Approximation Algorithms for the Bottleneck Asymmetric Traveling Salesman Problem

Hyung-Chan An   Robert D. Kleinberg   David B. Shmoys

Bottleneck Asymmetric Traveling Salesman Problem

- Given n vertices and costs between them
- Find a Hamiltonian cycle over these n vertices
- Minimizing the *bottleneck* (or maximum-edge) cost

Bottleneck Asymmetric Traveling Salesman Problem

- Given n vertices and costs between them
- Find a Hamiltonian cycle over these n vertices
- Minimizing the *bottleneck* (or maximum-edge) cost
- Asymmetric
  - $c(a, b)$ need not equal to $c(b, a)$
- Metric cost
  - $c(x, z) \leq c(x, y) + c(y, z)$
  - c defined over the complete graph
- NP-hard

The Four Variants

	Bottleneck	Min-Sum
Asymmetric		
Symmetric		

[1] [Asadpour, Goemans, Mądry, Oveis Gharan and Saberi 10], [Christofides 76], [Fleischner 74], [Frieze, Galbiati and Maffioli 82], [Lau 81], [Parker and Rardin 84]

The Four Variants

	Bottleneck	Min-Sum
Asymmetric		$O(\log n / \log\log n)$ $O(\log n)$
Symmetric	2	$3/2$

- No nontrivial performance guarantee known

[1] [Asadpour, Goemans, Mądry, Oveis Gharan and Saberi 10], [Christofides 76], [Fleischner 74], [Frieze, Galbiati and Maffioli 82], [Lau 81], [Parker and Rardin 84]

The Four Variants

	Bottleneck	Min-Sum
Asymmetric	$O(\log n / \log \log n)^{\dagger}$	$O(\log n / \log \log n)$ $O(\log n)$
Symmetric	2	$3/2$

- No nontrivial performance guarantee known
- This paper†

[1] [Asadpour, Goemans, Mądry, Oveis Gharan and Saberi 10], [Christofides 76], [Fleischner 74], [Frieze, Galbiati and Maffioli 82], [Lau 81], [Parker and Rardin 84]

Intro: Combinatorial Problem

Intro: Combinatorial Problem

Problem

Given $H = (V, A)$, either

- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v \rangle$ iff H has path of length $\leq \rho$ from u to v.

Intro: Combinatorial Problem

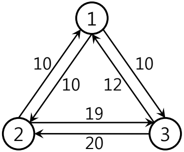
Problem

Given $H = (V, A)$, either

- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v \rangle$ iff H has path of length $\leq \rho$ from u to v.

Example Suppose oracle access for $\rho = 2$



Intro: Combinatorial Problem

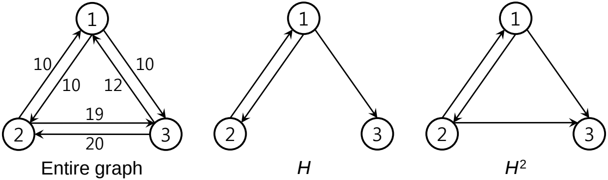
Problem

Given $H = (V, A)$, either

- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v \rangle$ iff H has path of length $\leq \rho$ from u to v.

Example Suppose oracle access for $\rho = 2$



$\tau = 10$

Intro: Combinatorial Problem

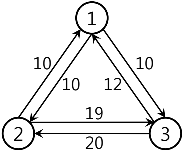
Problem

Given $H = (V, A)$, either

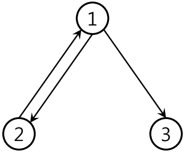
- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v\rangle$ iff H has path of length $\leq \rho$ from u to v.

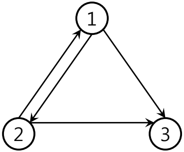
Example Suppose oracle access for $\rho = 2$



Entire graph

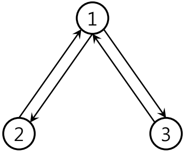


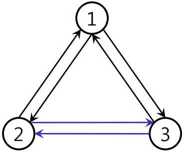
H



H^2

$\tau = 12$





Intro: Combinatorial Problem

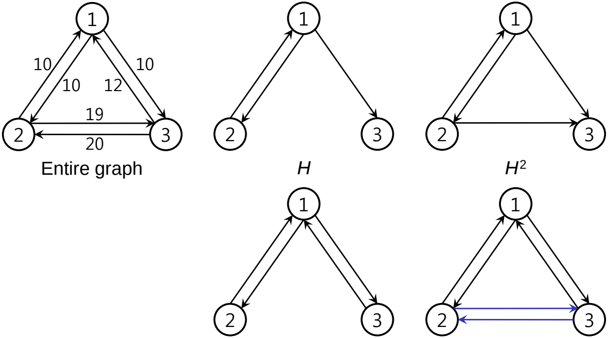
Problem

Given $H = (V, A)$, either

- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v \rangle$ iff H has path of length $\leq \rho$ from u to v.

Example Suppose oracle access for $\rho = 2$



$\tau = 12$

- $\text{OPT} \geq \tau$

Intro: Combinatorial Problem

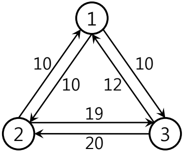
Problem

Given $H = (V, A)$, either

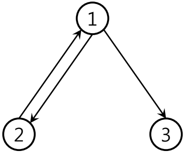
- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v\rangle$ iff H has path of length $\leq \rho$ from u to v.

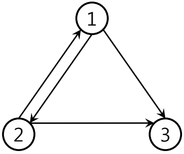
Example Suppose oracle access for $\rho = 2$



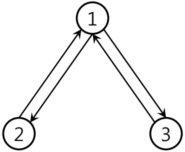
Entire graph

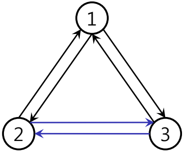


H



H^2





$\tau = 12$

- OPT $\geq \tau$
- Costs of added arcs $\leq 2\tau$
- ALG $\leq 2\tau$

Intro: Combinatorial Problem

Problem

Given $H = (V, A)$, either

- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v\rangle$ iff H has path of length $\leq \rho$ from u to v.

- Reduced from/to the ρ-approximation optimization problem

Intro: Combinatorial Problem

Problem

Given $H = (V, A)$*, either*

- *find a Hamiltonian cycle in* H^ρ*, or*
- *conclude that* H *is non-Hamiltonian;*

where H^ρ *contains* $\langle u, v \rangle$ *iff* H *has path of length* $\leq \rho$ *from* u *to* v*.*

- Reduced from/to the ρ-approximation optimization problem
- Purely combinatorial problem statement

Intro: Combinatorial Problem

Problem

Given $H = (V, A)$, either

- *find a Hamiltonian cycle in H^ρ, or*
- *conclude that H is non-Hamiltonian;*

where H^ρ contains $\langle u, v\rangle$ iff H has path of length $\leq \rho$ from u to v.

- Reduced from/to the ρ-approximation optimization problem
- Purely combinatorial problem statement
- Techniques invented in min-sum version not amenable

Intro: Comparison to the Min-Sum Asymmetric Case

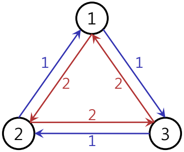
- Approximation algorithms
  - Find a spanning Eulerian circuit with bounded cost
  - Shortcut the circuit to obtain a Hamiltonian cycle
    - Eulerian circuit $v_1 - v_2 - v_3 - v_2 - v_1 - v_4 - v_1$
      After shortcutting $v_1 - v_2 - v_3 - \quad v_4 - v_1$
    - Path $v_3 - v_2 - v_1 - v_4$ is shortcut into a direct arc $\langle v_3, v_4 \rangle$

Intro: Comparison to the Min-Sum Asymmetric Case

- Approximation algorithms
  - Find a spanning Eulerian circuit with bounded cost
  - Shortcut the circuit to obtain a Hamiltonian cycle
    - Eulerian circuit $v_1 - v_2 - v_3 - v_2 - v_1 - v_4 - v_1$
      After shortcutting $v_1 - v_2 - v_3 - \quad v_4 - v_1$
    - Path $v_3 - v_2 - v_1 - v_4$ is shortcut into a direct arc $\langle v_3, v_4 \rangle$
  - Shortcutting a circuit does not increase its cost

Intro: Comparison to the Min-Sum Asymmetric Case

- Shortcutting may increase objective in bottleneck setting

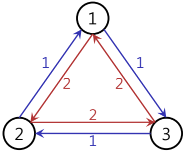


Circuit 1-3-2-1-3-2-1 cost 1

Circuit 1 -2 -3 -1 cost 2

Intro: Comparison to the Min-Sum Asymmetric Case

- Shortcutting may increase objective in bottleneck setting

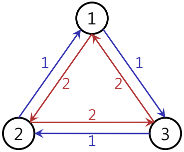


Circuit 1-3-2-1-3-2-1 cost 1
Circuit 1 -2 -3 -1 cost 2

  - Shortcutting becomes the critical step that determines solution cost

Intro: Comparison to the Min-Sum Asymmetric Case

- Shortcutting may increase objective in bottleneck setting



Circuit	1-3-2-1-3-2-1	cost 1
Circuit	1 -2 -3 -1	cost 2

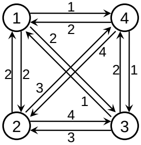
  - Shortcutting becomes the critical step that determines solution cost

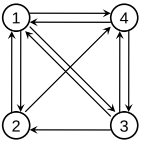
- We devise *bounded length shortcutting lemma*
  - Constructive proof

Overview of the Algorithm

Overview of the Algorithm

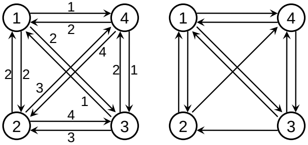
- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$



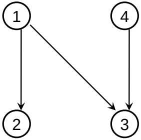


Overview of the Algorithm

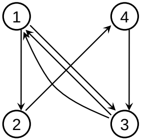
- Determine some τ^* such that H is non-Hamiltonian for all $\tau < \tau^*$



- Find a “nice” weakly-connected acyclic subgraph

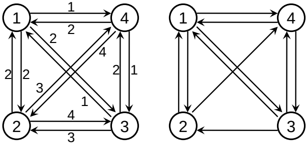


- Augment the graph into a “nice” Eulerian graph

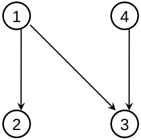


Overview of the Algorithm

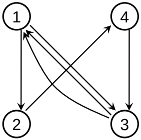
- Determine some τ^* such that H is non-Hamiltonian for all $\tau < \tau^*$



- Find a “nice” weakly-connected acyclic subgraph



- Augment the graph into a “nice” Eulerian graph



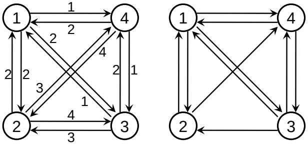
- Shortcut the Eulerian circuit without introducing too long shortcuts

1-3-1-2-4-3-1

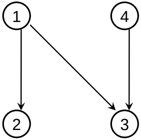
1-3- 2-4- 1

Overview of the Algorithm

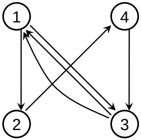
- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]



- Find a “nice” weakly-connected acyclic subgraph



- Augment the graph into a “nice” Eulerian graph



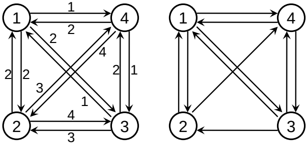
- Shortcut the Eulerian circuit without introducing too long shortcuts

1-3-1-2-4-3-1

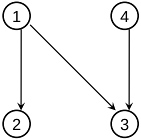
1-3- 2-4- 1

Overview of the Algorithm

- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]

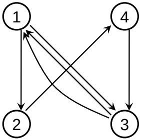


- Find a "nice" weakly-connected acyclic subgraph
  - α-thin trees [AGM+10]



- Augment the graph into a "nice" Eulerian graph

- Shortcut the Eulerian circuit without introducing too long shortcuts

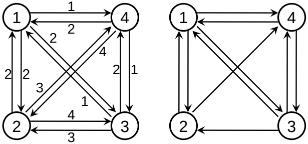


1-3-1-2-4-3-1

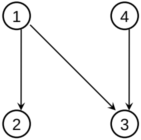
1-3- 2-4- 1

Overview of the Algorithm

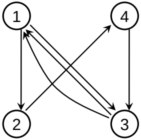
- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]



- Find a “nice” weakly-connected acyclic subgraph
  - α-thin trees [AGM$^+$10]



- Augment the graph into a “nice” Eulerian graph
  - Degree-bounded spanning circuit



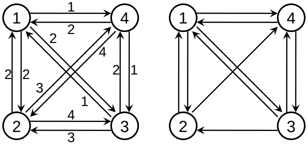
- Shortcut the Eulerian circuit without introducing too long shortcuts

1-3-1-2-4-3-1

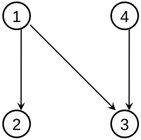
1-3-   2-4-   1

Overview of the Algorithm

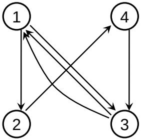
- Determine some τ^* such that H is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]



- Find a "nice" weakly-connected acyclic subgraph
  - α-thin trees [AGM$^+$10]



- Augment the graph into a "nice" Eulerian graph
  - Degree-bounded spanning circuit



- Shortcut the Eulerian circuit without introducing too long shortcuts
  - Bounded length shortcutting lemma

1-3-1-2-4-3-1

1-3- 2-4- 1

Overview of the Algorithm

- Determine some τ^* such that H is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]
- Find a "nice" weakly-connected acyclic subgraph
  - α-thin trees [AGM$^+$10]

  $\alpha = O(\log n / \log\log n)$ [AGM$^+$10]
- Augment the graph into a "nice" Eulerian graph
  - Degree-bounded spanning circuit

  $k = \lceil 4\alpha \rceil$
- Shortcut the Eulerian circuit without introducing too long shortcuts
  - Bounded length shortcutting lemma

  $(2k - 1)$-approx. alg.

Overview of the Algorithm

- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]
- Find a "nice" weakly-connected acyclic subgraph
  - α-thin trees [AGM$^+$10]
- Augment the graph into a "nice" Eulerian graph
  - Degree-bounded spanning circuit
- Shortcut the Eulerian circuit without introducing too long shortcuts
  - Bounded length shortcutting lemma

Preliminaries

- Held-Karp relaxation [HK70]

For $G = (V, A)$,

$$\begin{cases} \sum_{a \in \delta^+(\{v\})} x_a = \sum_{a \in \delta^-(\{v\})} x_a = 1 & \forall v \in V \\ \sum_{a \in \delta^+(S)} x_a \geq 1 & \forall S \subsetneq V, S \neq \emptyset \\ x \geq 0. \end{cases}$$

$$x \in \mathbb{R}^A$$

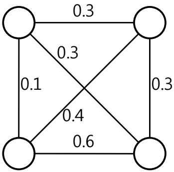
- Relaxation: provides certificate
- Polynomial-time solvable

Overview of the Algorithm

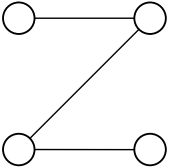
- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]
- Find a “nice” weakly-connected acyclic subgraph
  - α-thin trees [AGM$^+$10]
- Augment the graph into a “nice” Eulerian graph
  - Degree-bounded spanning circuit
- Shortcut the Eulerian circuit without introducing too long shortcuts
  - Bounded length shortcutting lemma

Preliminaries

- Thin tree
  - $z \in \mathbb{R}^E$, T: a spanning tree of $G = (V, E)$



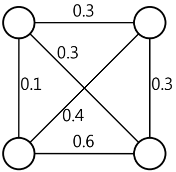
$G = (V, E), z \in \mathbb{R}^E$



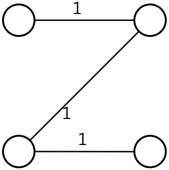
T

Preliminaries

- Thin tree
  - $z \in \mathbb{R}^E$, T: a spanning tree of $G = (V, E)$
  - Consider G as weighted by z, T as unit-weighted



$G = (V, E), z \in \mathbb{R}^E$



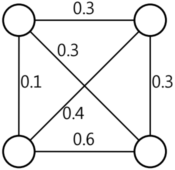
T

Preliminaries

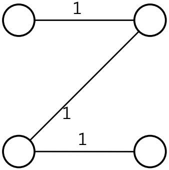
- Thin tree
  - $z \in \mathbb{R}^E$, T: a spanning tree of $G = (V, E)$
  - Consider G as weighted by z, T as unit-weighted

Definition ([AGM+10])

T is *α-thin* if the weight of every cut in T is at most α times its weight in G.



$G = (V, E), z \in \mathbb{R}^E$



T

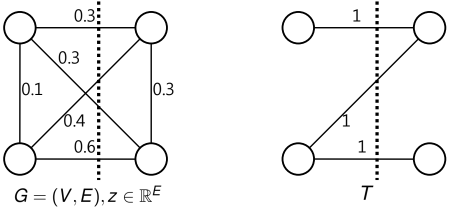
- An α-thin tree can be considered as a succinct representation of α-approx. lower-bounds of cut weights

Preliminaries

- Thin tree
  - $z \in \mathbb{R}^E$, T: a spanning tree of $G = (V, E)$
  - Consider G as weighted by z, T as unit-weighted

Definition ([AGM+10])

T is *α-thin* if the weight of every cut in T is at most α times its weight in G.



- An α-thin tree can be considered as a succinct representation of α-approx. lower-bounds of cut weights
- $1 + 1 + 1 \leq 2 \cdot (0.3 + 0.3 + 0.4 + 0.6) = 2 \cdot 1.6$
- Given T is 2-thin w.r.t. z.

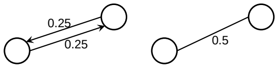
Preliminaries

- Thin tree
  - $z \in \mathbb{R}^E$, T: a spanning tree of $G = (V, E)$
  - Consider G as weighted by z, T as unit-weighted

Definition ([AGM+10])

T is *α-thin* if the weight of every cut in T is at most α times its weight in G.

- x^*: extreme point solution to HK relaxation
- $z^*_{uv} := x^*_{uv} + x^*_{vu}$



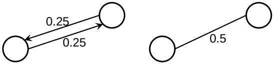
Preliminaries

- Thin tree
  - $z \in \mathbb{R}^E$, T: a spanning tree of $G = (V, E)$
  - Consider G as weighted by z, T as unit-weighted

Definition ([AGM+10])

T is *α-thin* if the weight of every cut in T is at most α times its weight in G.

- x^*: extreme point solution to HK relaxation
- $z^*_{uv} := x^*_{uv} + x^*_{vu}$



Theorem ([AGM+10])

There exists a probabilistic algorithm that produces an α-thin tree T with respect to z^* with high probability, for $\alpha = \frac{4 \ln n}{\ln \ln n}$.

Overview of the Algorithm

- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]
- Find a "nice" weakly-connected acyclic subgraph
  - α-thin trees [AGM$^+$10]
- Augment the graph into a "nice" Eulerian graph
  - Degree-bounded spanning circuit
- Shortcut the Eulerian circuit without introducing too long shortcuts
  - Bounded length shortcutting lemma

Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, ..., $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, ..., $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

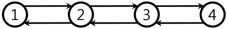
Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, ..., $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Example



Circuit 1-2-3-4-3-2-1 on G,
$k = 2$
P_1 =1 2, P_2 =3 4, P_3 =3 2

- Union of P_2 and P_3 contains $3(> 2)$ distinct vertices.

Proof

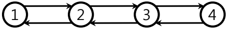
Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, …, $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Example



Circuit 1-2-3-4-3-2-1 on G,
$k = 2$
$P_1 =$1 2, $P_2 =$3 4, $P_3 =$3 2

- Union of P_2 and P_3 contains $3 (> 2)$ distinct vertices.

Proof

- $\{P_1, \ldots, P_\ell\}$ has a transversal: can choose one vertex from each P with no duplicates.

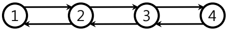
Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, …, $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Example



Circuit 1-2-3-4-3-2-1 on G,
$k = 2$
$P_1 =$1 2, $P_2 =$3 4, $P_3 =$3 2

- Union of P_2 and P_3 contains $3(> 2)$ distinct vertices.

Proof

- $\{P_1, \ldots, P_\ell\}$ has a transversal: can choose one vertex from each P with no duplicates.

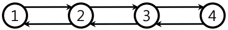
Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, …, $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Example



Circuit 1-2-3-4-3-2-1 on G, $k = 2$

$P_1 =$1 2, $P_2 =$3 4, $P_3 =$3 2

- Union of P_2 and P_3 contains $3 (> 2)$ distinct vertices.

Proof

- $\{P_1, \ldots, P_\ell\}$ has a transversal: can choose one vertex from each P with no duplicates.
- Take any subsequence containing every vertex exactly once and including the transversal

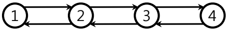
Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, …, $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Example



Circuit 1-2-3-4-3-2-1 on G,
$k = 2$
$P_1 =$1 2, $P_2 =$3 4, $P_3 =$3 2

- Union of P_2 and P_3 contains $3 (> 2)$ distinct vertices.

Proof

- $\{P_1, \ldots, P_\ell\}$ has a transversal: can choose one vertex from each P with no duplicates.
- Take any subsequence containing every vertex exactly once and including the transversal

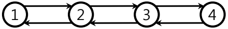
Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, …, $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Example



Circuit 1-2-3-4-3-2-1 on G, $k = 2$

$P_1 =$1 2, $P_2 =$3 4, $P_3 =$3 2

- Union of P_2 and P_3 contains 3($>$ 2) distinct vertices.
- G^3 is Hamiltonian.

Proof

- $\{P_1, \ldots, P_\ell\}$ has a transversal: can choose one vertex from each P with no duplicates.
- Take any subsequence containing every vertex exactly once and including the transversal
- Any two contiguous vertices in this subsequence are $\leq$ 2k-1 arcs apart

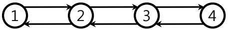
Algorithm: Bounded Length Shortcutting Lemma

- $v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, …, $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.

Lemma

If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Example



Circuit 1-2-3-4-3-2-1 on G, $k = 2$

$P_1 =$1 2, $P_2 =$3 4, $P_3 =$3 2

- Union of P_2 and P_3 contains $3 (> 2)$ distinct vertices.
- G^3 is Hamiltonian.

Proof

- $\{P_1, \ldots, P_\ell\}$ has a transversal: can choose one vertex from each P with no duplicates.
- Take any subsequence containing every vertex exactly once and including the transversal
- Any two contiguous vertices in this subsequence are $\leq 2k$-1 arcs apart
- Polynomial time $\square$

Algorithm: Bounded Length Shortcutting Lemma

Lemma

$v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, …, $P_\ell = v_{(\ell-1)k+1} \ldots v_m$. If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Definition

A *degree-bounded spanning circuit* with bound k is a circuit that visits every vertex at least once and at most k times.

Algorithm: Bounded Length Shortcutting Lemma

Lemma

$v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, ..., $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.
If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Definition

A *degree-bounded spanning circuit* with bound k is a circuit that visits every vertex at least once and at most k times.

Lemma

Given a degree-bounded spanning circuit on G with bound k, a Hamiltonian cycle in G^{2k-1} can be found in polynomial time.

Algorithm: Bounded Length Shortcutting Lemma

Lemma

$v_1, \ldots, v_m, v_1$: a (non-simple) spanning circuit on G, partitioned into $P_1 = v_1 \ldots v_k$, $P_2 = v_{k+1} \ldots v_{2k}$, ..., $P_\ell = v_{(\ell-1)k+1} \ldots v_m$.
If, for all t, the union of any t sets in $\{P_1, \ldots, P_\ell\}$ contains at least t distinct vertices, G^{2k-1} is Hamiltonian.

Definition

A *degree-bounded spanning circuit* with bound k is a circuit that visits every vertex at least once and at most k times.

Lemma

Given a degree-bounded spanning circuit on G with bound k, a Hamiltonian cycle in G^{2k-1} can be found in polynomial time.

Proof

- For any t sets in $\{P_1, \ldots, P_\ell\}$, sum of their cardinalities is strictly greater than $(t-1)k$
- At least t distinct vertices in union $\square$

$$
\begin{array}{ll}
 & \overbrace{\quad\quad\quad\quad\quad\quad\quad\quad}^{k} \\
P_1 & v_1 \quad v_2 \quad \ldots \quad v_k \\
P_2 & \boxed{v_{k+1} \quad v_{k+2} \quad \ldots \quad v_{2k}} \\
 & \vdots \\
P_\ell & \boxed{v_{**} \quad \ldots \quad v_m}
\end{array}
$$

Overview of the Algorithm

- Determine some τ^* such that *H* is non-Hamiltonian for all $\tau < \tau^*$
  - Non-Hamiltonicity certificate: Held-Karp relaxation [HK70]
- Find a “nice” weakly-connected acyclic subgraph
  - α-thin trees [AGM$^+$10]
- Augment the graph into a “nice” Eulerian graph
  - Degree-bounded spanning circuit
- Shortcut the Eulerian circuit without introducing too long shortcuts
  - Bounded length shortcutting lemma

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- x^*: HK solution, z^* unoriented version
- T: α-thin tree w.r.t. z^*, $T_\rightarrow$: directed version of T

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- x^*: HK solution, z^* unoriented version
- T: α-thin tree w.r.t. z^*, $T_\rightarrow$: directed version of T
- Circulation problem

$$\begin{aligned} l(a) &= \begin{cases} 1 & \text{if } a \in T_\rightarrow \\ 0 & \text{otherwise} \end{cases} \\ u(a) &= \begin{cases} 2\alpha x_a^* + 1 & \text{if } a \in T_\rightarrow \\ 2\alpha x_a^* & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

  - (1) is feasible [AGM+10]

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- x^*: HK solution, z^* unoriented version
- T: α-thin tree w.r.t. z^*, $T_\rightarrow$: directed version of T
- Circulation problem

$$l(a) = \begin{cases} 1 & \text{if } a \in T_\rightarrow \\ 0 & \text{otherwise} \end{cases}$$
$$u(a) = \begin{cases} 2\alpha x_a^* + 1 & \text{if } a \in T_\rightarrow \\ 2\alpha x_a^* & \text{otherwise.} \end{cases} \tag{1}$$

  - (1) is feasible [AGM+10]
  - Need an integral solution
  - Highest degree vertex in the support of HK extreme point solution can have $\Theta(n)$ indegree and outdegree
    - Simple round-up does not work

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

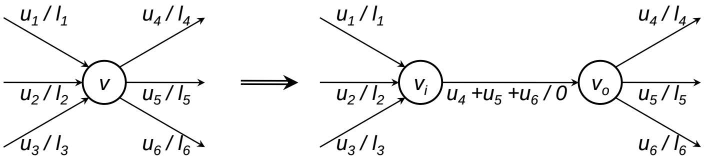
- x^*: HK solution, z^* unoriented version
- T: α-thin tree w.r.t. z^*, $T_\rightarrow$: directed version of T
- Circulation problem

$$\begin{aligned} l(a) &= \begin{cases} 1 & \text{if } a \in T_\rightarrow \\ 0 & \text{otherwise} \end{cases} \\ u(a) &= \begin{cases} 2\alpha x^*_a + 1 & \text{if } a \in T_\rightarrow \\ 2\alpha x^*_a & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

  - (1) is feasible [AGM+10]
  - Need an integral solution
  - Highest degree vertex in the support of HK extreme point solution can have $\Theta(n)$ indegree and outdegree
    - Simple round-up does not work
  - *Vertex capacities* are introduced

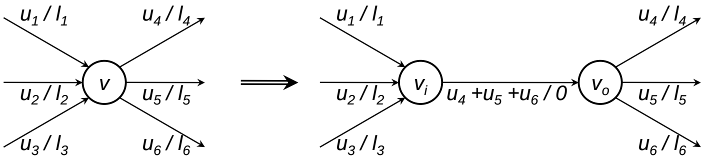
Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- Circulation problem instance (cont'd)
  - *Vertex capacities* are introduced



Algorithm: Constructing Deg.-Bdd. Spanning Circuit

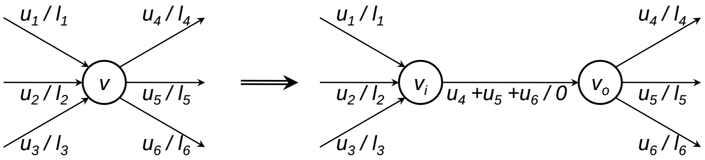
- Circulation problem instance (cont'd)
  - *Vertex capacities* are introduced



  - Feasibility unchanged
  - Contracting split vertices yields solution to original instance

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- Circulation problem instance (cont'd)
  - *Vertex capacities* are introduced



  - Feasibility unchanged
  - Contracting split vertices yields solution to original instance
  - Round up the upper bounds: introduces < 1 error (degree can be bounded by $\lceil u_4 + u_5 + u_6 \rceil$, not $\lceil u_4 \rceil + \lceil u_5 \rceil + \lceil u_6 \rceil$)
  - Want: sum of outgoing upper bounds bounded by $O(\log n / \log\log n)$

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- Bounding fractional outdegree

$$u(a) = \begin{cases} 2\alpha x_a^* + 1 & \text{if } a \in T_\rightarrow \\ 2\alpha x_a^* & \text{otherwise.} \end{cases}$$

  - Upper bound is the sum of Held-Karp solution scaled by 2α and tree-induced lower bound

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- Bounding fractional outdegree

$$u(a) = \begin{cases} 2\alpha x_a^* + 1 & \text{if } a \in T_\rightarrow \\ 2\alpha x_a^* & \text{otherwise.} \end{cases}$$

  - Upper bound is the sum of Held-Karp solution scaled by 2α and tree-induced lower bound
  - Fractional outdegree of HK solution is exactly 1

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- Bounding fractional outdegree

$$u(a) = \begin{cases} 2\alpha x_a^* + 1 & \text{if } a \in T_\rightarrow \\ 2\alpha x_a^* & \text{otherwise.} \end{cases}$$

  - Upper bound is the sum of Held-Karp solution scaled by 2α and tree-induced lower bound
  - Fractional outdegree of HK solution is exactly 1
  - Fractional degree of the unoriented version is exactly 2
  - Maximum degree of α-thin tree is bounded by 2α by the thinness of singleton cuts

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

- Bounding fractional outdegree

$$u(a) = \begin{cases} 2\alpha x_a^* + 1 & \text{if } a \in T_\rightarrow \\ 2\alpha x_a^* & \text{otherwise.} \end{cases}$$

  - Upper bound is the sum of Held-Karp solution scaled by 2α and tree-induced lower bound
  - Fractional outdegree of HK solution is exactly 1
  - Fractional degree of the unoriented version is exactly 2
  - Maximum degree of α-thin tree is bounded by 2α by the thinness of singleton cuts

  - Thus, the fractional degree is bounded by $4\alpha = O(\log n / \log\log n)$.

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

Lemma

Given an α-thin tree w.r.t. to unoriented HK solution, a degree-bounded spanning circuit with bound $\lceil 4\alpha \rceil$ can be found in polynomial time.

Algorithm: Constructing Deg.-Bdd. Spanning Circuit

Lemma

Given an α-thin tree w.r.t. to unoriented HK solution, a degree-bounded spanning circuit with bound $\lceil 4\alpha \rceil$ can be found in polynomial time.

Lemma

Given a degree-bounded spanning circuit on G with bound k, a Hamiltonian cycle in G^{2k-1} can be found in polynomial time.

Theorem

There exists a probabilistic $O(\frac{\log n}{\log \log n})$-approximation algorithm for the bottleneck asymmetric TSP under a metric cost.

Open Question

Open Question

Question

When G has a feasible Held-Karp relaxation, is G^2 Hamiltonian?

Thank you.