Design and Analysis of Algorithms (XIII)

Linear Programming: Applications

Guoqiang Li
School of Software

SHANGHAI JIAO TONG UNIVERSITY

Shortest Path

Shortest Path

Shortest path problem gives a weighted, directed graph $G = (V, E)$, with weight function $w : E \to \mathbb{Q}^+$ mapping edges to real-valued weights, a source vertex s, and destination vertex t. We wish to compute the weight of a shortest path from s to t.

Shortest Path in LP

$$\max d_t$$

$$\begin{aligned} d_v &\le d_u + w(u,v) && (u,v) \in E \\ d_s &= 0 && \\ d_i &\ge 0 && i \in V \end{aligned}$$

Shortest Path in LP

$$\max d_t$$

$$\begin{aligned} d_v &\le d_u + w(u,v) && (u,v) \in E \\ d_s &= 0 \\ d_i &\ge 0 && i \in V \end{aligned}$$

Q: Another formalization?

Shortest Path in LP

Shortest Path in LP

Let $\mathcal{S} = \{S \subseteq V : s \in S, t \notin S\}$; that is, $\mathcal{S}$ is the set of all s-t cuts in the graph. Then we can model the shortest s-t path problem with the following integer program,

$$\min \sum_{e \in E} w_e x_e$$

$$\begin{aligned} \sum_{e \in \delta(S)} x_e \geq 1 \quad & S \in \mathcal{S} \\ x_e \in \{0, 1\} \quad & e \in E \end{aligned}$$

where $\delta(S)$ is the set of all edges that have one endpoint in S and the other endpoint not in S.

Shortest Path in LP

Let $\mathcal{S} = \{S \subseteq V : s \in S, t \notin S\}$; that is, $\mathcal{S}$ is the set of all s-t cuts in the graph. Then we can model the shortest s-t path problem with the following integer program,

$$\min \sum_{e \in E} w_e x_e$$

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad S \in \mathcal{S}$$
$$x_e \in \{0,1\} \quad e \in E$$

where $\delta(S)$ is the set of all edges that have one endpoint in S and the other endpoint not in S.

- Can we relax the restriction $x_e \in \{0,1\}$ to $0 \leq x_e \leq 1$?

Shortest Path in LP

Let $\mathcal{S} = \{S \subseteq V : s \in S, t \notin S\}$; that is, $\mathcal{S}$ is the set of all s-t cuts in the graph. Then we can model the shortest s-t path problem with the following integer program,

$$\min \sum_{e \in E} w_e x_e$$

$$\begin{aligned} \sum_{e \in \delta(S)} x_e \geq 1 \quad & S \in \mathcal{S} \\ x_e \in \{0,1\} \quad & e \in E \end{aligned}$$

where $\delta(S)$ is the set of all edges that have one endpoint in S and the other endpoint not in S.

- Can we relax the restriction $x_e \in \{0,1\}$ to $0 \leq x_e \leq 1$?
- How about $x_e \geq 0$?

Shortest Path in LP

$$\min \sum_{e \in E} w_e x_e$$

$$\sum_{e \in \delta(S)} x_e \geq 1 \qquad S \in \mathcal{S}$$

$$x_e \geq 0 \qquad e \in E$$

Shortest Path in LP

$$\min \sum_{e \in E} w_e x_e$$

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad S \in \mathcal{S}$$

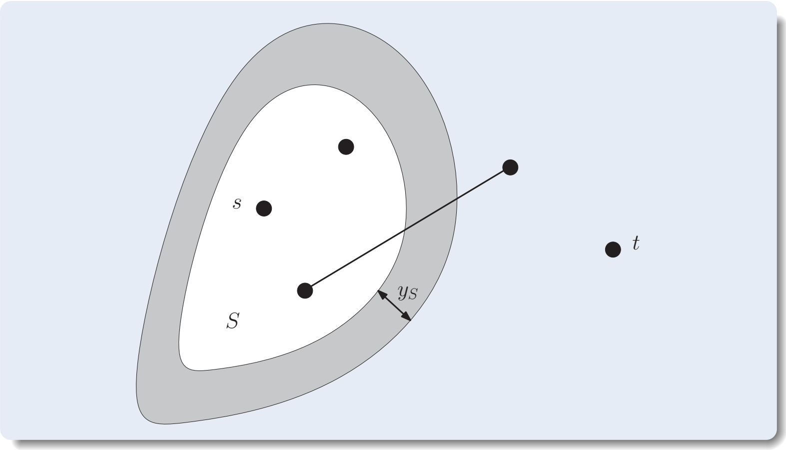
$$x_e \geq 0 \quad e \in E$$

$$\max \sum_{S \in \mathcal{S}} y_S$$

$$\sum_{S \in \mathcal{S}, e \in \delta(S)} y_S \leq w_e \quad e \in E$$

$$y_S \geq 0 \quad S \in \mathcal{S}$$

The Moat



Max-Flow Min-Cut in LP

Shipping Oil

SHANGHAI JIAO TONG UNIVERSITY

We have a network of pipelines along which oil can be sent.

Shipping Oil

SHANGHAI JIAO TONG UNIVERSITY

We have a network of pipelines along which oil can be sent.

The goal is to ship as much oil as possible from the source to the sink.

Shipping Oil

We have a network of pipelines along which oil can be sent.

The goal is to ship as much oil as possible from the source to the sink.

Each pipeline has a maximum capacity it can handle,

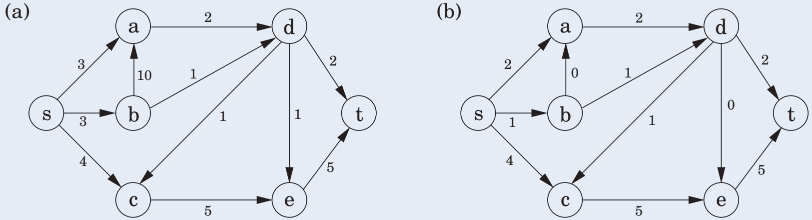
Shipping Oil

We have a network of pipelines along which oil can be sent.

The goal is to ship as much oil as possible from the source to the sink.

Each pipeline has a maximum capacity it can handle, and there are no opportunities for storing oil en route.

A Flow Example



Maximizing Flow

The networks consist of a directed graph $G = (V, E)$;

Maximizing Flow

SHANGHAI JIAO TONG UNIVERSITY

The networks consist of a directed graph $G = (V, E)$; two special nodes $s, t \in V$, a source and sink of G;

Maximizing Flow

The networks consist of a directed graph $G = (V, E)$; two special nodes $s, t \in V$, a source and sink of G; and capacities $c_e > 0$ on the edges.

Maximizing Flow

The networks consist of a directed graph $G = (V, E)$; two special nodes $s, t \in V$, a source and sink of G; and capacities $c_e > 0$ on the edges.

Aim to send as much oil as possible from s to t without exceeding the capacities of any of the edges.

Maximizing Flow

A flow consists of a variable f_e for each edge e of the network, satisfying the following two properties:

Maximizing Flow

A flow consists of a variable f_e for each edge e of the network, satisfying the following two properties:

1. It doesn't violate edge capacities: $0 \leq f_e \leq c_e$ for all $e \in E$.

Maximizing Flow

A flow consists of a variable f_e for each edge e of the network, satisfying the following two properties:

1. It doesn't violate edge capacities: $0 \le f_e \le c_e$ for all $e \in E$.
2. For all nodes u except s and t, the amount of flow entering u equals the amount leaving

$$\sum_{(w,v)\in E} f_{wu} = \sum_{(u,z)\in E} f_{uz}$$

In other words, flow is conserved.

Maximizing Flow

The value of a flow is the total quantity sent from s to t

Maximizing Flow

SHANGHAI JIAO TONG UNIVERSITY

The value of a flow is the total quantity sent from s to t and, by the conservation principle, is equal to the quantity leaving s:

$$\mathtt{val}(f) = \sum_{(s,u)\in E} f_{su}$$

Maximizing Flow

The value of a flow is the total quantity sent from s to t and, by the conservation principle, is equal to the quantity leaving s:

$$\texttt{val}(f) = \sum_{(s,u)\in E} f_{su}$$

Our goal is to assign values to $\{f_e | e \in E\}$ that will satisfy a set of linear constraints and maximize a linear objective function.

Maximizing Flow

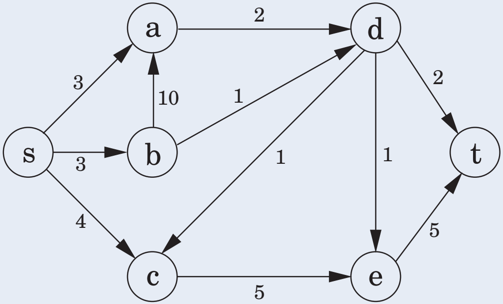
The value of a flow is the total quantity sent from s to t and, by the conservation principle, is equal to the quantity leaving s:

$$\texttt{val}(f) = \sum_{(s,u)\in E} f_{su}$$

Our goal is to assign values to $\{f_e | e \in E\}$ that will satisfy a set of linear constraints and maximize a linear objective function.

This is a linear program. The maximum-flow problem reduces to linear programming.

The Example



LP

LP

11 variables, one per edge.

LP

11 variables, one per edge.

maximize $f_{sa} + f_{sb} + f_{sc}$

LP

11 variables, one per edge.

maximize $f_{sa} + f_{sb} + f_{sc}$

27 constraints:

- 11 for nonnegativity (such as $f_{sa} \geq 0$),

LP

11 variables, one per edge.

maximize $f_{sa} + f_{sb} + f_{sc}$

27 constraints:

- 11 for nonnegativity (such as $f_{sa} \geq 0$),
- 11 for capacity (such as $f_{sa} \leq 3$),

11 variables, one per edge.

maximize $f_{sa} + f_{sb} + f_{sc}$

27 constraints:

- 11 for nonnegativity (such as $f_{sa} \geq 0$),
- 11 for capacity (such as $f_{sa} \leq 3$),
- 5 for flow conservation (one for each node of the graph other than s and t, such as $f_{sc} + f_{dc} = f_{ce}$).

Another Representation

First, introduce a fictitious edge of infinite capacity from t to s thus converting the flow to a circulation;

Another Representation

First, introduce a fictitious edge of infinite capacity from t to s thus converting the flow to a circulation;

The objective is to maximize the flow on this edge, denoted by f_{ts}.

Another Representation

First, introduce a fictitious edge of infinite capacity from t to s thus converting the flow to a circulation;

The objective is to maximize the flow on this edge, denoted by f_{ts}.

The advantage of making this modification is that we can now require flow conservation at s and t as well.

Another Representation

$$
\begin{aligned}
& \max f_{ts} \\
& f_{ij} \leq c_{ij} && (i,j) \in E \\
& \sum_{(w,i)\in E} f_{wi} - \sum_{(i,z)\in E} f_{iz} \leq 0 && i \in V \\
& f_{ij} \geq 0 && (i,j) \in E
\end{aligned}
$$

Remark

Ford-Fulkersons algorithm can be regarded as a special algorithm of linear programs.

Min-Max Relations and Duality

LP for Max Flow

$$\max f_{ts}$$

$$f_{ij} \leq c_{ij} \qquad (i,j) \in E$$

$$\sum_{(w,i)\in E} f_{wi} - \sum_{(i,z)\in E} f_{iz} \leq 0 \qquad i \in V$$

$$f_{ij} \geq 0 \qquad (i,j) \in E$$

LP-Duality

$$\max f_{ts}$$

$$f_{ij} \leq c_{ij} \qquad (i,j) \in E$$

$$\sum_{(w,i)\in E} f_{wi} - \sum_{(i,z)\in E} f_{iz} \leq 0 \qquad i \in V$$

$$f_{ij} \geq 0 \qquad (i,j) \in E$$

LP-Duality

$$
\begin{aligned}
&\max f_{ts} \\
&f_{ij} \le c_{ij} && (i,j) \in E \\
&\sum_{(w,i)\in E} f_{wi} - \sum_{(i,z)\in E} f_{iz} \le 0 && i \in V \\
&f_{ij} \ge 0 && (i,j) \in E
\end{aligned}
$$

$$
\begin{aligned}
&\min \sum_{(i,j)\in E} c_{ij} d_{ij} \\
&d_{ij} - p_i + p_j \ge 0 && (i,j) \in E \\
&p_s - p_t \ge 1 \\
&d_{ij} \ge 0 && (i,j) \in E \\
&p_i \ge 0 && i \in V
\end{aligned}
$$

Explanation of the Dual

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 && (i,j) \in E \\ p_s - p_t &\geq 1 && \\ d_{ij} &\in \{0,1\} && (i,j) \in E \\ p_i &\in \{0,1\} && i \in V \end{aligned}$$

Explanation of the Dual

SHANGHAI JIAO TONG UNIVERSITY

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 && (i,j) \in E \\ p_s - p_t &\geq 1 && \\ d_{ij} &\in \{0,1\} && (i,j) \in E \\ p_i &\in \{0,1\} && i \in V \end{aligned}$$

To obtain the dual program we introduce variables d_{ij} and p_i corresponding to the two types of inequalities in the primal.

- d_{ij}: distance labels on edges;

Explanation of the Dual

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 && (i,j) \in E \\ p_s - p_t &\geq 1 && \\ d_{ij} &\in \{0,1\} && (i,j) \in E \\ p_i &\in \{0,1\} && i \in V \end{aligned}$$

To obtain the dual program we introduce variables d_{ij} and p_i corresponding to the two types of inequalities in the primal.

- d_{ij}: distance labels on edges;
- p_i: potentials on nodes.

Integer Program

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 & (i,j) \in E \\ p_s - p_t &\geq 1 & \\ d_{ij} &\in \{0,1\} & (i,j) \in E \\ p_i &\in \{0,1\} & i \in V \end{aligned}$$

Integer Program

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 && (i,j) \in E \\ p_s - p_t &\geq 1 && \\ d_{ij} &\in \{0,1\} && (i,j) \in E \\ p_i &\in \{0,1\} && i \in V \end{aligned}$$

Let $(\mathbf{d}^*, \mathbf{p}^*)$ be an optimal solution to this integer program.

Integer Program

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 && (i,j) \in E \\ p_s - p_t &\geq 1 && \\ d_{ij} &\in \{0,1\} && (i,j) \in E \\ p_i &\in \{0,1\} && i \in V \end{aligned}$$

Let $(\mathbf{d}^*, \mathbf{p}^*)$ be an optimal solution to this integer program.

The only way to satisfy the inequality $p_s^* - p_t^* \geq 1$ with a $0/1$ substitution is to set $p_s^* = 1$ and $p_t^* = 0$.

Integer Program

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 & (i,j) \in E \\ p_s - p_t &\geq 1 & \\ d_{ij} &\in \{0,1\} & (i,j) \in E \\ p_i &\in \{0,1\} & i \in V \end{aligned}$$

Let $(\mathbf{d}^*, \mathbf{p}^*)$ be an optimal solution to this integer program.

The only way to satisfy the inequality $p_s^* - p_t^* \geq 1$ with a $0/1$ substitution is to set $p_s^* = 1$ and $p_t^* = 0$.

This solution defines an $s-t$ cut $(X, \overline{X})$, where X is the set of potential 1 nodes, and $\overline{X}$ the set of potential 0 nodes.

Integer Program

SHANGHAI JIAO TONG UNIVERSITY

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 && (i,j) \in E \\ p_s - p_t &\geq 1 \\ d_{ij} &\in \{0,1\} && (i,j) \in E \\ p_i &\in \{0,1\} && i \in V \end{aligned}$$

Integer Program

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j &\geq 0 \quad & (i,j) \in E \\ p_s - p_t &\geq 1 & \\ d_{ij} &\in \{0,1\} & (i,j) \in E \\ p_i &\in \{0,1\} & i \in V \end{aligned}$$

Consider an edge (i,j) with $i \in X$ and $j \in \overline{X}$, Since $p_i^* = 1$ and $p_j^* = 0$, and thus $d_{ij}^* = 1$.

Integer Program

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned} d_{ij} - p_i + p_j \geq 0 \quad & (i,j) \in E \\ p_s - p_t \geq 1 \quad & \\ d_{ij} \in \{0,1\} \quad & (i,j) \in E \\ p_i \in \{0,1\} \quad & i \in V \end{aligned}$$

Consider an edge (i, j) with $i \in X$ and $j \in \overline{X}$, Since $p_i^* = 1$ and $p_j^* = 0$, and thus $d_{ij}^* = 1$.

The distance label for each of the remaining edges can be set to either 0 or 1 without violating the first constraints.

Integer Program

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

$$\begin{aligned}
d_{ij} - p_i + p_j \geq 0 \quad & (i,j) \in E \\
p_s - p_t \geq 1 \quad & \\
d_{ij} \in \{0,1\} \quad & (i,j) \in E \\
p_i \in \{0,1\} \quad & i \in V
\end{aligned}$$

Consider an edge (i, j) with $i \in X$ and $j \in \overline{X}$, Since $p_i^* = 1$ and $p_j^* = 0$, and thus $d_{ij}^* = 1$.

The distance label for each of the remaining edges can be set to either 0 or 1 without violating the first constraints.

The objective function value is precisely the capacity of the cut $(X, \overline{X})$, and hence $(X, \overline{X})$ must be a minimum $s - t$ cut.

Relaxation of the Integer Program

The integer program is a formulation of the minimum $s - t$ cut problem.

Relaxation of the Integer Program

The integer program is a formulation of the minimum $s-t$ cut problem.

The dual program can be viewed as a relaxation of the integer program where the integrality constraint on the variables is dropped.

Relaxation of the Integer Program

The integer program is a formulation of the minimum $s - t$ cut problem.

The dual program can be viewed as a relaxation of the integer program where the integrality constraint on the variables is dropped.

This leads to the constraints $1 \geq d_{ij} \geq 0$ for $(i, j) \in E$ and $1 \geq p_i \geq 0$ for $i \in V$.

Relaxation of the Integer Program

The integer program is a formulation of the minimum $s-t$ cut problem.

The dual program can be viewed as a relaxation of the integer program where the integrality constraint on the variables is dropped.

This leads to the constraints $1 \geq d_{ij} \geq 0$ for $(i,j) \in E$ and $1 \geq p_i \geq 0$ for $i \in V$.

The upper bound constraints on the variables are redundant; their omission cannot give a better solution.

Relaxation of the Integer Program

SHANGHAI JIAO TONG UNIVERSITY

The integer program is a formulation of the minimum $s - t$ cut problem.

The dual program can be viewed as a relaxation of the integer program where the integrality constraint on the variables is dropped.

This leads to the constraints $1 \geq d_{ij} \geq 0$ for $(i, j) \in E$ and $1 \geq p_i \geq 0$ for $i \in V$.

The upper bound constraints on the variables are redundant; their omission cannot give a better solution.

We will say that this program is the LP relaxation of the integer program.

Relaxation of the Integer Program

SHANGHAI JIAO TONG UNIVERSITY

The best fractional $s - t$ cut could have lower capacity than the best integral cut. This does not happen here.

Relaxation of the Integer Program

The best fractional $s - t$ cut could have lower capacity than the best integral cut. This does not happen here.

Now, it can be proven that each vertex solution is integral, with each coordinate being 0 or 1.

Relaxation of the Integer Program

SHANGHAI JIAO TONG UNIVERSITY

The best fractional $s - t$ cut could have lower capacity than the best integral cut. This does not happen here.

Now, it can be proven that each vertex solution is integral, with each coordinate being 0 or 1.

The constraint matrix of this program is totally unimodular, Thus, the dual program always has an integral optimal solution.

More Examples

Set Cover

Set Cover

- Input: A set of elements U, sets $S_1, \ldots, S_m \subseteq U$
- Output: A selection of the S_i whose union is U.
- Cost: Number of sets picked.

Set Cover

Set Cover

$$
\begin{aligned}
\min \quad & \sum_{S \in \mathcal{S}} x_S & \\
& \sum_{S: e \in S} x_S \geq 1, & e \in U \\
& x_S \geq 0, & S \in \mathcal{S}
\end{aligned}
$$

Set Cover

$$
\begin{aligned}
\min \quad & \sum_{S \in \mathcal{S}} x_S & \\
& \sum_{S : e \in S} x_S \geq 1, & e \in U \\
& x_S \geq 0, & S \in \mathcal{S}
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & \sum_{e \in U} y_e & \\
& \sum_{e : e \in S} y_e \leq 1, & S \in \mathcal{S} \\
& y_e \geq 0, & e \in U
\end{aligned}
$$

Set Cover

- Input: A set of elements U, sets $S_1, \dots, S_m \subseteq U$, and a cost function $c : \mathcal{S} \to \mathbb{Q}^+$.
- Output: A selection of the S_i whose union is U.
- Cost: Sum of costs of set picked.

The special case, in which all subsets are of unit cost, will be called the cardinality set cover problem.

Set Cover

SHANGHAI JIAO TONG UNIVERSITY

$$
\begin{aligned}
\min \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\
& \sum_{S : e \in S} x_S \geq 1, && e \in U \\
& x_S \geq 0, && S \in \mathcal{S}
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & \sum_{e \in U} y_e \\
& \sum_{e : e \in S} y_e \leq c(S), && S \in \mathcal{S} \\
& y_e \geq 0, && e \in U
\end{aligned}
$$

Quiz: Set Multicover

Each element, e, needs to be covered a specified integer number, r_e, of times.

The objective again is to cover all elements up to their coverage requirements at minimum cost.

Each set can be picked at most once.

Integer Program

Let $r_e \in \mathbb{Z}_+$ be the coverage requirement for each element $e \in U$.

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\ & \sum_{S : e \in S} x_S \geq r_e, && e \in U \\ & x_S \in \{0, 1\}, && S \in \mathcal{S} \end{aligned}$$

Linear Program Relaxation

In the LP-relaxation, the constraints $x_S \leq 1$ are no longer redundant.

$$
\begin{aligned}
\min \quad & \sum_{S \in \mathcal{S}} c(S) x_S & \\
& \sum_{S : e \in S} x_S \geq r_e, & e \in U \\
& -x_S \geq -1, & S \in \mathcal{S} \\
& x_S \geq 0, & S \in \mathcal{S}
\end{aligned}
$$

Dual Program

SHANGHAI JIAO TONG UNIVERSITY

The additional constraints in the primal lead to new variables, z_S, in the dual.

$$
\begin{aligned}
\max \quad & \sum_{e \in U} r_e y_e - \sum_{S \in \mathcal{S}} z_S & \\
& \Big(\sum_{e: e \in S} y_e\Big) - z_S \leq c(S), & S \in \mathcal{S} \\
& y_e \geq 0, & e \in U \\
& z_S \geq 0, & S \in \mathcal{S}
\end{aligned}
$$

Referred Materials

Referred Materials

Shanghai Jiao Tong University

Content of this lecture comes from Section 12.2 in [Vaz04] and Section 7.3 in [WS11].

Suggest to read Section 26.1 and 26.2 in [CLRS09].