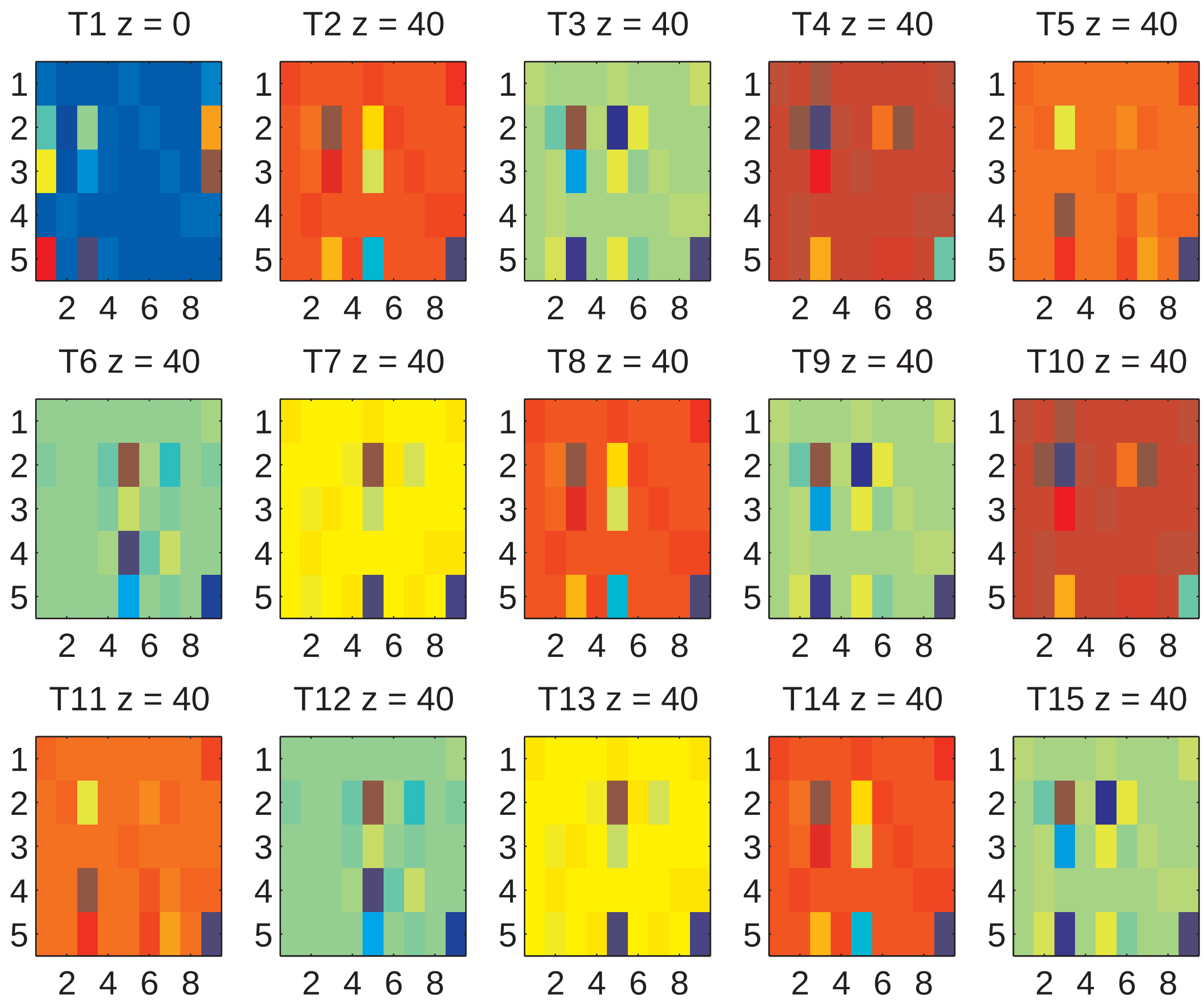
# Cycling

- We have the following problem in canonical form, which is an instance of the aforementioned pathological family.

$$
\begin{array}{ll}
\text{maximize} & x_0 = 20x_1 + \frac{1}{2}x_2 - 6x_3 + \frac{3}{4}x_4 \\
\text{subject to} & \left\{ \begin{array}{rcrcrcrcr}
x_1 & & & & & & & \leq & 2 \\
8x_1 & - & x_2 & + & 9x_3 & + & \frac{1}{4}x_4 & \leq & 16 \\
12x_1 & - & \frac{1}{2}x_2 & + & 3x_3 & + & \frac{1}{2}x_4 & \leq & 24 \\
 & & x_2 & & & & & \leq & 1 \\
 & & & & & & x_1, x_2, x_3, x_4 & \geq & 0
\end{array} \right.
\end{array}
$$

- 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.

# Cycling: Matlab example




- ... ok.. let's check with matlab directly.

# Remedies to Cycling

# Remedies to Cycling: Perturbation

- The no-brainer approach: perturb the constraints objective $\mathbf{b}$ 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.

# Lexicographic Rule

**Definition 1.** *A vector* $\mathbf{u} \in \mathbf{R}^n$ *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).*

- $\begin{bmatrix} 0 \\ 2 \\ 1 \\ 4 \end{bmatrix} \succ \begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \\ 1 \\ 4 \end{bmatrix} \prec \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}$
- When $\mathbf{u} \succ \mathbf{0}$, $\mathbf{u}$ is said to be lexicographically positive.

# Lexicographic Pivot Rule

1. Choose any entering index $e$ as long as $c_e - z_e > 0$. Consider $\mathbf{y}_e = B_{\mathbf{I}}^{-1}\mathbf{a}_e$.

2. Select $I = \operatorname{argmin}_{k=1,..,m} \left\{ \frac{x_{i_k}}{y_{k,e}} | \; y_{k,e} > 0 \right\}$.
   - 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

$$\begin{array}{|cccc|c|} \hline 0 & 5 & 3 & \cdots & 1 \\ 4 & 6 & -1 & \cdots & 2 \\ 0 & 7 & 9 & \cdots & 3 \\ \hline \end{array}$$

- 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.

$$\begin{array}{|cccc|c|} \hline 0 & \frac{5}{3} & 1 & \cdots & \frac{1}{3} \\ * & * & * & \cdots & * \\ 0 & \frac{7}{9} & 1 & \cdots & \frac{1}{3} \\ \hline \end{array}$$

- Pick $r = 3$ this time since the third row is lexicographically smaller.

# Lexicographic Pivot Works

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

2. *Define* $I = \operatorname{argmin}_{k=1,..,m} \left\{ \frac{x_{i_k}}{y_{k,e}} | \; y_{k,e} > 0 \right\}$ *and pick* $\boldsymbol{r}$ *as the smallest element of* $I$.

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

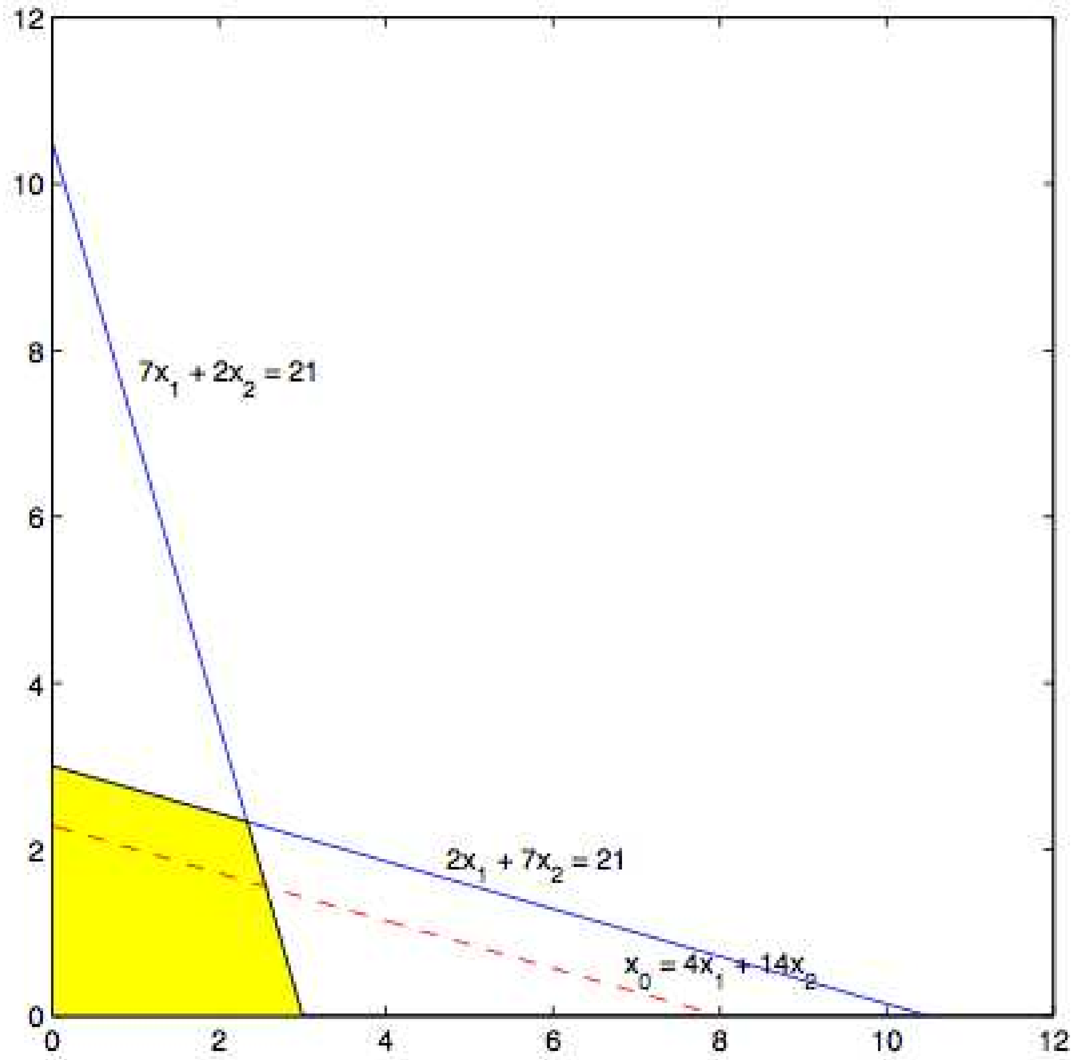
- Also called smallest subscript rule.
- More efficient and cheaper.

# Short comment on infinite number of optimal solutions

- 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.