A Simple Boosting Framework for Transshipment

Goran Zuzic

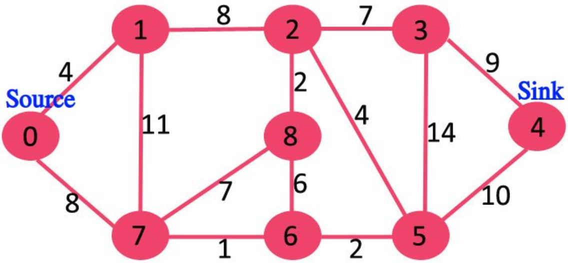
ETH Zürich

01 Mar 2022

Original motivation

We want to solve the **single-source shortest path** problem (**SSSP**).

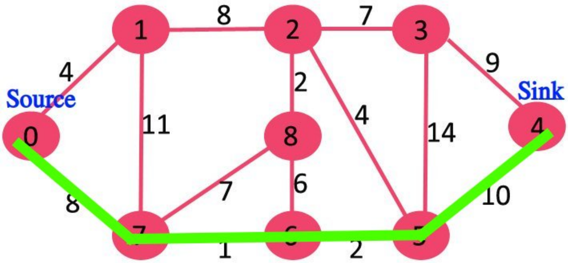
- Given an undirected graph where edges have weights. Compute shortest path from source to all other nodes.



Original motivation

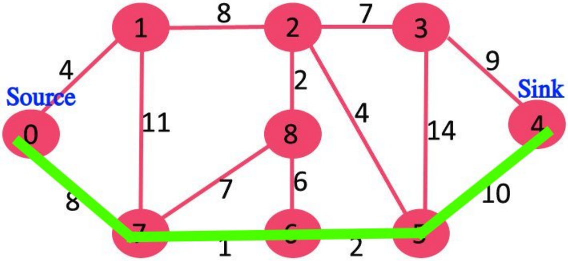
We want to solve the **single-source shortest path** problem (**SSSP**).

- Given an undirected graph where edges have weights. Compute shortest path from source to all other nodes.



We want to solve the **single-source shortest path** problem (**SSSP**).

- Given an undirected graph where edges have weights. Compute shortest path from source to all other nodes.



One of the oldest problems in computer science.
Sequential setting .. easy! Dijkstra's famous $\tilde{O}(m+n)$-time algorithm is optimal (modulo log).

What about parallel or distributed settings? The problem seems much harder: Dijkstra fails miserably.

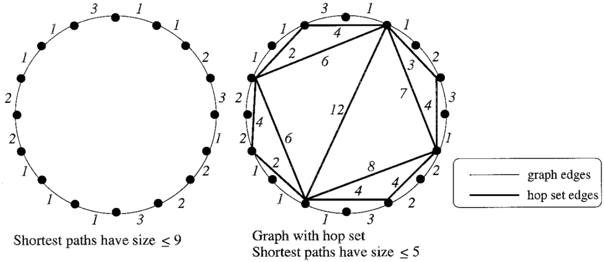
In these settings, significant progress has been made on the $(1+\varepsilon)$-approximate shortest path problem.

Main ideas:

In these settings, significant progress has been made on the $(1+\varepsilon)$-approximate shortest path problem.

Main ideas:

- **Hopset:** Add a small number of edges to a graph such that original shortest paths are $(1+\varepsilon)$-approximated with new paths with small number of hops. (Figure taken from Cohen'00)

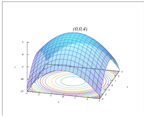


In these settings, significant progress has been made on the $(1+\varepsilon)$-approximate shortest path problem.

Main ideas:

- **Continuous optimization:** (Today) Generalize the shortest path to transshipment. Find an bad approximate solution and boost it to an $(1+\varepsilon)$-approximate.

Recent wins of the continious-optimization approach



Recent results based on **continuous optimization**.

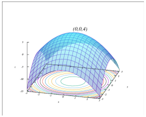
Parallel.

$(1+\varepsilon)$-apx with $\tilde{O}(1)$ depth and $\tilde{O}(m)$ work. [Li; 2020] [ASZ; 2020]

$(1+\varepsilon)$-apx <u>deterministic</u> with $\tilde{O}(1)$ depth and $\tilde{O}(m)$ work. [RGHZL; 2022]

Distributed.

Recent results based on **continuous optimization**.



Parallel.

$(1+\varepsilon)$-apx with $\tilde{O}(1)$ depth and $\tilde{O}(m)$ work. [Li; 2020] [ASZ; 2020]

$(1+\varepsilon)$-apx deterministic with $\tilde{O}(1)$ depth and $\tilde{O}(m)$ work. [RGHZL; 2022]

Distributed.

$(1+\varepsilon)$-apx in $OPT(G) \cdot n^{o(1)}$ rounds. [ZGYHS'22].

1 Introduction

2 Main Aspects of the Solution

- Idea 1: Transshipment generalizes shortest path
- Notation: Consider the LP primal-dual formulation
- Idea 2: Transshipment boosting with duals
- Idea 3: Multiplicative weights
- Idea 4: Self-reduction of transshipment and consequences

3 Conclusion

Idea 1: Transshipment generalizes shortest path

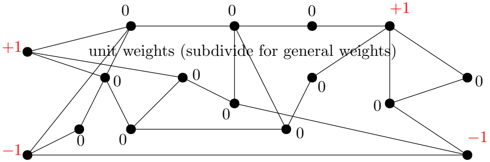
Transshipment.

Idea 1: Transshipment generalizes shortest path

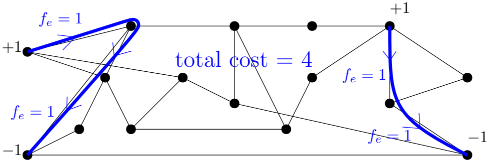
Transshipment. Given a graph $G = (V, E)$ and a demand vector $d \in \mathbb{R}^V$ satisfying $\sum_v d(v) = 0$. Find a flow of minimum cost that satisfies the demands.

Idea 1: Transshipment generalizes shortest path

Transshipment. Given a graph $G = (V, E)$ and a demand vector $d \in \mathbb{R}^V$ satisfying $\sum_v d(v) = 0$. Find a flow of minimum cost that satisfies the demands.

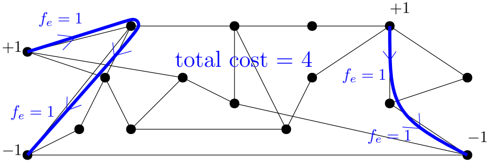


<u>Transshipment.</u> Given a graph $G = (V, E)$ and a demand vector $d \in \mathbb{R}^V$ satisfying $\sum_v d(v) = 0$. Find a flow of minimum cost that satisfies the demands.



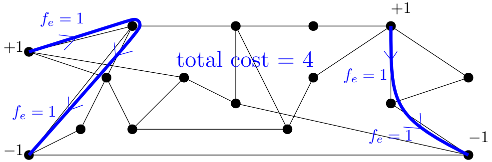
Idea 1: Transshipment generalizes shortest path

Transshipment. Given a graph $G = (V, E)$ and a demand vector $d \in \mathbb{R}^V$ satisfying $\sum_v d(v) = 0$. Find a flow of minimum cost that satisfies the demands.



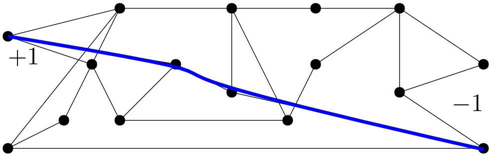
Also known as: uncapacitated min-cost flow, earth mover's distance, Wasserstein metric, optimal transport, transshipment.

Transshipment. Given a graph $G = (V, E)$ and a demand vector $d \in \mathbb{R}^V$ satisfying $\sum_v d(v) = 0$. Find a flow of minimum cost that satisfies the demands.



Also known as: uncapacitated min-cost flow, earth mover's distance, Wasserstein metric, optimal transport, transshipment.

Note. Generalizes $(s - t)$ shortest path. (Also generalizes SSSP.)



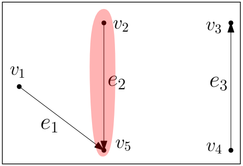
1 Introduction

2 Main Aspects of the Solution
- Idea 1: Transshipment generalizes shortest path
- Notation: Consider the LP primal-dual formulation
- Idea 2: Transshipment boosting with duals
- Idea 3: Multiplicative weights
- Idea 4: Self-reduction of transshipment and consequences

3 Conclusion

Notation: Consider the LP primal-dual formulation

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



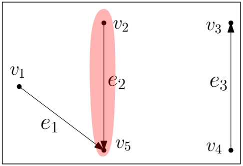
$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \ldots \\ & +1 & & \ldots \\ & & -1 & \ldots \\ & & +1 & \ldots \\ -1 & -1 & & \ldots \end{bmatrix} \end{array}$$

Primal.

Dual.

Notation: Consider the LP primal-dual formulation

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



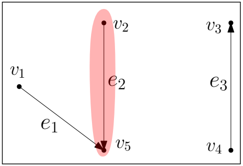
$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & +1 & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & -1 & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \quad \|f\|_1 : Bf = d$$

Dual.

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & +1 & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & -1 & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\boxed{\min_f \quad \|f\|_1 : Bf = d}$$

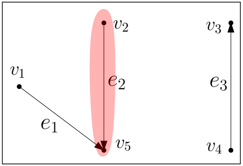
$f_e = 0$ if no flow along e

$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

Dual.

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & +1 & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & -1 & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \quad \|f\|_1 : Bf = d$$

$f_e = 0$ if no flow along e

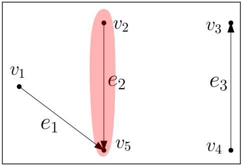
$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

$(Bf)_v = 0$ if flow conserved at v

Dual.

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & +1 & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & -1 & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \quad \|f\|_1 : Bf = d$$

$f_e = 0$ if no flow along e

$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

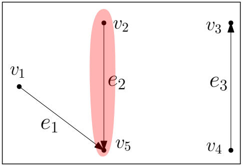
$(Bf)_v = 0$ if flow conserved at v

SP .. $f^* =$ shortest path from s to t

Dual.

Notation: Consider the LP primal-dual formulation

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & \color{red}{e_2} & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & \color{red}{+1} & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & \color{red}{-1} & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \quad \|f\|_1 : Bf = d$$

$f_e = 0$ if no flow along e

$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

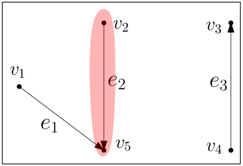
$(Bf)_v = 0$ if flow conserved at v

SP .. $f^* =$ shortest path from s to t

Dual.

$$\max_\phi \quad \langle d, \phi \rangle : \left\| B^\top \phi \right\|_\infty \le 1.$$

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & +1 & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & -1 & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \ \|f\|_1 : Bf = d$$

$f_e = 0$ if no flow along e

$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

$(Bf)_v = 0$ if flow conserved at v

SP .. $f^* =$ shortest path from s to t

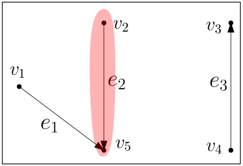
Dual.

$$\max_\phi \ \langle d, \phi \rangle : \left\| B^\top \phi \right\|_\infty \leq 1.$$

$\phi_v =$ potential (height) of v

Notation: Consider the LP primal-dual formulation

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \dots \end{array} \\ \begin{bmatrix} +1 & & & \dots \\ & +1 & & \dots \\ & & -1 & \dots \\ & & +1 & \dots \\ -1 & -1 & & \dots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \quad \|f\|_1 : Bf = d$$

$f_e = 0$ if no flow along e

$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

$(Bf)_v = 0$ if flow conserved at v

SP .. $f^* =$ shortest path from s to t

Dual.

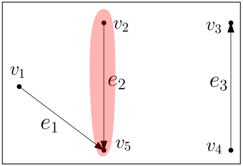
$$\max_\phi \quad \langle d, \phi \rangle : \left\| B^\top \phi \right\|_\infty \leq 1.$$

$\phi_v =$ potential (height) of v

$(B^T \phi)_e = \phi_a - \phi_b$ is height difference

Notation: Consider the LP primal-dual formulation

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & +1 & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & -1 & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \quad \|f\|_1 : Bf = d$$

$f_e = 0$ if no flow along e

$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

$(Bf)_v = 0$ if flow conserved at v

SP .. $f^* =$ shortest path from s to t

Dual.

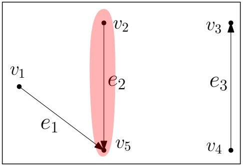
$$\max_\phi \quad \langle d, \phi \rangle : \left\|B^\top \phi\right\|_\infty \leq 1.$$

$\phi_v =$ potential (height) of v

$(B^T\phi)_e = \phi_a - \phi_b$ is height difference

$\left\|B^\top \phi\right\|_\infty \leq 1$ height diff must be small

Write the graph $G = (V, E)$ using the node-edge incidence matrix B.
Steps: (1) Orient edges arbitrarily. (2) For each arc, add column to B.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & \cdots \end{array} \\ \begin{bmatrix} +1 & & & \cdots \\ & +1 & & \cdots \\ & & -1 & \cdots \\ & & +1 & \cdots \\ -1 & -1 & & \cdots \end{bmatrix} \end{array}$$

Primal.

$$\min_f \quad \|f\|_1 : Bf = d$$

$f_e = 0$ if no flow along e

$f_e > 0$ if flow in same direction as e

$f_e < 0$ if flow in opposite direction

$(Bf)_v = 0$ if flow conserved at v

SP .. $f^* =$ shortest path from s to t

Dual.

$$\max_\phi \quad \langle d, \phi \rangle : \left\| B^\top \phi \right\|_\infty \leq 1.$$

$\phi_v =$ potential (height) of v

$(B^T\phi)_e = \phi_a - \phi_b$ is height difference

$\left\| B^\top \phi \right\|_\infty \leq 1$ height diff must be small

SP .. $\phi_v^* =$ distance of v from source

Idea 2: Transshipment boosting with duals

Why transshipment? Isn't it harder?

Amazing property: we can boost a bad approximation to a good approximation.

Idea 2: Transshipment boosting with duals

Why transshipment? Isn't it harder?

Amazing property: we can boost a bad approximation to a good approximation.

Primal.

$\min_f \; \|f\|_1 : Bf = d$

Dual.

$\max_\phi \; \langle d, \phi \rangle : \left\| B^\top \phi \right\|_\infty \leq 1.$

Idea 2: Transshipment boosting with duals

Why transshipment? Isn't it harder?

Amazing property: we can boost a bad approximation to a good approximation.

Primal.

$\min_f \ \|f\|_1 : Bf = d$

Dual.

$\max_\phi \ \langle d, \phi \rangle : \left\|B^\top \phi\right\|_\infty \leq 1.$

Theorem ([Sherman; 2013], [BFKL; 2016], [Zuzic; unpublished])

Fix G. Suppose we are given an oracle $O_G(\cdot)$ which, given a demand d, outputs an α-approximate feasible dual $O_G(d)$. There is an algorithm that produces a $(1+\varepsilon)$-approximate feasible dual by calling $O_G(\cdot)$ at most $\mathrm{poly}(\alpha, \varepsilon^{-1}, \log n)$ times.

Idea 2: Transshipment boosting with duals

Why transshipment? Isn't it harder?

Amazing property: we can boost a bad approximation to a good approximation.

Primal.
$\min_f \ \|f\|_1 : Bf = d$

Dual.
$\max_\phi \ \langle d, \phi \rangle : \left\|B^\top \phi\right\|_\infty \leq 1.$

Theorem ([Sherman; 2013], [BFKL; 2016], [Zuzic; unpublished])

Fix G. Suppose we are given an oracle $O_G(\cdot)$ which, given a demand d, outputs an α-approximate feasible dual $O_G(d)$. There is an algorithm that produces a $(1+\varepsilon)$-approximate feasible dual by calling $O_G(\cdot)$ at most $\text{poly}(\alpha, \varepsilon^{-1}, \log n)$ times.

Corollary

Given such a (dual) $n^{o(1)}$-approximation oracle, we can solve $(1 + \frac{1}{n^{o(1)}})$-approximate transshipment in $n^{o(1)}$ oracle calls.

Feasibility task

$\exists\phi \quad ||A\phi||_\infty + \langle b, \phi\rangle \leq \gamma$

Idea 3: Multiplicative weights

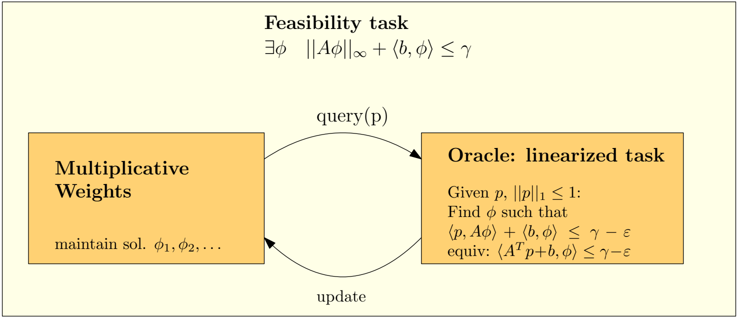
Feasibility task

$\exists \phi \quad ||A\phi||_\infty + \langle b, \phi \rangle \leq \gamma$

Multiplicative Weights

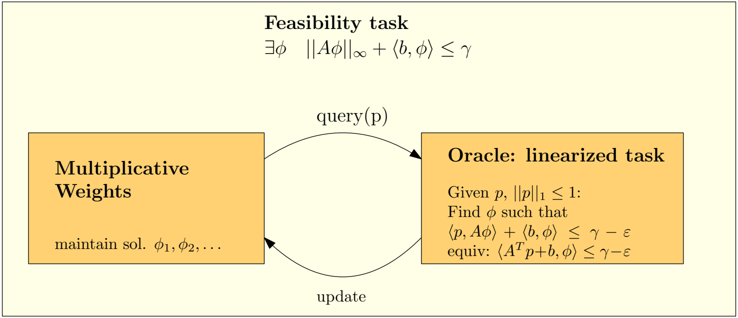
maintain sol. $\phi_1, \phi_2, \ldots$

Idea 3: Multiplicative weights



Question: How many oracle calls?

Idea 3: Multiplicative weights



Question: How many oracle calls?
The oracle will be queried poly$(\varepsilon^{-1}, \log n, \rho)$ times.
Here, $\rho \geq \|A\phi\|_\infty$ called **width of the oracle**.

Idea 4: Self-reduction of transshipment and consequences

Dual: $\max_{\phi} \quad \langle d, \phi \rangle$ such that $\left\| B^{\top} \phi \right\|_{\infty} \leq 1$

Idea 4: Self-reduction of transshipment and consequences

Dual: $\max_{\phi} \quad \langle d, \phi \rangle$ such that $\left\| B^{\top} \phi \right\|_{\infty} \leq 1$

Binary search g: $\exists \phi, \quad \langle d, \phi \rangle \geq g, \quad \left\| B^{\top} \phi \right\|_{\infty} \leq 1$

Idea 4: Self-reduction of transshipment and consequences

Dual: $\max_\phi \quad \langle d, \phi \rangle$ such that $\left\| B^\top \phi \right\|_\infty \leq 1$

Binary search g: $\exists \phi, \quad \langle d, \phi \rangle \geq g, \quad \left\| B^\top \phi \right\|_\infty \leq 1$

Rewrite: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq 1 \geq \left\| B^\top \phi \right\|_\infty$

Idea 4: Self-reduction of transshipment and consequences

Dual: $\max_\phi \quad \langle d, \phi \rangle$ such that $\left\| B^\top \phi \right\|_\infty \le 1$

Binary search g: $\exists \phi, \quad \langle d, \phi \rangle \ge g, \quad \left\| B^\top \phi \right\|_\infty \le 1$

Rewrite: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \ge 1 \ge \left\| B^\top \phi \right\|_\infty$

Eliminate middle: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \ge \left\| B^\top \phi \right\|_\infty$

Idea 4: Self-reduction of transshipment and consequences

Dual: $\max_\phi \quad \langle d, \phi \rangle$ such that $\left\| B^\top \phi \right\|_\infty \leq 1$

Binary search g: $\exists \phi, \quad \langle d, \phi \rangle \geq g, \quad \left\| B^\top \phi \right\|_\infty \leq 1$

Rewrite: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq 1 \geq \left\| B^\top \phi \right\|_\infty$

Eliminate middle: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq \left\| B^\top \phi \right\|_\infty$

Multiplicative weights: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq \left\langle p, B^\top \phi \right\rangle + \varepsilon$

Idea 4: Self-reduction of transshipment and consequences

Dual: $\max_\phi \quad \langle d, \phi \rangle$ such that $\left\| B^\top \phi \right\|_\infty \leq 1$

Binary search g: $\exists \phi, \quad \langle d, \phi \rangle \geq g, \quad \left\| B^\top \phi \right\|_\infty \leq 1$

Rewrite: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq 1 \geq \left\| B^\top \phi \right\|_\infty$

Eliminate middle: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq \left\| B^\top \phi \right\|_\infty$

Multiplicative weights: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq \left\langle p, B^\top \phi \right\rangle + \varepsilon$

Rewrite: $\exists \phi, \quad \left\langle \underbrace{\frac{1}{g} d - Bp}_{d_{\text{residual}}}, \phi \right\rangle \geq \varepsilon$

Dual: $\max_\phi \quad \langle d, \phi \rangle$ such that $\left\| B^\top \phi \right\|_\infty \le 1$

Binary search g: $\exists \phi, \quad \langle d, \phi \rangle \ge g, \quad \left\| B^\top \phi \right\|_\infty \le 1$

Rewrite: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \ge 1 \ge \left\| B^\top \phi \right\|_\infty$

Eliminate middle: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \ge \left\| B^\top \phi \right\|_\infty$

Multiplicative weights: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \ge \left\langle p, B^\top \phi \right\rangle + \varepsilon$

Rewrite: $\exists \phi, \quad \left\langle \underbrace{\frac{1}{g} d - Bp}_{d_{\text{residual}}}, \phi \right\rangle \ge \varepsilon$

Oracle: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \ge \varepsilon$. Note: oracle width $\rho \ge \left\| B^T \phi \right\|_\infty$

Idea 4: Self-reduction of transshipment and consequences

Dual: $\max_\phi \quad \langle d, \phi \rangle$ such that $\left\| B^\top \phi \right\|_\infty \leq 1$

Binary search g: $\exists \phi, \quad \langle d, \phi \rangle \geq g, \quad \left\| B^\top \phi \right\|_\infty \leq 1$

Rewrite: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq 1 \geq \left\| B^\top \phi \right\|_\infty$

Eliminate middle: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq \left\| B^\top \phi \right\|_\infty$

Multiplicative weights: $\exists \phi, \quad \frac{1}{g} \langle d, \phi \rangle \geq \left\langle p, B^\top \phi \right\rangle + \varepsilon$

Rewrite: $\exists \phi, \quad \left\langle \underbrace{\frac{1}{g} d - Bp}_{d_{\text{residual}}}, \phi \right\rangle \geq \varepsilon$

Oracle: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon$. Note: oracle width $\rho \geq \left\| B^T \phi \right\|_\infty$

Equiv: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon / \rho, \quad \left\| B^T \phi \right\|_\infty \leq 1$

Self-reduction consequences

$$\textbf{Oracle: } \exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon/\rho, \quad \left\| B^T \phi \right\|_\infty \leq 1$$

Notes:

- Q: Why does this make progress at all?

Self-reduction consequences

Oracle: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon / \rho, \quad \left\| B^T \phi \right\|_\infty \leq 1$

Notes:

- Q: Why does this make progress at all? Ans: Note that we can increase ρ, the final answer is still $(1+\varepsilon)$-approximate but runtime increases.

Self-reduction consequences

Oracle: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon / \rho, \quad \left\| B^T \phi \right\|_\infty \leq 1$

Notes:

- Q: Why does this make progress at all? Ans: Note that we can increase ρ, the final answer is still $(1 + \varepsilon)$-approximate but runtime increases. Also: we can now handle approximations in the answer. An α-approx changes the width of the oracle.
Approximation $\rightarrow$ runtime.

$$\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon / (\alpha \cdot \rho), \quad \left\| B^T \phi \right\|_\infty \leq 1$$

Self-reduction consequences

Oracle: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon/\rho, \quad \left\| B^T \phi \right\|_\infty \leq 1$

Notes:

- Q: Why does this make progress at all? Ans: Note that we can increase ρ, the final answer is still $(1+\varepsilon)$-approximate but runtime increases. Also: we can now handle approximations in the answer. An α-approx changes the width of the oracle. Approximation $\rightarrow$ runtime.

$$\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon/(\alpha \cdot \rho), \quad \left\| B^T \phi \right\|_\infty \leq 1$$

- Q: What if there is no solution to the new problem?

Self-reduction consequences

Oracle: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon / \rho, \quad \left\| B^T \phi \right\|_\infty \leq 1$

Notes:

- Q: Why does this make progress at all? Ans: Note that we can increase ρ, the final answer is still $(1+\varepsilon)$-approximate but runtime increases. Also: we can now handle approximations in the answer. An α-approx changes the width of the oracle. Approximation $\rightarrow$ runtime.

$$\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon / (\alpha \cdot \rho), \quad \left\| B^T \phi \right\|_\infty \leq 1$$

- Q: What if there is no solution to the new problem? Ans: Look at primal and prove the original problem has no solution.

Oracle: $\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon/\rho, \quad \left\| B^T \phi \right\|_\infty \leq 1$

Notes:

- Q: Why does this make progress at all? Ans: Note that we can increase ρ, the final answer is still $(1+\varepsilon)$-approximate but runtime increases. Also: we can now handle approximations in the answer. An α-approx changes the width of the oracle. Approximation $\rightarrow$ runtime.

$$\exists \phi, \quad \langle d_{\text{residual}}, \phi \rangle \geq \varepsilon/(\alpha \cdot \rho), \quad \left\| B^T \phi \right\|_\infty \leq 1$$

- Q: What if there is no solution to the new problem? Ans: Look at primal and prove the original problem has no solution.
- Computationally, we start with the dual LP and end with a dual LP. Result: Any α-approximate dual solver can be boosted to $(1+\varepsilon)$-approximate dual solver. Runtime: poly$(\alpha, \varepsilon^{-1}, \log n)$ query calls.

1. Introduction
2. Main Aspects of the Solution
   - Idea 1: Transshipment generalizes shortest path
   - Notation: Consider the LP primal-dual formulation
   - Idea 2: Transshipment boosting with duals
   - Idea 3: Multiplicative weights
   - Idea 4: Self-reduction of transshipment and consequences
3. Conclusion

Thank you!