Improved Approximations for Ordered TSP on Near-Metric Graphs

Hans-Joachim Böckenhauer and Monika Steinová

Department of Computer Science, ETH Zürich

January 29th, 2013

Overview

- Near-metric graphs
- Traveling salesman problem (TSP)
- Approximating the TSP in near-metric graphs
- Ordered TSP
- An old algorithm for Ordered TSP
- An improved algorithm

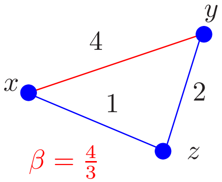
Near-metric graphs

Definition

An undirected (complete) graph $G = (V, E)$ with edge cost function *cost* satisfies the β-triangle inequality if

$$cost(x, y) \leq \beta \cdot (cost(x, z) + cost(z, y))$$

for all $x, y, z \in V$.



$\beta = 1$: Standard triangle inequality

$\beta > 1$: Relaxed triangle inequality

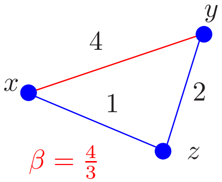
Near-metric graphs

Definition

An undirected (complete) graph $G = (V, E)$ with edge cost function *cost* satisfies the β-triangle inequality if

$$cost(x, y) \leq \beta \cdot (cost(x, z) + cost(z, y))$$

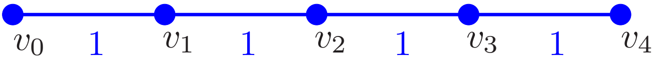
for all $x, y, z \in V$.



$\beta = 1$: Standard triangle inequality

$\beta > 1$: Relaxed triangle inequality

Relaxed triangle inequality – Properties

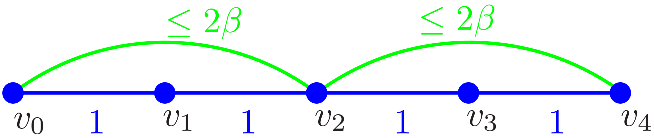


Lemma [Bandelt et al. 1994]

Let $v_0, v_1, \ldots, v_{2^k}$ be a path in G. Then

$$cost(v_0, v_{2^k}) \leq \beta^k \cdot \sum_{i=0}^{2^k-1} cost(v_i, v_{i+1})$$

Relaxed triangle inequality – Properties

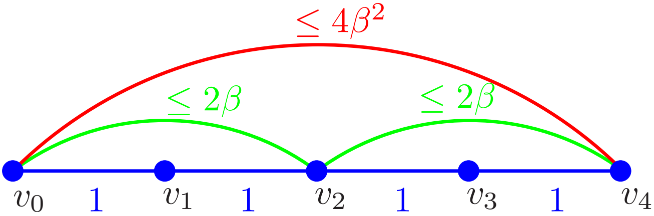


Lemma [Bandelt et al. 1994]

Let $v_0, v_1, \ldots, v_{2^k}$ be a path in G. Then

$$cost(v_0, v_{2^k}) \leq \beta^k \cdot \sum_{i=0}^{2^k-1} cost(v_i, v_{i+1})$$

Relaxed triangle inequality – Properties

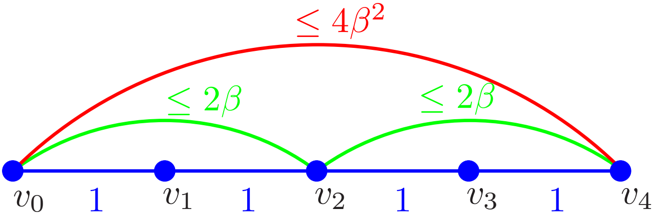


Lemma [Bandelt et al. 1994]

Let $v_0, v_1, \ldots, v_{2^k}$ be a path in G. Then

$$cost(v_0, v_{2^k}) \leq \beta^k \cdot \sum_{i=0}^{2^k-1} cost(v_i, v_{i+1})$$

Relaxed triangle inequality – Properties



Lemma [Bandelt et al. 1994]

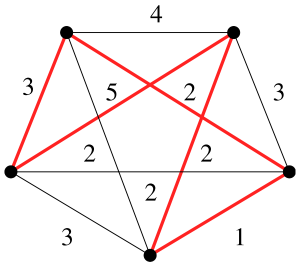
Let $v_0, v_1, \ldots, v_{2^k}$ be a path in G. Then

$$cost(v_0, v_{2^k}) \leq \beta^k \cdot \sum_{i=0}^{2^k-1} cost(v_i, v_{i+1})$$

Definition of TSP

Traveling Salesman Problem (Δ_β-TSP)

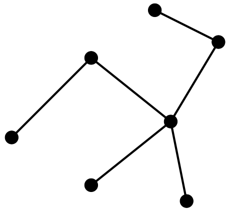
- Input: Undirected complete graph with edge costs
- Edge costs satisfy relaxed β-triangle inequality for $\beta > 1$
- Find a cheapest cycle visiting each vertex exactly once.



Strategy for approximating Δ-TSP ($\beta = 1$)

1. Find spanning subgraph with cost bounded by $\alpha \cdot \mathit{cost}(\mathit{Opt})$ for some α
2. Follow this subgraph to construct Hamiltonian cycle using jumps

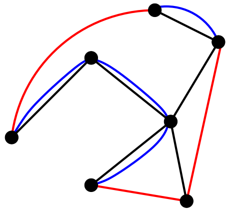
⇒ Jumps do not increase cost due to triangle inequality



Strategy for approximating Δ-TSP ($\beta = 1$)

1. Find spanning subgraph with cost bounded by $\alpha \cdot \textit{cost}(\textit{Opt})$ for some α
2. Follow this subgraph to construct Hamiltonian cycle using jumps

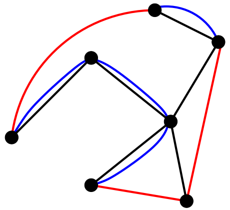
$\Rightarrow$ Jumps do not increase cost due to triangle inequality



Strategy for approximating Δ-TSP ($\beta = 1$)

1. Find spanning subgraph with cost bounded by $\alpha \cdot \mathit{cost}(\mathit{Opt})$ for some α
2. Follow this subgraph to construct Hamiltonian cycle using jumps

⇒ Jumps do not increase cost due to triangle inequality



Strategy for approximating Δ_β-TSP ($\beta > 1$)

- Similar strategy, but:
- Jumps are expensive, their cost grows with the number of bypassed edges

$\Rightarrow$ Try to use short jumps

Example

Along a spanning tree, jumps of length 3 are sufficient

$\Rightarrow$ Approximation ratio $2\beta^2$

Strategy for approximating Δ_β-TSP ($\beta > 1$)

- Similar strategy, but:
- Jumps are expensive, their cost grows with the number of bypassed edges

$\Rightarrow$ Try to use short jumps

Example

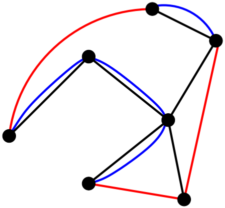
Along a spanning tree, jumps of length 3 are sufficient

$\Rightarrow$ Approximation ratio $2\beta^2$

Strategy for approximating Δ_β-TSP ($\beta > 1$)

- Similar strategy, but:
- Jumps are expensive, their cost grows with the number of bypassed edges

$\Rightarrow$ Try to use short jumps



Example

Along a spanning tree, jumps of length 3 are sufficient

$\Rightarrow$ Approximation ratio $2\beta^2$

Known approximation ratios for Δ_β-TSP

spanning subgraph	approx. ratio	best for
tree [Andreae & Bandelt 1995, Andreae 2001]	$\beta^2 + \beta$	$2 \leq \beta \leq 3$
biconnected graph [Bender & Chekuri 1999]	4β	$\beta \geq 3$
tree + path matching [Böckenhauer et al. 2000]	$\frac{3}{2}\beta^2$	$\beta \leq 2$

Best known ratio for Δ_β-TSP

$$\min\left\{\frac{3}{2}\beta^2, \beta^2 + \beta, 4\beta\right\}$$

Known approximation ratios for Δ_β-TSP

spanning subgraph	approx. ratio	best for
tree [Andreae & Bandelt 1995, Andreae 2001]	$\beta^2 + \beta$	$2 \leq \beta \leq 3$
biconnected graph [Bender & Chekuri 1999]	4β	$\beta \geq 3$
tree + path matching [Böckenhauer et al. 2000]	$\frac{3}{2}\beta^2$	$\beta \leq 2$

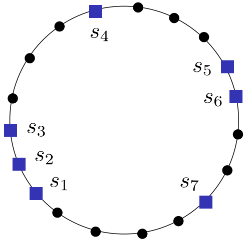
Best known ratio for Δ_β-TSP

$$\min\left\{\frac{3}{2}\beta^2, \beta^2 + \beta, 4\beta\right\}$$

Definition of k-Δ_βOTSP

Ordered TSP (k-Δ_βOTSP)

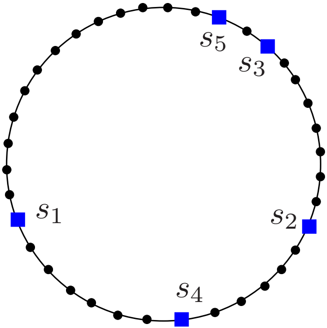
Find a cheapest cycle visiting each vertex exactly once that in addition visits terminal vertices $s_1, \ldots, s_k$ in prescribed order.



Known approximation algorithm for k-Δ_βOTSP

Simple cycle algorithm
[Böckenhauer et al. 2006]

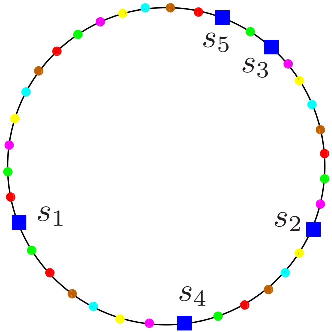
1. Build Hamiltonian cycle H
2. Color non-terminals periodically with $k + 1$ colors
3. Use non-terminals of color i to connect s_i with s_{i+1}
4. Use non-terminals of color $k + 1$ to connect not used non-terminals



Known approximation algorithm for k-Δ_βOTSP

Simple cycle algorithm
[Böckenhauer et al. 2006]

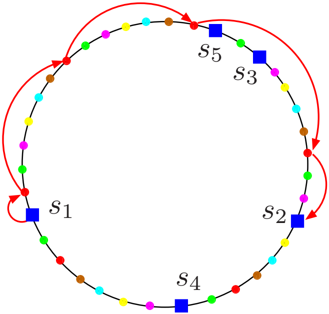
1. Build Hamiltonian cycle H
2. Color non-terminals periodically with $k + 1$ colors
3. Use non-terminals of color i to connect s_i with s_{i+1}
4. Use non-terminals of color $k + 1$ to connect not used non-terminals



Known approximation algorithm for k-Δ_βOTSP

Simple cycle algorithm
[Böckenhauer et al. 2006]

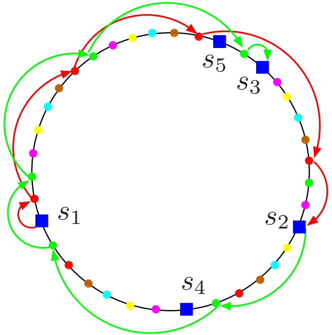
1. Build Hamiltonian cycle H
2. Color non-terminals periodically with $k+1$ colors
3. Use non-terminals of color i to connect s_i with s_{i+1}
4. Use non-terminals of color $k+1$ to connect not used non-terminals



Known approximation algorithm for k-Δ_βOTSP

Simple cycle algorithm
[Böckenhauer et al. 2006]

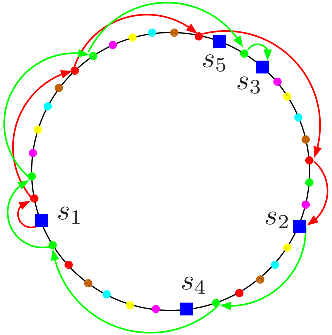
1. Build Hamiltonian cycle H
2. Color non-terminals periodically with $k + 1$ colors
3. Use non-terminals of color i to connect s_i with s_{i+1}
4. Use non-terminals of color $k + 1$ to connect not used non-terminals



Known approximation algorithm for k-Δ_βOTSP

Simple cycle algorithm
[Böckenhauer et al. 2006]

1. Build Hamiltonian cycle H
2. Color non-terminals periodically with $k + 1$ colors
3. Use non-terminals of color i to connect s_i with s_{i+1}
4. Use non-terminals of color $k + 1$ to connect not used non-terminals



Approximation ratio of known k-Δ_βOTSP algorithms

Analysis of simple cycle algorithm

The simple cycle algorithm achieves approx. ratio of

$$\underbrace{(k+1)}_{\#\text{ rounds}} \cdot \underbrace{\beta^{1+\log_2(k-1)}}_{\text{log of jump length}} \cdot \underbrace{\min\left\{\frac{3}{2}\beta^2, \beta^2+\beta, 4\beta\right\}}_{\text{approx. of cycle}}$$

Improvement [Böckenhauer et al. 2010]

- Based on shortening Eulerian tour through doubled spanning tree
- Achieves approximation ratio of

$$k \cdot \beta^{\log_2(\lfloor 3k/2 \rfloor + 1)}$$

Approximation ratio of known k-Δ_βOTSP algorithms

Analysis of simple cycle algorithm

The simple cycle algorithm achieves approx. ratio of

$$\underbrace{(k+1)}_{\#\text{ rounds}} \cdot \underbrace{\beta^{1+\log_2(k-1)}}_{\text{log of jump length}} \cdot \underbrace{\min\left\{\frac{3}{2}\beta^2, \beta^2+\beta, 4\beta\right\}}_{\text{approx. of cycle}}$$

Improvement [Böckenhauer et al. 2010]

- Based on shortening Eulerian tour through doubled spanning tree
- Achieves approximation ratio of

$$k \cdot \beta^{\log_2(\lfloor 3k/2 \rfloor + 1)}$$

Improved cycle algorithm – Idea

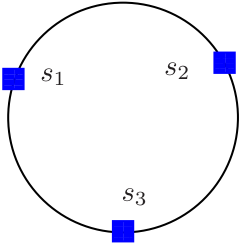
Idea

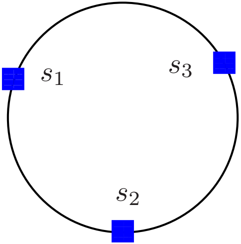
- Three terminals on the Hamiltonian cycle are always in correct order – either clockwise or counterclockwise

⇒ Connect triple of terminals in one round around the Hamiltonian cycle

⇒ Reduces number of rounds

⇒ Reduces jump lengths





Improved cycle algorithm – Idea

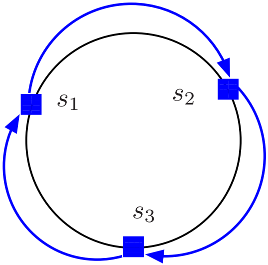
Idea

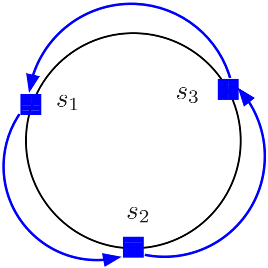
- Three terminals on the Hamiltonian cycle are always in correct order – either clockwise or counterclockwise

⇒ Connect triple of terminals in one round around the Hamiltonian cycle

⇒ Reduces number of rounds

⇒ Reduces jump lengths





Improved cycle algorithm – Idea

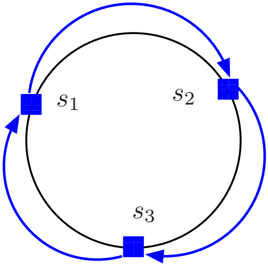
Idea

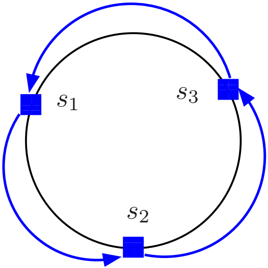
- Three terminals on the Hamiltonian cycle are always in correct order – either clockwise or counterclockwise

⇒ Connect triple of terminals in one round around the Hamiltonian cycle

⇒ Reduces number of rounds

⇒ Reduces jump lengths





Improved cycle algorithm – Idea

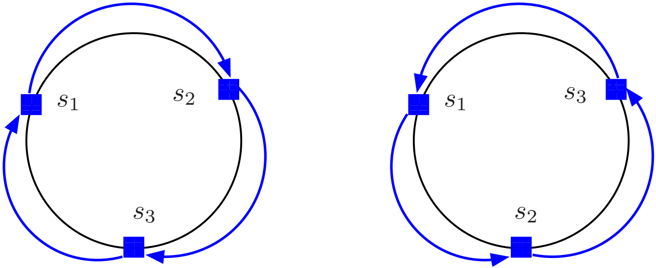
Idea

- Three terminals on the Hamiltonian cycle are always in correct order – either clockwise or counterclockwise

⇒ Connect triple of terminals in one round around the Hamiltonian cycle

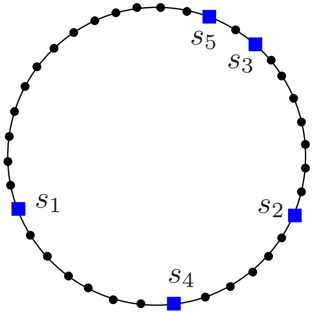
⇒ Reduces number of rounds

⇒ Reduces jump lengths



Improved cycle algorithm – Connecting triples of terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially using $\lceil k/2 \rceil$ colors
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H

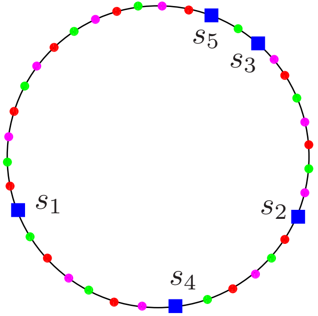


Observation

Each vertex of the graph appears in the cycle *at most* once. Some non-terminals are *not* connected.

Improved cycle algorithm – Connecting triples of terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially using $\lceil k/2 \rceil$ colors
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H

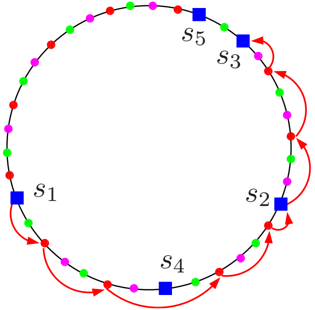


Observation

Each vertex of the graph appears in the cycle *at most* once. Some non-terminals are *not* connected.

Improved cycle algorithm – Connecting triples of terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially using $\lceil k/2 \rceil$ colors
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H

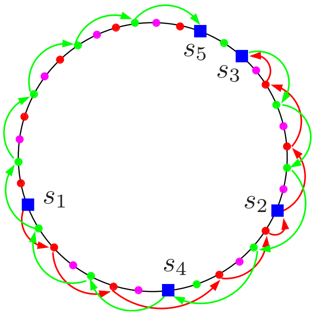


Observation

Each vertex of the graph appears in the cycle *at most* once. Some non-terminals are *not* connected.

Improved cycle algorithm – Connecting triples of terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially using $\lceil k/2 \rceil$ colors
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H

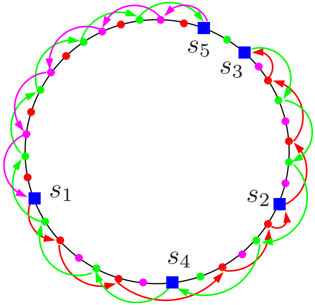


Observation

Each vertex of the graph appears in the cycle *at most* once. Some non-terminals are *not* connected.

Improved cycle algorithm – Connecting triples of terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially using $\lceil k/2 \rceil$ colors
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H

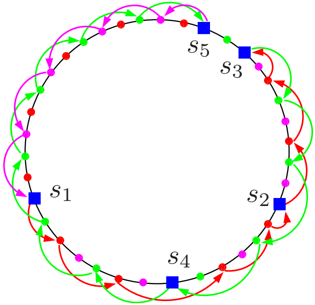


Observation

Each vertex of the graph appears in the cycle *at most* once. Some non-terminals are *not* connected.

Improved cycle algorithm – Connecting triples of terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially using $\lceil k/2 \rceil$ colors
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H

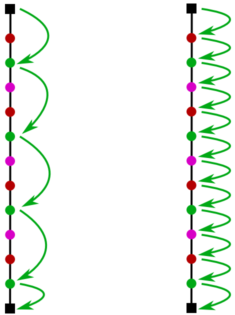


Observation

Each vertex of the graph appears in the cycle *at most* once. Some non-terminals are *not* connected.

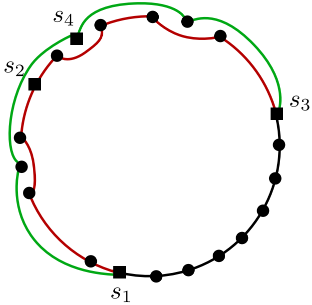
Improved cycle algorithm – Including unused non-terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H
4. Merge non-terminals into paths
5. Extend two paths to cover remaining non-terminals



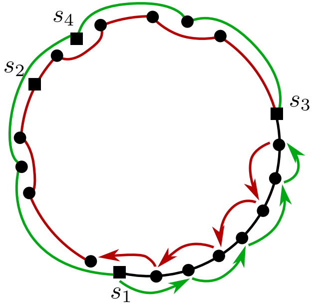
Improved cycle algorithm – Including unused non-terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H
4. Merge non-terminals into paths
5. Extend two paths to cover remaining non-terminals



Improved cycle algorithm – Including unused non-terminals

1. Build a Hamiltonian cycle H
2. Color non-terminals sequentially
3. Connect s_{2i} to s_{2i+1} and s_{2i+1} to s_{2i+2} via vertices of color i on H
4. Merge non-terminals into paths
5. Extend two paths to cover remaining non-terminals

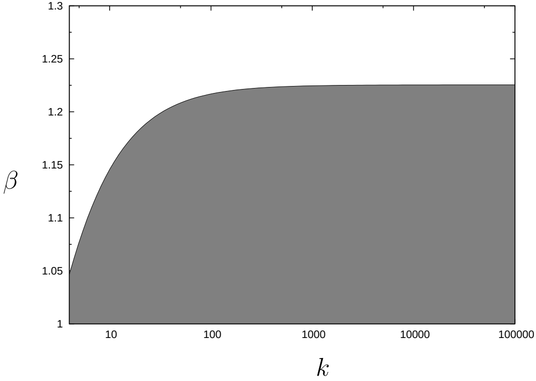


Approximation ratio of improved cycle algorithm

Theorem

The approximation ratio of the improved cycle algorithm is

$$\left\lceil \frac{k}{2} \right\rceil \cdot \beta^{\log_2(2\lceil k/2\rceil + k - 3)} \cdot \min\left\{\frac{3}{2}\beta^2, 4\beta, \beta^2 + \beta\right\}$$

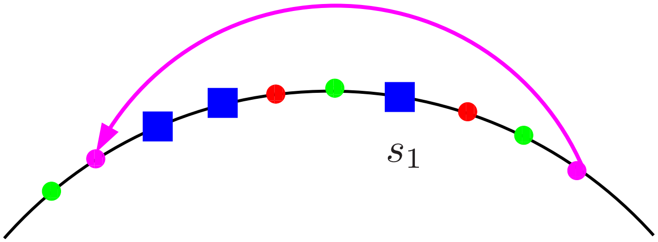


Further possible improvements

Observation

- Jump length $2\lceil k/2 \rceil + k - 3$ can only occur when bypassing s_1

$\Rightarrow$ Finding best starting position for the coloring might lower the jump length, especially for $k \ll n$

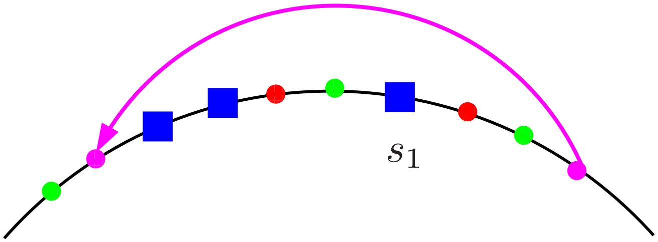


Further possible improvements

Observation

- Jump length $2\lceil k/2 \rceil + k - 3$ can only occur when bypassing s_1

$\Rightarrow$ Finding best starting position for the coloring might lower the jump length, especially for $k \ll n$



Thanks for your attention!