s-t shortest path

Dijkstra’s Shortest Path as Primal-Dual

Steiner Forest Problem
**s-t Shortest Path**

**Problem**
We are given an undirected graph \( G = (V, E) \), nonnegative edge costs \( c_e \geq 0 \) on all edges \( e \in E \), and a pair of distinguished vertices \( s \) and \( t \). The goal is to find the minimum cost path from \( s \) to \( t \).

**RECAP: Primal-Dual**

\[
\begin{align*}
\min \sum_{e} c_e x_e & \quad \text{(Primal-LP1)} \\
\text{subject to} \quad \sum_{e \in \delta(S)} x_e \geq 1 & \quad \forall S \subseteq V, s \in S, t \notin S \\
\quad x_e \geq 0 & \quad \forall e \in E
\end{align*}
\]
**s-t Shortest Path**

**Problem**

We are given an undirected graph $G = (V, E)$, nonnegative edge costs $c_e \geq 0$ on all edges $e \in E$, and a pair of distinguished vertices $s$ and $t$. The goal is to find the minimum cost path from $s$ to $t$.

**RECAP: Primal-Dual**

\[
\min \sum_e y_S \\
\text{subject to} \\
\sum_{S: e \in \delta(S)} y_S \leq c_e \quad \forall e \in E \\
y_S \geq 0 \quad \forall S \subseteq V, s \in S, t \not\in S
\]
**s-t Shortest Path**

**Algorithm.**

1. Set $y = 0$
2. Set $x = 0$, $\mathcal{F} = \phi$
3. While $s$ and $t$ is not connected
   3.1 Pick the minimally connected component containing $s$, call it $C$
   3.2 Raise the dual variable $y_C$ until a dual constraint becomes tight.
   3.3 Pick the edge $e$ corresponding to tight dual constraint in $\mathcal{F}$.
4. Let $P$ be the final solution obtained from $\mathcal{F}$ by reverse-deleting edges that are not required to maintain a $s$-$t$ path. Set $x_e$ corresponding to edges in $P$ to 1.
s-t Shortest Path

Analysis.

\[ \text{cost}(P) = \sum_{e \in P} c_e \]

\[ = \sum_{e \in P} \sum_{S: e \in \delta(S)} y_S \]

\[ = \sum_{S} y_S |\delta(S) \cap P| \]

- We showed after any iteration \( \mathcal{F} \) is a tree.
- \( |\delta(S) \cap P| \geq 2 \) implies a cycle in \( \mathcal{F} \).
- Hence \( \text{cost}(P) \leq \sum_S y_S \leq \text{OPT} \)
$s$-$t$ Shortest Path

Analysis.

**Primal Complementary Slackness Condition.**
For all $e \in E$ either $x_e = 0$ or $\sum_{S : e \in \delta(S)} y_S = c_e$ holds.

**Dual Complementary Slackness Condition.**
For all $S \subseteq V$, $s \in S$, $t \not\in S$ either $y_S = 0$, or $\sum_{e : e \in \delta(S)} x_e = 1$ holds.

- If $y_S > 0$ then $\sum_{e : e \in \delta(S)} x_e = |\delta(S) \cap P| = 1$
Approximate Complementary Slackness Condition.

**Primal**

\[
\begin{align*}
\text{min} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{i,j} x_j \geq b_i, \quad i = 1, 2, \ldots, m \\
& \quad x_j \geq 0, \quad j = 1, 2, \ldots, n
\end{align*}
\]

**Dual**

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{m} b_i y_i \\
\text{subject to} & \quad \sum_{i=1}^{m} a_{i,j} y_i \leq c_j, \quad j = 1, 2, \ldots, n \\
& \quad y_i \geq 0, \quad i = 1, 2, \ldots, m
\end{align*}
\]
Approximate Complementary Slackness Condition.

Approximate Primal Complementary Slackness Condition.
Let $\alpha \geq 1$
For each $1 \leq j \leq n$ either $x_j = 0$, or $\frac{c_j}{\alpha} \leq \sum_{i=1}^{m} a_{i,j}y_i \leq c_j$

Approximate Dual Complementary Slackness Condition.
Let $\beta \geq 1$
For each $1 \leq i \leq m$ either $y_i = 0$, or $b_i \leq \sum_{j=1}^{n} a_{i,j}x_j \leq \beta b_i$

Exercise
If $x$ and $y$ are primal and dual feasible solutions satisfying the conditions stated above then

$$\sum_{j=1}^{n} c_j x_j \leq \alpha \beta \sum_{i=1}^{m} b_i y_i$$

Exercise
For the primal dual analysis for set cover show $\alpha = 1$, $\beta = f$. 
Garrett: What does dual variable $y_S$ mean in the $s$-$t$ path problem?

Garrett: In the set cover problem the dual variables imply prize for items. I do not understand what $y_S$ mean.

B: It is the same packing problem. Here edge $e$ corresponds to set that we are trying to pack as tightly as possible using cuts.

Garrett: But what does it mean?

Vazirani: Dual program is trying to maximize the sum of the dual variables $y_S$ subject to the condition that no edge feels more dual than its cost, i.e., no edge is overtight.

Garrett: Shmoys & Williamson: Dual variables $y_S$ can be interpreted as “moats” surrounding the set $S$ of width $y_S$. Moats must be non-overlapping, and thus for any edge $e$, we cannot have edge $e$ crossing moats of total width more than $c_e$. 

Garrett:
Garrett: What does dual variable $y_S$ mean in the $s$-$t$ path problem?

Garrett: In the set cover problem the dual variables imply prize for items. I do not understand what $y_S$ mean.
B: It is the same packing problem. Here edge $e$ corresponds to set that we are trying to pack as tightly as possible using cuts.
Garrett: But what does it mean?
Vazirani: Dual program is trying to maximize the sum of the dual variables $y_S$ subject to the condition that no edge feels more dual than its cost, i.e., no edge is overtight.
Garrett: Unhappy!!
Shmoys & Williamson: Dual variables $y_S$ can be interpreted as “moats” surrounding the set $S$ of width $y_S$. Moats must be non overlapping, and thus for any edge $e$, we cannot have edge $e$ crossing moats of total width more than $c_e$.
Garrett: Unhappy!!
Dijkstra's Shortest Path Algorithm.

1. \( d(s) = 0, \ d(v) = \infty, \ v \neq s. \)
2. \( \text{current} \rightarrow s. \ \text{unvisited}(v) = \text{true}, \forall v \in V \)
3. For the current node, set \( \text{unvisited}(\text{current}) = \text{false} \)
   3.1 Consider all of its unvisited neighbors and calculate their tentative distances.
   3.2 If the newly calculated tentative distance is less than its current assigned value, update it to the new value.
4. If \( \text{unvisited}(t) = \text{false} \) return distance of \( t \)
5. Select the unvisited node that is marked with the smallest tentative distance, and set it as the new ”current node” then go back to step 3.
Dijkstra's Shortest Path Algorithm.
Dijkstra’s Shortest Path Algorithm.
s-t shortest path: primal dual
Steiner Forest Problem

Problem

*Given an undirected graph* $G = (V, E)$, *a cost function on edges* $c : E \to \mathbb{Q}^+$, *and a collection of disjoint subsets of vertices* $S_1, S_2, \ldots, S_k$, *find a minimum cost subgraph in which each pair of vertices belonging to the same set* $S_i$ *is connected.*

*Define connectivity requirement function* $r$ *that maps unordered pairs of vertices to* $\{0, 1\}$ *as follows:*

$$r(u, v) = \begin{cases} 
1 & \text{if } u \text{ and } v \text{ belong to the same set } S_i; \\
0 & \text{otherwise.}
\end{cases}$$
LP-Relaxation and Dual

Define a function on all cuts in $G$, $f : 2^V \rightarrow \{0, 1\}$

\[ f(S) = \begin{cases} 
1 & \text{if } \exists u \in S \text{ and } v \in \bar{S} \text{ and } r(u, v) = 1; \\
0 & \text{otherwise.} 
\end{cases} \]

\[
\begin{align*}
\min & \quad \sum_{e \in E} c_e x_e \\
\text{subject to} & \quad \sum_{e : e \in \delta(S)} x_e \geq f(S), \\
& \quad x_e \geq 0 \quad \forall e \in E
\end{align*}
\]

\[
\begin{align*}
\min & \quad \sum_{S \subseteq V} f(S) y_S \\
\text{subject to} & \quad \sum_{S : e \in \delta(S)} y_S \leq c_e, \\
& \quad y_S \geq 0 \quad \forall S \subseteq V
\end{align*}
\]
First Attempt: Primal Dual
First Attempt: Primal Dual

1. $y \leftarrow 0$
2. $F \leftarrow \phi$
3. while there exists a $u$-$v$ pair with $r(u, v) = 1$ but not connected in $F$
   ▶ Let $C$ be a connected component with $f(C) = 1$. Increase $y_C$ until there is an edge $e' \in \delta(C)$ such that $\sum_{S: e' \in \delta S} y_S = c_{e'}$
   ▶ $F \leftarrow F \cup \{e'\}$
4. Visit edges in $F$ in reversed-order and remove any edge that is not required. Call the final solution $F'$
First Attempt: Primal Dual

Analysis.

\[
\text{cost}(F') = \sum_{e \in F'} c_e \\
= \sum_{e \in F'} \sum_{S : e \in \delta(S)} y_S \\
= \sum_S y_S |\delta(S) \cap F'| 
\]
First Attempt: Primal Dual

Analysis.

$$\text{cost}(F') = \sum_{e \in F'} c_e$$

$$= \sum_{e \in F'} \sum_{S: e \in \delta(S)} y_S$$

$$= \sum_S y_S |\delta(S) \cap F'|$$

Can be as bad as $k$
Primal Dual: Simultaneously Increase Dual Variables

1. \( y \leftarrow 0 \)
2. \( F \leftarrow \emptyset \)
3. \( l \leftarrow 0 \)
4. while not all \( u-v \) pairs with \( r(u, v) = 1 \) are connected in \((V, F)\) do
   ▶ \( l \leftarrow l + 1 \)
   ▶ Let \( C \) be the set of all connected components \( C \) of \((V, F)\) such that \( f(C) = 1 \)
   ▶ Increase \( y_C \) for all \( C \in C \) uniformly until for some \( e_l \in \delta(C') \), \( C' \in C \sum_{S: e_l \in \delta(S)} y_S = c_{e_l} \)
   ▶ \( F \leftarrow F \cup \{e_l\} \)
5. \( F' \leftarrow F \)
6. for \( k \leftarrow l \) downto 1 do
   ▶ if \( F' - e_k \) is a feasible solution then
   ▶ Remove \( e_k \) from \( F' \)
Primal Dual: Simultaneously Increase Dual Variables

$r(s,t)=1, \ r(u,v)=1$

$OPT=45$
Primal Dual: Simultaneously Increase Dual Variables

COST = 16 + 20 + 6 + 12 = 54
OPT = 45
Primal Dual: Simultaneously Increase Dual Variables

Analysis.

\[
\text{cost}(F') = \sum_{e \in F'} c_e \\
= \sum_{e \in F'} \sum_{S : e \in \delta(S)} y_S \\
= \sum_{S} y_S |\delta(S) \cap F'| \\
\leq 2 \sum_{S} y_S \quad \text{Want to show!}
\]
Primal Dual: Simultaneously Increase Dual Variables

Analysis.

\[ \sum_S y_S |\delta(S) \cap F'| \leq 2 \sum_S y_S \quad \text{Want to show!} \]

Initially the relation is true.
Consider any iteration.
- \( y_S, S \in C \) say increase by \( \epsilon \).
- LHS increases by \( \epsilon \sum_{S \in C} |\delta(S) \cap F'| \)
- RHS increases by \( 2\epsilon |C| \)

We will show for every iteration \( \sum_{S \in C} |\delta(S) \cap F'| \leq 2|C| \)
Analysis. We will show for every iteration
\[ \sum_{S \in C} |\delta(S) \cap F'| \leq 2|C| \]
- Show \( F \) is a forest
- Shrink connected components at the beginning of the iteration to vertices.
- \( |\delta(S) \cap F'| \) is the degree of \( S \) at the beginning of the iteration, considering final edges in \( F' \)
- Average degree in a forest is at most 2.
Primal Dual: Simultaneously Increase Dual Variables

**Analysis.** We will show for every iteration
\[ \sum_{S \in \mathcal{C}} |\delta(S) \cap F'| \leq 2|\mathcal{C}| \]

- Show \( F \) is a forest (initially \( F \) is a forest, always adds edge with only one end point in a connected component)
- Shrink connected components at the beginning of the iteration to vertices.
- \( |\delta(S) \cap F'| \) is the degree of \( S \) at the beginning of the iteration, considering final edges in \( F' \)
- Average degree in a forest is at most 2.
Primal Dual: Simultaneously Increase Dual Variables

Analysis. We will show for every iteration
\[ \sum_{S \in C} |\delta(S) \cap F'| \leq 2|C| \]

- Show \( F \) is a forest (initially \( F \) is a forest, always adds edge with only one end point in a connected component)
- Shrink connected components at the beginning of the iteration to vertices.
- \( |\delta(S) \cap F'| \) is the degree of \( S \) at the beginning of the iteration, considering final edges in \( F' \)
- Average degree in a forest is at most 2. (this may include some connected components with \( y_S = 0 \))
Primal Dual: Simultaneously Increase Dual Variables

Analysis. We will show for every iteration
\[ \sum_{S \in C} |\delta(S) \cap F'| \leq 2|C| \]

- Show $F$ is a forest (initially $F$ is a forest, always adds edge with only one end point in a connected component)
- Shrink connected components at the beginning of the iteration to vertices.
- $|\delta(S) \cap F'|$ is the degree of $S$ at the beginning of the iteration, considering final edges in $F'$
- Average degree in a forest is at most 2. (this may include some connected components with $y_S = 0$, $f(S) = 0$)
  - Their degrees are always at least 2.
  - Hence average degree of components with $y_S > 0$ is at most 2.
Primal Dual: Simultaneously Increase Dual Variables

**Analysis.** Reds nodes are connected components with $f(S) = 1$. Blue nodes are connected components with $f(S) = 0$

- Blue nodes cannot be leaves. Then the edge crossing the cut of a leaf blue node is redundant.