Optimization Theory and Applications

Kun Zhu (zhukun@nuaa.edu.cn)

December 19, 2018

General Integer Programming Problem

- An *integer programming problem* is a mathematical optimization program in which all of the variables are restricted to be integers
- A general integer program has the form

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbb{Z}^n \end{aligned}$$

- In general, integer programming is NP-hard

Integer Linear Programming

- *Integer linear programming (ILP)*: Linear programs with additional constraint that the variables are integers
- A general ILP has the form

$$\begin{array}{rl} \text{minimize} & \mathbf{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \mathbf{0} \\ & \boldsymbol{x} \in \mathbb{Z}^n. \end{array}$$

Variants of Integer Linear Programming

- Mixed integer linear programming (MILP):
  - Some of the variables are constrained to be integers, while others are allowed to be non-integers
- 0-1 binary integer linear programming
  - The variables are restricted to be either 0 or 1
  - Note that any bounded integer variable can be expressed as a combination of binary variables

    E.g., for $0 \leq x \leq b$, the variable x can be expressed using $\lfloor \log_2 b \rfloor + 1$ binary variables

    $$x = x_1 + 2x_2 + 4x_3 + \cdots + 2^{\lfloor \log_2 b \rfloor} x_{\lfloor \log_2 b \rfloor + 1}$$
  - 0-1 binary integer linear programming is one Karp's 21 NP-complete problems

Applications of Integer Linear Programming

- Examples of problems that can be formulated as ILPs
  - Traveling salesman
  - Knapsack (Packing) problem (e.g., WDP in combinational auction)
  - Assignment problem (e.g., channel assignment)
- Two main motivations for using integer variables:
  - The integer variables represent quantities that can only be integer
  - The integer variables represent decisions and so should only take on the value 0 and 1

Applications of Integer Linear Programming

- Example: shipping problem

Cargo	Volume (m^3/box)	Weight (100kg/box)	Profit ($/box)
1	5	2	20
2	4	5	10
Transportation Restriction	24 m^3	1300 kg	

- How to ship the cargoes to have the largest profit?

Applications of Integer Linear Programming

- Denote x_1, x_2 the quantities of the shipped box of cargo 1 and 2, respectively
- The problem can be formulated as an ILP as

$$\begin{aligned} \text{maximize} \quad & 20x_1 + 10x_2 \\ \text{subject to} \quad & 5x_1 + 4x_2 \leq 24 \\ & 2x_1 + 5x_2 \leq 13 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{aligned}$$

Solving Methods for ILP

- Intuitive idea: LP relaxation of the ILP
- Consider special case: using unimodularity
- Exact algorithms
  - Cutting plane method
  - Dynamic programming
  - Branch and bound method
  - Branch and cut method
- Heuristic methods

LP Relaxation

- Main idea:
  - Remove the constraint that $\mathbf{x}$ is integer
  - Solve the corresponding LP
  - Then round the entries of the solution to the LP relaxation
- Problems:
  - The solution may not be optimal
  - The solution may not even be feasible
- However, optimal objective value of LP relaxation gives a lower bound of the ILP problem

LP Relaxation

- *Example*: the original ILP

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 5x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_2 \leq 6 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{aligned}$$

- The LP Relaxation is

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 5x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$

- The solution is $x_1 = 2, x_2 = 6$

- Luckily, we get the solution by LP relaxation

LP Relaxation

- *Another Example*: the original ILP

$$\begin{aligned} \text{maximize} \quad & 7x_1 + 6x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \leq 3 \\ & x_1 + 4x_2 \leq 4 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{aligned}$$

- The LP Relaxation is

$$\begin{aligned} \text{maximize} \quad & 7x_1 + 6x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \leq 3 \\ & x_1 + 4x_2 \leq 4 \\ & x_1, x_2 \geq 0 \end{aligned}$$

- The solution is $x_1 = 8/7, x_2 = 5/7$
- We round the solution and get $x_1 = 1, x_2 = 1$
- Question: is it a feasible solution?

Special Case: ILP with Unimodularity

- In general, the solution of the LP relaxation will not be optimal
- However, for a special case, where $\mathbf{A}$ is unimodular, and $\mathbf{b} \in \mathbb{Z}^n$, every basic feasible solution is integral
- Consequently, the solution returned by the simplex algorithm is guaranteed to be integral
- ***Definition***: An $m \times n$ integer matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}, m < n$, is ***unimodular*** if all its nonzero mth-order minors (the determinant of the mth square sub-matrix) are ± 1 (i.e., either 1 or -1)

Special Case: ILP with Unimodularity

- Consider the linear system $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{Z}^{m \times n}, m \leq n$
- Let $\mathbf{B}$ be a basis
- Then the unimodularity of $\mathbf{A}$ is equivalent to $|det\ \mathbf{B}| = 1$ for any such $\mathbf{B}$
- The following lemma connects unimodularity with integer basic solutions

Special Case: ILP with Unimodularity

- ***Lemma***: Consider the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{Z}^{m\times n}, m \leq n$ is unimodular and $\mathbf{b} \in \mathbb{Z}^m$. Then all basic solutions are integral
- Proof:
  - Suppose the first m columns of $\mathbf{A}$ constitute a basis $\mathbf{B}$, then the corresponding basic solution is

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

  - Since all elements of $\mathbf{A}$ are integers, B is an integer matrix
  - Moreover, because $\mathbf{A}$ is unimodular, $|det\mathbf{B}| = 1$, which implies that $\mathbf{B}^{-1}$ is also an integer matrix
  - Therefore, $\mathbf{x}^*$ is an integer vector

Special Case: ILP with Unimodularity

- ***Corollary***: Consider the LP constraint $\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$, where $\mathbf{A}$ is unimodular, $\mathbf{A} \in \mathbb{Z}^{m \times n}$, and $\mathbf{b} \in \mathbb{Z}^m$. Then all basic feasible solutions are integral
- Unimodularity allows us to solve an ILP problem by solving the corresponding LP problem

$$\begin{array}{rl} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{x} \in \mathbb{Z}^n \end{array} \quad \Leftrightarrow \quad \begin{array}{rl} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{array}$$

If $\mathbf{A}$ is unimodular and $\mathbf{b} \in \mathbb{Z}^m$

Special Case: ILP with Unimodularity

- Example: Consider the following ILP problem

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 5x_2 \\ \text{subject to} \quad & x_1 + x_3 = 4 \\ & x_2 + x_4 = 6 \\ & x_1 + x_2 + x_5 = 8 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \\ & x_1, x_2, x_3, x_4, x_5 \in \mathbb{Z} \end{aligned}$$

- We can write this problem in matrix form with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 4 \\ 6 \\ 8 \end{bmatrix}$$

Special Case: ILP with Unimodularity

- Note that $\mathbf{b} \in \mathbb{Z}^3$
- Moreover, it is easy to check that $\mathbf{A}$ is unimodular
- Hence, the ILP problem above can be solved by solving the LP problem

$$
\begin{aligned}
\text{maximize} \quad & 2x_1 + 5x_2 \\
\text{subject to} \quad & x_1 + x_3 = 4 \\
& x_2 + x_4 = 6 \\
& x_1 + x_2 + x_5 = 8 \\
& x_1, x_2, x_3, x_4, x_5 \geq 0.
\end{aligned}
$$

- The optimal solution is $[2, 6, 2, 0, 0]^T$

Special Case: ILP with Unimodularity

- Next, we consider the ILP problems of the form

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \\ & \boldsymbol{x} \in \mathbb{Z}^n \end{aligned}$$

- We can transform it into standard form with $[\mathbf{A}, \mathbf{I}]\mathbf{y} = \mathbf{b}$ ($\mathbf{y}$ contains $\mathbf{x}$ and the slack variables)
- To deal with matrix of the form $[\mathbf{A}, \mathbf{I}]$, we need another definition

Special Case: ILP with Unimodularity

- ***Definition***: An $m \times n$ integer matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is ***totally unimodular*** if all its nonzero minors are ± 1
- From this definition, if an integer matrix is totally unimodular, then each entry is 0, 1, or -1
- The next proposition relates the total unimodularity of $\mathbf{A}$ with the unimodularity of $[\mathbf{A}, \mathbf{I}]$

Special Case: ILP with Unimodularity

- ***Proposition***: If an $m \times n$ integer matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is totally unimodular, then the matrix $[\mathbf{A}, \mathbf{I}]$ is unimodular
- Combining the result above with previous lemma, we obtain the following corollary
- ***Corollary***: Consider the LP constraint

$$\begin{aligned} [\mathbf{A}, \mathbf{I}]\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq 0 \end{aligned}$$

  where $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is totally unimodular and $\mathbf{b} \in \mathbb{Z}^m$. Then, all basic feasible solutions have integer components

Special Case: ILP with Unimodularity

- Total unimodularity of $\mathbf{A}$ allows us to solve an ILP problem of the following form by solving the corresponding LP problem

$$\begin{array}{rl} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \\ & \boldsymbol{x} \in \mathbb{Z}^n \end{array} \quad \Leftrightarrow \quad \begin{array}{rl} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0}. \end{array}$$

Special Case: ILP with Unimodularity

- Example: Consider the following ILP problem

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 5x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_2 \leq 6 \\ & x_1 + x_2 \leq 8 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

- This problem can be written in the matrix form with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 4 \\ 6 \\ 8 \end{bmatrix}$$

Special Case: ILP with Unimodularity

- It is easy to check that $\mathbf{A}$ is totally unimodular
- Hence, the ILP problem can be solved by solving the LP problem

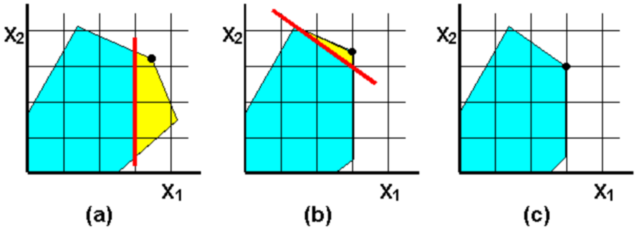
$$\begin{array}{ll} \text{maximize} & 2x_1 + 5x_2 \\ \text{subject to} & x_1 + x_3 = 4 \\ & x_2 + x_4 = 6 \\ & x_1 + x_2 + x_5 = 8 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0, \end{array}$$

Exact Algorithm: Cutting-Plane Method

- Proposed by R. E. Gomory in 1958
- Main idea:
  - Solve the LP relaxation of the given ILP
  - Check the solution
    - If the optimal solution is integer, then we get the solution
    - If not, we add additional constraints to remove the noninteger optimal solutions from the feasible set
  - The process is repeated until the optimal solution is an integer vector

Exact Algorithm: Cutting-Plane Method

- The additional constraints are referred as Gomory cuts and they will not eliminate integer feasible solutions from the feasible set
- An illustrative figure (maximizing $x_1 + x_2$ under certain constraints)



Exact Algorithm: Cutting-Plane Method

- To begin, we first introduce the floor operator
- Definition: The floor of a real number, denoted $\lfloor x \rfloor$, is the integer obtained by rounding x toward $-\infty$
- For example: $\lfloor 3.4 \rfloor = 3$ and $\lfloor -3.4 \rfloor = -4$

Exact Algorithm: Cutting-Plane Method

- Consider the ILP problem

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{x} \in \mathbb{Z}^n \end{aligned}$$

- We begin by obtaining an optimal basic feasible solution to the LP problem

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

Exact Algorithm: Cutting-Plane Method

- Suppose the first m columns form the basis. The corresponding canonical augmented matrix is

$$
\begin{array}{cccccccccc}
\boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_i & \cdots & \boldsymbol{a}_m & \boldsymbol{a}_{m+1} & \cdots & \boldsymbol{a}_n & \boldsymbol{y}_0 \\
1 & 0 & \cdots & 0 & \cdots & 0 & y_{1,m+1} & \cdots & y_{1,n} & y_{10} \\
0 & 1 & \cdots & 0 & \cdots & 0 & y_{2,m+1} & \cdots & y_{2,n} & y_{20} \\
\vdots & \vdots & & \vdots & & \vdots & & & \vdots & \vdots \\
0 & 0 & \cdots & 1 & \cdots & 0 & y_{i,m+1} & \cdots & y_{i,n} & y_{i0} \\
\vdots & \vdots & & \vdots & & \vdots & & & \vdots & \vdots \\
0 & 0 & \cdots & 0 & \cdots & 1 & y_{m,m+1} & \cdots & y_{m,n} & y_{m0}
\end{array}
$$

- Consider the ith component of the optimal basic feasible solution, y_{i0}. Suppose that y_{i0} is not an integer
- Note that any feasible vector $\mathbf{x}$ satisfies the equality constraint

$$x_i + \sum_{j=m+1}^{n} y_{ij} x_j = y_{i0}$$

Exact Algorithm: Cutting-Plane Method

- Based on this, we can derive an additional constraint
  - Eliminate the current optimal noninteger solution from the feasible set
  - Keep any integer feasible solution
- To see how, consider the inequality constraint

$$x_i + \sum_{j=m+1}^{n} \lfloor y_{ij} \rfloor x_j \le y_{i0}$$

- Observations:
  - Since $\lfloor y_{ij} \rfloor \le y_{ij}$, any $\mathbf{x} \ge 0$ that satisfies the above equality constraint also satisfies this inequality constraint
  - Accordingly, any feasible $\mathbf{x}$ satisfies this inequality constraint
- Moreover, for any integer feasible vector $\mathbf{x}$, the left-hand side is an integer. Therefore, we can have

$$x_i + \sum_{j=m+1}^{n} \lfloor y_{ij} \rfloor x_j \le \lfloor y_{i0} \rfloor$$

Exact Algorithm: Cutting-Plane Method

- Subtracting the above inequality from the equation, we deduce that any integer feasible solution satisfies

$$\sum_{j=m+1}^{n}(y_{ij} - \lfloor y_{ij} \rfloor)x_j \geq y_{i0} - \lfloor y_{i0} \rfloor$$

- Note that the non-integer optimal basic feasible solution does not satisfy this inequality (since the left side is 0, while the right side is positive)

- With this new constraint, the current optimal BFS is no longer feasible, but every integer feasible vector remains feasible

- This new constraint is called a Gomory cut

Exact Algorithm: Cutting-Plane Method

- We can transform the new LP into standard form by introducing a surplus variable x_{n+1}

$$\sum_{j=m+1}^{n}(y_{ij} - \lfloor y_{ij} \rfloor)x_j - x_{n+1} = y_{i0} - \lfloor y_{i0} \rfloor$$

- We can now solve the new LP using simplex method and check the resulting optimal BFS
    - If it is integer, then we are done
    - If not, then we introduce another Gomory cut and repeat the process
- This is the *Gomory cutting-plane method*
- Note that the additional variables introduced by slack variables are not constrained to be integers

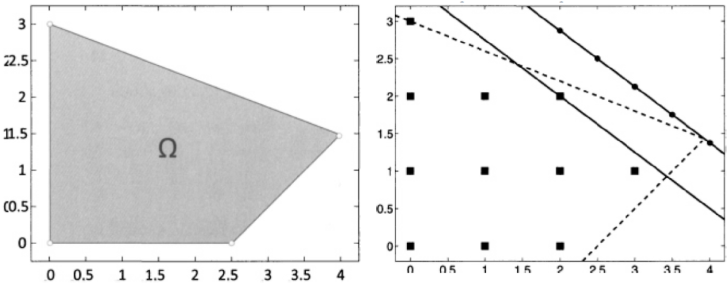
Exact Algorithm: Cutting-Plane Method

- Example: Consider the following ILP problem

$$
\begin{aligned}
\text{maximize} \quad & 3x_1 + 4x_2 \\
\text{subject to} \quad & \frac{2}{5}x_1 + x_2 \leq 3 \\
& \frac{2}{5}x_1 - \frac{2}{5}x_2 \leq 1 \\
& x_1, x_2 \geq 0 \\
& x_1, x_2 \in \mathbb{Z}.
\end{aligned}
$$

Exact Algorithm: Cutting-Plane Method

- First solve the problem graphically



Exact Algorithm: Cutting-Plane Method

- We now solve the problem using the Gomory cutting-plane method
- First we represent the associated LP problem in standard form

$$\begin{array}{ll} \text{maximize} & 3x_1 + 4x_2 \\ \text{subject to} & \frac{2}{5}x_1 + x_2 + x_3 = 3 \\ & \frac{2}{5}x_1 - \frac{2}{5}x_2 + x_4 = 1 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{array}$$

- And the first tableau is

	$\boldsymbol{a}_1$	$\boldsymbol{a}_2$	$\boldsymbol{a}_3$	$\boldsymbol{a}_4$	$\boldsymbol{b}$
	$\frac{2}{5}$	1	1	0	3
	$\frac{2}{5}$	$-\frac{2}{5}$	0	1	1
$\boldsymbol{c}^\top$	-3	-4	0	0	0

Exact Algorithm: Cutting-Plane Method

- We perform elementary operations and obtain

$$\begin{array}{cccccc} & \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 & \boldsymbol{a}_4 & \boldsymbol{b} \\ & 0 & 1 & \frac{10}{14} & -\frac{10}{14} & \frac{20}{14} \\ & 1 & 0 & \frac{10}{14} & \frac{25}{14} & \frac{55}{14} \\ \boldsymbol{r}^\top & 0 & 0 & 5 & \frac{5}{2} & \frac{35}{2} \end{array}$$

- The corresponding optimal basic feasible solution is

  $$[\tfrac{55}{14}, \tfrac{10}{7}, 0, 0]^T$$

  which does not satisfy the integer constraints

Exact Algorithm: Cutting-Plane Method

- We start introducing the Gomory cut corresponding to the first row of the tableau. We obtain

$$\frac{10}{14}x_3 + \frac{4}{14}x_4 - x_5 = \frac{6}{14}$$

- We add this constraint to our tableau

	$\boldsymbol{a}_1$	$\boldsymbol{a}_2$	$\boldsymbol{a}_3$	$\boldsymbol{a}_4$	$\boldsymbol{a}_5$	$\boldsymbol{b}$
	0	1	$\frac{10}{14}$	$-\frac{10}{14}$	0	$\frac{20}{14}$
	1	0	$\frac{10}{14}$	$\frac{25}{14}$	0	$\frac{55}{14}$
	0	0	$\frac{10}{14}$	$\frac{4}{14}$	-1	$\frac{6}{14}$
$\boldsymbol{r}^\top$	0	0	5	$\frac{5}{2}$	0	$\frac{35}{2}$

Exact Algorithm: Cutting-Plane Method

- Perform some elementary operations and obtain

	$\boldsymbol{a}_1$	$\boldsymbol{a}_2$	$\boldsymbol{a}_3$	$\boldsymbol{a}_4$	$\boldsymbol{a}_5$	$\boldsymbol{b}$
	0	1	0	-1	1	1
	1	0	0	$\frac{3}{2}$	1	$\frac{7}{2}$
	0	0	1	$\frac{2}{5}$	$-\frac{7}{5}$	$\frac{3}{5}$
$\boldsymbol{r}^\top$	0	0	0	$\frac{1}{2}$	7	$\frac{29}{2}$

- The corresponding optimal BFS is $[7/2, 1, 3/5, 0, 0]^T$, which is still not integer

- Next, we construct the Gomory cut for the second row of the tableau

$$\frac{1}{2}x_4 - x_6 = \frac{1}{2}$$

Exact Algorithm: Cutting-Plane Method

- We add this constraint to our tableau to obtain

$$\begin{array}{cccccccc} & \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 & \boldsymbol{a}_4 & \boldsymbol{a}_5 & \boldsymbol{a}_6 & \boldsymbol{b} \\ & 0 & 1 & 0 & -1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & \frac{3}{2} & 1 & 0 & \frac{7}{2} \\ & 0 & 0 & 1 & \frac{2}{5} & -\frac{7}{5} & 0 & \frac{3}{5} \\ & 0 & 0 & 0 & \frac{1}{2} & 0 & -1 & \frac{1}{2} \\ \boldsymbol{r}^\top & 0 & 0 & 0 & \frac{1}{2} & 7 & 0 & \frac{29}{2} \end{array}$$

- Perform some elementary operations, we get

$$\begin{array}{cccccccc} & \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 & \boldsymbol{a}_4 & \boldsymbol{a}_5 & \boldsymbol{a}_6 & \boldsymbol{b} \\ & 0 & 1 & 0 & 0 & 1 & -2 & 2 \\ & 1 & 0 & 0 & 0 & 1 & 3 & 2 \\ & 0 & 0 & 1 & 0 & -\frac{7}{5} & \frac{4}{5} & \frac{1}{5} \\ & 0 & 0 & 0 & 1 & 0 & -2 & 1 \\ \boldsymbol{r}^\top & 0 & 0 & 0 & 0 & 7 & 1 & 14 \end{array}$$

- We obtain the optimal BFS $[2, 2]^T$ which is also the solution to the ILP

Exact Algorithm: Cutting-Plane Method

- Note that in previous example, only the first two components are integers, the slack variables and variables introduced by the Gomory cuts are not constrained to be integers
- However, if we have only integer values in constraint data, then, all variables are automatically integer

Further Reading

- Algorithms
  - Dynamic programming
  - Branch and bound method
  - Branch and cut method
  - Hungarian algorithm (for assignment problem)
- Knapsack problems
  - Single knapsack problem
  - Multiple knapsack problem
  - Reference: Knapsack problems algorithms and computer implementations