Operations Research - Lecture 6

Olariu E. Florentin

November 3, 2025

Table of contents

Introduction

- Many discrete optimization problems can be modeled as an Integer Linear Programming problem (*ILP* or simply *IP*).
- A pure ILP problem is an LP problem with the additional restriction that all variables are integer

$$\boxed{\begin{array}{rl} \text{maximize} & z = c^T x, \\ \text{subject to} & Ax \leqslant b, \\ & x \in \mathbb{Z}^n. \end{array} \qquad (1)}$$

- From a (more) geometric point of view an ILP problem requires to maximize (or minimize) a linear function $c^T x$ over the integer vectors of the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leqslant b\}$.
- The problem (1) can be written as

$$\max\{c^T x : x \in \mathcal{P}, x \in \mathbb{Z}^n\}. \qquad (2)$$

Introduction

- Sometimes such problems are *mixed*: some, but not all, of the variables are constrained to be integer and the rest of them are unrestricted.
- In this way we get the Mixed Integer Linear Programming (*MILP* or *MIP*) problem:

$$\begin{aligned} \text{maximize} \quad & z = \mathrm{c}^T \mathrm{x}, \\ \text{subject to} \quad & \mathrm{Ax} \leqslant \mathrm{b}, \\ & x_i \in \mathbb{Z}, \forall i \in \mathcal{I}. \end{aligned} \tag{3}$$

where $\varnothing \neq \mathcal{I} \subsetneq \{1, 2, \ldots, n\}$.

- We next present some discrete optimization problems written as integer linear programs.

Knapsack Problem

- Suppose we have a knapsack that can carry a maximum weight b and there are n types of items that we could take: an item of type i has weight $a_i > 0$.
- We want to load the knapsack with items without exceeding the knapsack capacity.
- On the other hand, suppose that an item of type i has value $c_i \geqslant 0$.
- The problem of loading the knapsack so as to maximize the value of all loaded items is the knapsack problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} c_i x_i, \\ \text{subject to} \quad & \sum_{i=1}^{n} a_i x_i \leqslant b, \\ & \mathrm{x} \in \mathbb{Z}^n, \mathrm{x} \geqslant 0. \end{aligned} \tag{4}$$

Knapsack Problem

- If only one item of each type is allowed to be loaded, then we can use binary variables instead of general integers.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} c_i x_i, \\ \text{subject to} \quad & \sum_{i=1}^{n} a_i x_i \leqslant b, \\ & x_i \in \{0,1\}, \forall i = \overline{1,n}. \end{aligned}$$

- In this way we get a Binary Integer Linear Programming (*BILP* or *BIP*) problem:

$$\begin{aligned} \text{maximize} \quad & z = \mathrm{c}^T \mathrm{x}, \\ \text{subject to} \quad & \mathrm{A}\mathrm{x} \leqslant \mathrm{b}, \\ & \mathrm{x} \in \{0,1\}^n. \end{aligned} \tag{5}$$

Set Packing Problem

- Suppose we have a finite set X, and a family of subsets $\mathcal{F} \subseteq 2^X$.
- A subfamily $\mathcal{F}' \subseteq \mathcal{F}$ is said to be a *packing* of $\mathcal{F}$ if $\mathcal{F}'$ contains only pairwise disjoint sets.
- The *Set Packing* problem is: given X, $\mathcal{F} \subseteq 2^X$, and $k \in \mathrm{N}$, there exists a packing of cardinality of least k?
- The optimization version of this problem is the Maximum Set Packing problem: what is the maximum number of pairwise disjoint sets in $\mathcal{F}$?

$$\begin{aligned} \text{maximize} \quad & \sum_{F \in \mathcal{F}} x_F, \\ \text{subject to} \quad & \sum_{F \in \mathcal{F}, x \in F} x_F \leqslant 1, \forall x \in X, \\ & x_F \in \{0, 1\}, \forall F \in \mathcal{F}. \end{aligned}$$

(The solution x is the characteristic vector of $\mathcal{F}'$.)

Set Cover Problem

- A subfamily $\mathcal{F}' \subseteq \mathcal{F}$ is said to be a *covering* of $\mathcal{F}$ if every element of X is covered by some set in $\mathcal{F}'$ (that is, $\bigcup F_{F \in \mathcal{F}'} = X$).
- The *Set Cover* problem is: given X, $\mathcal{F} \subseteq 2^X$, and $k \in \mathbb{N}$, there exists a covering of cardinality of most k?
- The optimization version of this problem is the Minimum Set Cover problem: what is the minimum cardinality covering of $\mathcal{F}$?

$$
\begin{aligned}
\text{minimize} \quad & \sum_{F \in \mathcal{F}} x_F, \\
\text{subject to} \quad & \sum_{F \in \mathcal{F}, x \in F} x_F \geqslant 1, \forall x \in X, \\
& x_F \in \{0, 1\}, \forall F \in \mathcal{F}.
\end{aligned}
$$

(The solution x is the characteristic vector of $\mathcal{F}'$.)

3-SAT Problem

- Suppose we have a finite set X of boolean variables, U the set of (positive or negative) literals over X, and $\mathcal{C}$ a formula in disjunctive normal form (that is, a disjunction of conjunctions of literals) where each clause is limited to at most three literals from U.
- The 3-*SAT* problem is: given X and $\mathcal{C}$ like above, there exists an assignment of truth on X that satisfies all the clauses in $\mathcal{C}$?
- The optimization version of this problem is the Max 3-SAT problem: what is the maximum number of satisfiable clauses in $\mathcal{C}$?
- In order to give the LP description of this problem we define $X = \{x_1, x_2, \ldots, x_k\}$, $u_i = x_i, v_i = \overline{x}_i$ for all $i = \overline{1,k}$, $L = \{u_i, v_i : i = \overline{1,k}\}$, and and $\mathcal{C} = \{C_1, C_2, \ldots, C_p\}$, where $C_j = w_1^j \vee w_2^j \vee w_3^j$, with $w_i^j \in L$.

3-SAT Problem

- The LP formulation of Max 3-SAT is

$$\begin{aligned} \text{maximize} \quad & \sum_{C \in \mathcal{C}} x_C, \\ \text{subject to} \quad & w_1^j + w_2^j + w_3^j - x_{C_j} \geqslant 0, \forall j = \overline{1,p}, \\ & u_i + v_i = 1, \forall i = \overline{1,k} \\ & x_C \in \{0,1\}, \forall C \in \mathcal{C}, \\ & u_i, v_i \in \{0,1\}, \forall i = \overline{1,k}. \end{aligned}$$

- In the corresponding truth assignment the boolean variable x_i is true iff $u_i = 1$.
- Constraint $u_i + v_i = 1$ was introduced to insure that x_i is true iff $\overline{x}_i$ is false.
- The construction of the linear program was made such that, for an optimal solution, $x_C = 1$ iff the clause C is satisfied by the current truth assignment.

Assignment Problem

- Suppose we have n people and n tasks; every pair person/task has a certain value c_{ij}.
- The assignment problem: allocate exactly one person to each task so that the total value is maximized.

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}, \\
\text{subject to} \quad & \sum_{i=1}^{n} x_{ij} = 1, \forall j = \overline{1,n} \\
& \sum_{j=1}^{n} x_{ij} = 1, \forall i = \overline{1,n} \\
& x_{ij} \in \{0,1\}, \forall i,j = \overline{1,n}.
\end{aligned}$$

- All the problems from above, except for the last, are NP-hard problems.
- This observation implies that a typical ILP problem is hard to solve.
- The assignment problem is also called *maximum weight bipartite matching problem* and can be solved in polynomial time complexity.

Integer Programming Models

- There are some classes of ILP problems which can be solved in polynomial time; this is possible for particular combinatorial problems.
- Generally we don't expect to find a polynomial time algorithm for solving an ILP problem, thus we are interested in finding broad methods to solve such a problem.
- We will study two approaches for solving ILP problems:
  - One approach is based on the particular structure of the problem (namely, particular properties of matrix A, such as *total unimodularity*);
  - The second approach is a more general one and is based on solving basic LP problems (using Simplex algorithm or other tools) and consists in several strategies: *branch-and-bound*, *cutting plane*, *branch-and cut methods* etc.

Totally Unimodular Matrices

Definition

An matrix A *is called* totally unimodular *if every square submatrix of* A *has its determinant in the set* $\{-1, 0, 1\}$.

- Obviously, a totally unimodular matrix has entries from $\{-1, 0, 1\}$.
- The following result underlines the importance of totally unimodular matrices in connection with ILP.

Theorem

Let A *be a totally unimodular* $m \times n$ *matrix, and* $\mathrm{b} \in \mathbb{Z}^m$. *Then all the extreme points of the polyhedron* $\mathcal{P} = \{\mathrm{x} \in \mathbb{R}^n \ : \ \mathrm{Ax} \leqslant \mathrm{b}\}$ *are integral vectors (i. e., from* $\mathbb{Z}^n$*).*

Totally Unimodular Matrices

Proof. Let x be an extreme point of $\mathcal{P}$ and $\mathrm{A_x}$ be the submatrix which contains only those rows a'_j, for which $\langle \mathrm{a}'_j, \mathrm{x}\rangle = b_j$.

Lemma

Matrix $\mathrm{A_x}$ *has rank* n. [a]

---

[a] We dropped the restriction concerning the full row rank of A.

Proof (for lemma). If $rank(\mathrm{A_x}) < n$, then the n columns of $\mathrm{A_x}$ are linearly dependent: there exists $\mathrm{y} \in \mathbb{R}^n, \mathrm{y} \neq 0$ with $\mathrm{A_x y} = 0$. Now, we can find an $\varepsilon > 0$ such that for every row a'_j which doesn't occur in $\mathrm{A_x}$ (i. e., $\mathrm{a}'_j\mathrm{x} < b_j$) we have

$$\langle \mathrm{a}'_j, \mathrm{x} + \varepsilon\mathrm{y}\rangle \leqslant b_j \text{ and } \langle \mathrm{a}'_j, \mathrm{x} - \varepsilon\mathrm{y}\rangle \leqslant b_j.$$

Since $\mathrm{A_x y} = 0$ and $\mathrm{Ax} \leqslant \mathrm{b}$, we get

$$\mathrm{A}(\mathrm{x} + \varepsilon\mathrm{y}) \leqslant \mathrm{b} \text{ and } \mathrm{A}(\mathrm{x} - \varepsilon\mathrm{y}) \leqslant \mathrm{b},$$

which means that $(\mathrm{x} + \varepsilon\mathrm{y}), (\mathrm{x} - \varepsilon\mathrm{y}) \in \mathcal{P}$ - a contradiction (why?). □

Totally Unimodular Matrices

Proof (cont'd for theorem). Let x be an extreme vector of $\mathcal{P}$ and A_{x} defined as above. Since $rank(\mathrm{A}_{\mathrm{x}}) = n$, there exist a square submatrix of A_{x}, A_1, of rank n. Let b_1 be a vector formed with elements of b that correspond to A_1; we must have $\mathrm{A}_1\mathrm{x} = \mathrm{b}_1$. Therefore, $\mathrm{x} = \mathrm{A}_1^{-1}\mathrm{b}_1$; but, since $det(\mathrm{A}_1) = \pm 1$, $\mathrm{A}_1^{-1} \in \mathbb{Z}^{n\times n}$, hence x is an integral vector. □

Proposition

Let (x, y) *be an extreme point of the polyhedron* $\mathcal{P} = \{\mathrm{x} \in \mathbb{R}^n, \mathrm{y} \in \mathbb{R}^m_+ \,:\, \mathrm{A}\mathrm{x} + \mathrm{y} = \mathrm{b}\}$, *then* x *is an extreme point of the polyhedron* $\mathcal{P}' = \{\mathrm{x} \in \mathbb{R}^n \,:\, \mathrm{A}\mathrm{x} \leqslant \mathrm{b}\}$.

Totally Unimodular Matrices

Proof. Let (x, y) be an extreme point of $\mathcal{P}$. Suppose, on the contrary, that there exist two points $\mathrm{x}^1 \neq \mathrm{x}^2$ in $\mathcal{P}'$ such that

$$\mathrm{x} = \frac{1}{2}\mathrm{x}^1 + \frac{1}{2}\mathrm{x}^2.$$

Let $i \in \{1, 2, \ldots, m\}$; we have three cases

(i) $y_i = 0$. We define $y_i^1 = y_i^2 = 0$.

(ii) $y_i > 0$, $\mathrm{a}_i'\mathrm{x}^1 = b_i$, and $\mathrm{a}_i'\mathrm{x}^2 < b_i$. We define $y_i^1 = 0$ and $y_i^2 = b_i - \mathrm{a}_i'\mathrm{x}^2$; since $\mathrm{a}_i'\mathrm{x}^1 + \mathrm{a}_i'\mathrm{x}^2 = 2\mathrm{a}_i^T\mathrm{x} = 2b_i - 2y_i$, we must have $y_i^2 = 2y_i$.

(iii) $y_i > 0$, $\mathrm{a}_i'\mathrm{x}^1 < b_i$, and $\mathrm{a}_i'\mathrm{x}^2 < b_i$. We define $y_i^1 = b_i - \mathrm{a}_i'\mathrm{x}^1$ and $y_i^2 = b_i - \mathrm{a}_i'\mathrm{x}^2$; since $\mathrm{a}_i'\mathrm{x}^1 + \mathrm{a}_i'\mathrm{x}^2 = 2\mathrm{a}_i'\mathrm{x} = 2b_i - 2y_i$, we must have $y_i^1 + y_i^2 = 2y_i$.

Totally Unimodular Matrices

In all of the above situations we have

$$\mathrm{a}_i^T \mathrm{x}^1 + y_i^1 = \mathrm{a}_i^T \mathrm{x}^2 + y_i^2 = b_i, \text{ and } \frac{1}{2}(x_i^1, y_i^1) + \frac{1}{2}(x_i^2, y_i^2) = (x_i, y_i).$$

Obviously $(\mathrm{x}^1, \mathrm{y}^1) \neq (\mathrm{x}^2, \mathrm{y}^2)$ are points from $\mathcal{P}$. Since

$$\frac{1}{2}(\mathrm{x}^1, \mathrm{y}^1) + \frac{1}{2}(\mathrm{x}^2, \mathrm{y}^2) = (\mathrm{x}, \mathrm{y}),$$

we come to the conclusion that (x, y) cannot be an extreme point in $\mathcal{P}$, which is a contradiction. □

Totally Unimodular Matrices

- An immediate consequence of Theorem 3.1 and Proposition 3.1 is the following: when we have to optimize over a polyhedron defined by a totally unimodular matrix under integral restrictions, we can use the Simplex algorithm.
- Suppose that we have to optimize over the polyhedron $\mathcal{P}'$. We add the slack variables y and optimize over the new polyhedron $\mathcal{P}$; using the Simplex algorithm we eventually find an optimal basic feasible solution $(x, y) \in \mathcal{P}$.
- We already know that such a solution corresponds to an extreme point (x, y) of $\mathcal{P}$. Therefore, from Proposition 3.1, x is an extreme point and, also, an optimal solution in $\mathcal{P}'$.
- From Theorem 3.1, x must be an integral vector. Hence, we found an optimal solution in $\mathcal{P}'$ which is an integral vector.

Totally Unimodular Matrices

Definition

A polyhedron *$\mathcal{P}$ is* integral *if, for every $\mathrm{c} \in \mathbb{R}^n$, for which $\sup\{\mathrm{c}^T\mathrm{x} : \mathrm{x} \in \mathcal{P}\} \in \mathbb{R}$, the supremum is attained at an integral vector.*

- A simple consequence is

Corollary

If $\mathcal{P} = \{\mathrm{x} \in \mathbb{R}^n : \mathrm{A}\mathrm{x} \leqslant \mathrm{b}\}$, A is an $m \times n$ totally unimodular matrix, and $\mathrm{b} \in \mathbb{Z}^m$, then $\mathcal{P}$ is an integral polyhedron.

Totally Unimodular Matrices

Proof. Let $c \in \mathbb{R}^n$ and x_* an optimal solution to the problem

$$\max\{c^T x : x \in \mathcal{P}\}.$$

x_* can be chosen as an extreme point of $\mathcal{P}$ and by Theorem 3.1 such a point is integral.

Corollary

If A *is an* $m \times n$ *totally unimodular matrix,* $b \in \mathbb{Z}^m$*, and* $c \in \mathbb{Z}^n$*, then the following primal/dual pair of problems have integral optimum solutions (if their optima are finite)*

$$\max\{c^T x : Ax \leqslant b\} = \min\{b^T y : y \geqslant 0, A^T y = c\}$$

Proof. Using Corollary 3.1 and the (easy to prove) fact that the following matrix is also totally unimodular

$$A' = \begin{bmatrix} -I_m \\ A^T \\ -A^T \end{bmatrix}. \square$$

- It can be proved that the property listed in Corollary 3.1 is a characterization of total unimodularity.

Theorem

(Hoffman-Kruskal Theorem) Let A *be an integral* $m \times n$ *matrix. Then* A *is totally unimodular iff, for each* $b \in \mathbb{Z}^m$*, the polyhedron* $\mathcal{P} = \{x \in \mathbb{R}^n_+ \ : \ Ax \leqslant b\}$ *is integral.*

Totally Unimodular Matrices from Bipartite Graphs

- Let $G = (V, E)$ be a graph, with $V = \{v_1, v_2, \ldots, v_m\}$ and $E = \{e_1, e_2, \ldots, e_n\}$.
- The *incidence matrix* of G is $A \in \{0, 1\}^{m \times n}$, such that $a_{ij} = 1$ iff vertex v_i is adjacent with edge e_j.
- The incidence matrix can or cannot be totally unimodular, what we can prove is

Theorem

A graph G is bipartite iff its incidence matrix is totally unimodular.

Totally Unimodular Matrices from Bipartite Graphs

Proof.

"$\Longrightarrow$" Suppose A' is a square submatrix of A, of order k - we prove that $det(\mathrm{A}') \in \{-1, 0, 1\}$ by induction on $k \geqslant 2$ (for $k = 1$ the result obviously holds). We have three possible situations

(i) A' has a null column, then $det(\mathrm{A}') = 0$.

(ii) A' has a column with exactly one 1. Then, using the Laplace expansion, $det(\mathrm{A}') = \pm 1 \cdot det(\mathrm{A}'')$, where A'' is a square submatrix of A' (hence, of A) of order $(k-1)$. Therefore $det(\mathrm{A}'') \in \{-1, 0, 1\}$ and $det(\mathrm{A}') \in \{-1, 0, 1\}$.

(iii) Every column of A' has exactly two 1's. A' is the incidence matrix of a subgraph G' of G. Since G' is also bipartite, with bipartition (V_1', V_2'), if we add the rows corresponding to vertices from V_1' we get a row full of 1's, and the same result if we add the rows corresponding to vertices from V_2'. Therefore $det(A') = 0$.

Totally Unimodular Matrices from Bipartite Graphs

"$\Longleftarrow$" We will use the well-known characterization: a graph is bipartite iff it doesn't contain odd cycles. If G is not bipartite, then it contains an odd cycle through vertices $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$, and edges $e_{j_1}, e_{j_2}, \ldots, e_{j_k}$ (in this order).

The submatrix of A having rows $i_1, i_2, \ldots, i_k$ and columns $j_1, j_2, \ldots, j_k$ has the determinant equal with 2 (check!) - which is a contradiction.

□

- Using the last result and the Corollary 3.2, we can obtain some combinatorial results concerning bipartite graphs.
- One of these results links minimum cardinality edge covers to maximum cardinality stable sets in G; another one relates maximum cardinality matchings to minimum vertex covers (both König's theorems).
- These results are generalized in the following theorem.

Totally Unimodular Matrices from Bipartite Graphs

Theorem

Let G be bipartite graph and $w : E(G) \to \mathbb{N}$ be a weight function defined on its edges. Then

(i) *The maximum weight of a matching in G is equal to the minimum value of $\sum_{v \in V(G)} f(v)$, where f ranges over all functions $f : V(G) \to \mathbb{N}$, such that $f(u) + f(v) \geqslant w(uv), \forall uv \in E(G)$.*

(ii) *The minimum weight of an edge cover in G is equal to the maximum value of $\sum_{v \in V(G)} f(v)$, where f ranges over all functions $f : V(G) \to \mathbb{N}$, such that $f(u) + f(v) \leqslant w(uv), \forall uv \in E(G)$.*

Totally Unimodular Matrices from Bipartite Graphs

Proof. The properties are equivalent with

$$\max\{w^T x : Ax \leqslant 1, x \geqslant 0\} = \min\{1^T y : A^T y \geqslant w, y \geqslant 0\},$$

$$\min\{w^T x : Ax \geqslant 1, x \geqslant 0\} = \max\{1^T y : A^T y \leqslant w, y \geqslant 0\},$$

respectively, where A is the incidence matrix of G. Now, using Theorem 3.3 and Corollary 3.2 we conclude that both these relations are true. ☐

Totally Unimodular Matrices from Digraphs

- Let $D = (V, A)$ be a directed graph with $V = \{v_1, v_2, \ldots, v_m\}$ and $A = \{e_1, e_2, \ldots, e_n\}$. Let us define the incidence matrix of D, the $m \times n$ matrix M with entries

$$m_{ij} = \begin{cases} 1, & \text{if } e_j \text{ leaves } v_i \\ -1, & \text{if } e_j \text{ enters } v_i \\ 0, & \text{otherwise} \end{cases}$$

Theorem

The incidence matrix of a digraph is totally unimodular.

Proof. Let M′ be a square submatrix of M of order k. We proceed by induction on k. Suppose $k > 2$; we have three possible situation:

Totally Unimodular Matrices from Digraphs

(i) M' has a null column, then $det(M') = 0$.

(ii) M' has a column with exactly one non-null value. Then, using the Laplace expansion, $det(M') = \pm 1 \cdot det(M'')$, where M'' is a square submatrix of M' (hence, of M) of order $(k-1)$. $det(M'') \in \{-1, 0, 1\}$ by our induction hypothesis; therefore, $det(A') \in \{-1, 0, 1\}$.

(iii) Every column of M' has exactly two nonzero values (an 1 and a -1). By adding up all the rows of M' we get a null row, hence $det(M') = 0$. □

- The incidence matrix of a digraph relates with flows and circulations in D, because the homogeneous system of linear equations $Mx = 0$ is equivalent with

$$\sum_{e_j \in \delta^+(v)} x_j = \sum_{e_j \in \delta^-(v)} x_j, \forall v \in V$$

which is the *conservation law*.

Totally Unimodular Matrices from Digraphs

Definition

Let $D = (V, A)$ *be a digraph; a circulation of* D *is a function* $\mathrm{x} : A \to \mathbb{R}$ *which obeys the conservation law.*

Corollary

Let $D = (V, A)$ *be a digraph and* $\mathrm{c}_1, \mathrm{c}_2 : A \to \mathbb{Z}$*. There exists a circulation* x *of* D *with* $\mathrm{c}_1 \leqslant \mathrm{x} \leqslant \mathrm{c}_2$ *iff there exists an integral circulation* x *of* D *with* $\mathrm{c}_1 \leqslant \mathrm{x} \leqslant \mathrm{c}_2$*.*

Proof. Obviously we need to prove just the only if part.

Totally Unimodular Matrices from Digraphs

If there exists a circulation x of D with $c_1 \leqslant x \leqslant c_2$, then the following polytope (that is, a bounded polyhedron) is nonempty

$$\mathcal{P} = \{x \in \mathbb{R}^n : Mx = 0, c_1 \leqslant x \leqslant c_2\}.$$

Any extreme point, x_*, of $\mathcal{P}$ is an integral circulation of D, with $c_1 \leqslant x_* \leqslant c_2$. $\square$

- Another interesting application is the *max-flow min-cut theorem*:

Corollary

Let $D = (V, A)$ *be a digraph,* $s \neq t \in V$*, and* $c : A \to \mathbb{R}_+$ *be a capacity function. The maximum value of an st-flow is equal with the minimum capacity of an st-cut.*

Totally Unimodular Matrices from Digraphs

Proof. It is obvious that the value of any st-flow cannot exceed the capacity of any st-cut. It suffices to prove that there exist a flow x and cut (S, T) such that $v(\mathrm{x}) \geqslant \mathrm{c}(S, T)$.

Let M be the vertex-arc incidence matrix of G. If we delete from M the rows corresponding to s and t, we get a matrix M'. The flow conservation law is equivalent with $\mathrm{M}'\mathrm{x} = 0$. Now, if a_{i_0} is the row corresponding to s, then $\mathrm{a}_{i_0}\mathrm{x}$ is the value of the flow x.

The maximum value of a st-flow is

$$\max\{\mathrm{a}_{i_0}^T\mathrm{x} : 0 \leqslant \mathrm{x} \leqslant \mathrm{c}, \mathrm{M}'\mathrm{x} = 0\}$$

The dual problem is

$$\min\{\mathrm{c}^T\mathrm{y}'' : \mathrm{y}'' \geqslant 0, \mathrm{M}'^T\mathrm{y}' + \mathrm{y}'' \geqslant \mathrm{a}_{i_0}\},$$

Totally Unimodular Matrices from Digraphs

or

$$\min\{\mathrm{c}^T\mathrm{y}'' : \mathrm{y}'' \geqslant 0, \mathrm{M}_0^T(\mathrm{y}'^T\ \mathrm{y}''^T)^T \geqslant \mathrm{a}_0\},$$

where

$$\mathrm{M}_0 = \begin{bmatrix} \mathrm{M}' \\ I_n \end{bmatrix}, \mathrm{a}_0 = \begin{bmatrix} \mathrm{a}'^T_{i_0} \\ 0 \end{bmatrix}.$$

Obviously, M_0 is a totally unimodular matrix (why?). Since a_{i_0} is an integral vector, the optimal solution of the dual $(\mathrm{y}', \mathrm{y}'')$ is an integral vector. We can build a *st*-cut like this:

$$S = \{v \in V : v \neq s, t, y''_v \leqslant -1\} \cup \{s\}, T = V \setminus S.$$

The proof is completed by observing that $v(\mathrm{x}) \leqslant \mathrm{c}(S, T)$ (exercise). □

Bibliography

- Bertsimas, D., J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, Massachusetts, 1997.
- Chekuri, C., *Topics in Combinatorial Optimization*, University of Illinois Urbana-Champaign, Lecture Notes: `https://courses.engr.illinois.edu/cs598csc/sp2010/`, 2010.
- Conforti, M., G. Cornuejols, G. Zambelli, *Integer Programming*, Graduate Texts in Mathematics, Springer, 2014.
- Griva, I., S. G. Nash, A. Sofer, *Linear and Nonlinear Optimization*, 2nd edition, SIAM, 2009.
- Schrijver, A., *A Course in Combinatorial Optimization*, Electronic Edition: `homepages.cwi.nl/~lex/files/dict.pdf`, 2013.
- Vanderbei, R., J., *Linear Programming - Foundations and Extensions*, International Series in Operations Research & Management Science, Springer Science, 4th edition, 2014.