On Cyclic Solutions to the Min-Max Latency Multi-Robot Patrolling Problem

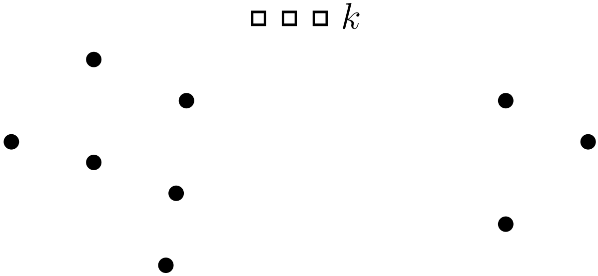
Peyman Afshani, Mark de Berg, Kevin Buchin, Jie Gao*, Maarten Löffler, Amir Nayyeri, Benjamin Raichel, Rik Sarkar, Haotian Wang, Hao-Tsung Yang

Rutgers University
http://sites.rutgers.edu/jie-gao

June 2023

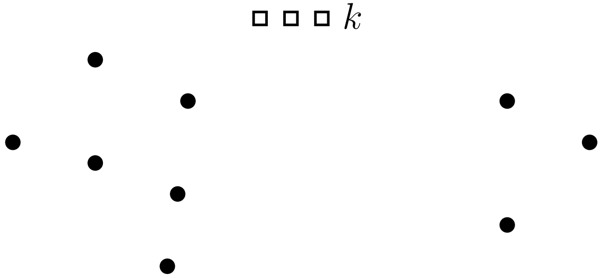
Robot patrolling problem

k robots with maximum unit speed collectively patrol *n* sites in a metric space.



Robot patrolling problem

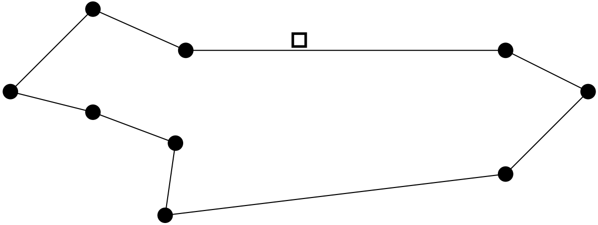
k robots with maximum unit speed collectively patrol n sites in a metric space.



Patrol schedules are infinite sequences.
Goal: minimize the maximum time duration (latency) between consecutive visits to any site.

Robot patrolling problem

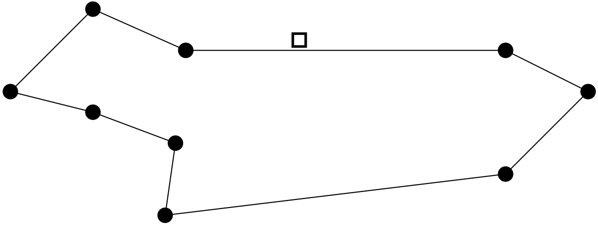
$k = 1$: travelling salesman problem. NP-hard.



Every site is visited every $L = |TSP|$ unit time.

Robot patrolling problem

$k = 1$: travelling salesman problem. NP-hard.



Every site is visited every $L = |TSP|$ unit time.

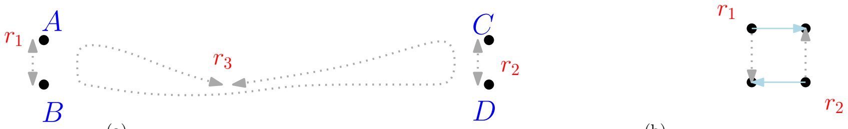
- General metric setting: $3/2$-approximation (Christofides'76), $3/2 - \delta$, $\delta > 10^{-36}$ (Karlin, Klein, Gharan'21).
- Euclidean setting: $(1 + \varepsilon)$-approximation (Arora'98, Mitchell'99).

Challenges

$k \geq 2$: there can be optimal solutions that are chaotic.

Challenges

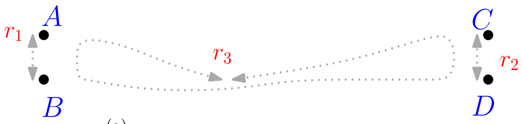
$k \geq 2$: there can be optimal solutions that are chaotic.



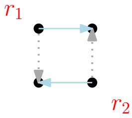
(a) (b)

Challenges

$k \geq 2$: there can be optimal solutions that are chaotic.



(a)



(b)

Not clear if the problem (whether opt latency $< L$) is decidable if distances are not integers.

What about approximations?

[ABBGLNRSWY'20] If there is an α-approximation to

- k-path cover problem: find k paths to cover n sites with min max path length.
- min-max k-tree cover problem: find k trees to cover n sites with min max tree weight.
- min-max k-cycle cover problem: find k cycles to cover n sites with min max cycle length.

there is a 2α-approximation to k-robot patrol problem.

What about approximations?

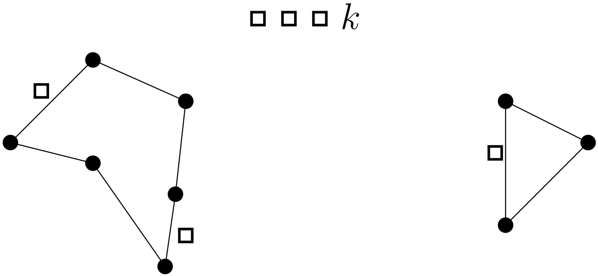
- k-path cover problem: 4 [Arkin, Hassin, Levin'06]
- min-max k-tree cover problem: **8/3** [Xu, Liang, Lin'13]
- min-max k-cycle cover problem: $16/3$ [Xu, Liang, Lin'13]

Best known approximation to k-robot patrol: **16/3**.

Cyclic solutions

We focus on cyclic solutions:

- n sites partitioned into $\ell \leq k$ groups $P_1, P_2, \cdots P_\ell$.
- Group P_i is allocated one or more robots, evenly spread along $TSP(P_i)$.



Main Results

1. An optimal cyclic solution is a $2(1 - 1/k)$ approximation to the optimal overall solution.

Main Results

1. An optimal cyclic solution is a $2(1 - 1/k)$ approximation to the optimal overall solution.

- When $k = 2$, the optimal cyclic solution is optimal overall.

Main Results

1. An optimal cyclic solution is a $2(1 - 1/k)$ approximation to the optimal overall solution.

- When $k = 2$, the optimal cyclic solution is optimal overall.

2. Approximating the optimal cyclic solution $\Rightarrow$ approximating TSP on some input, with

- an extra $1 + \varepsilon$ approximation factor.
- an extra $O((k/\varepsilon)^k)$ factor in running time.

Main Results

1. An optimal cyclic solution is a $2(1 - 1/k)$ approximation to the optimal overall solution.

- When $k = 2$, the optimal cyclic solution is optimal overall.

2. Approximating the optimal cyclic solution $\Rightarrow$ approximating TSP on some input, with

- an extra $1 + \varepsilon$ approximation factor.
- an extra $O((k/\varepsilon)^k)$ factor in running time.

3. Solving multi-robot patrol problem:

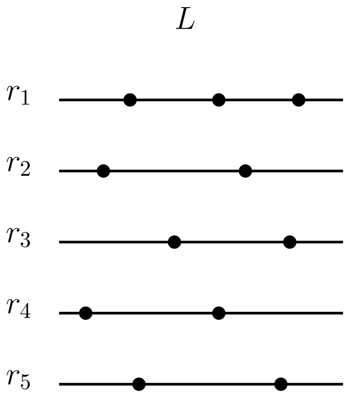
- General metric setting: $3 - 3/k + \varepsilon$ approximation.
- Euclidean setting: $2 - 2/k + \varepsilon$ approximation.

Outline

- Robot patrolling problem
- **Cyclic solutions**
- Find a $(1 + \varepsilon)$-approximte cyclic solution
- Open problems

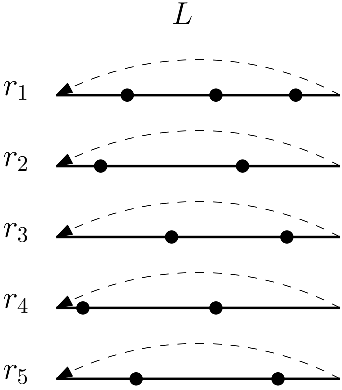
Periodic schedule w/ 2-approximation

Take an OPT solution w/ latency $= L$. All n sites must be visited during a time window L.



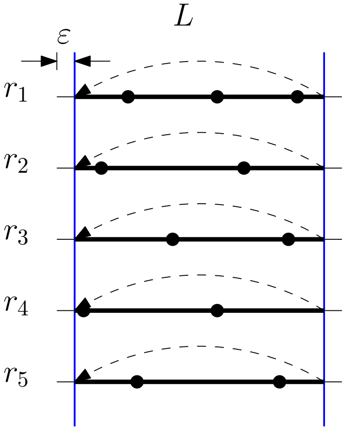
Periodic schedule w/ 2-approximation

Take an OPT solution w/ latency $= L$. All n sites must be visited during the time window L. $\Rightarrow$ Wrap around, we get a solution with latency $\leq 2L$.



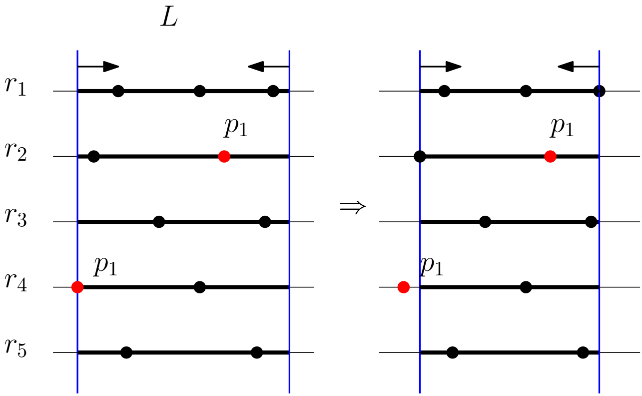
Improve 2-approximation?

If we can shrink the window by ε on both sides yet still cover all sites, we get latency $\leq 2(1 - 2\varepsilon) \cdot L$.



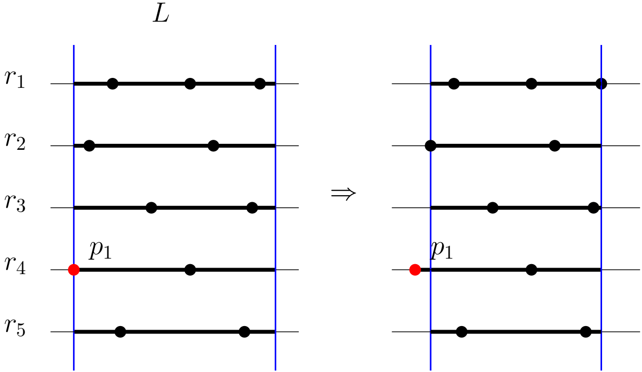
Shrinking process

Sweep from both ends inward, until we meet a site (say p_1).
If p_1 is still covered, ignore & keep going.



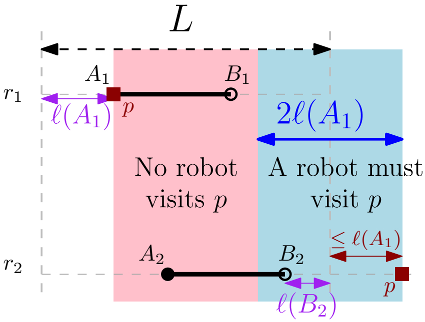
Shrinking process

Sweep from both ends inward, until we meet a site (say p_1).
If p_1 is "critical", freeze the endpoint of r_4 at p_1 .



After shrinking process

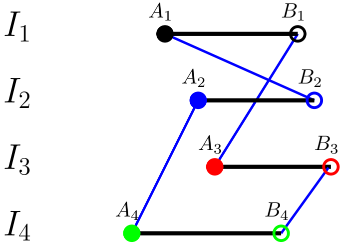
Frozen segments: left endpoints A, right endpoints B.
Robot r_1 at A_1 visits site p with sweep distance $\ell(A_1)$.



Shortcut: r_2 moves from B_2 to visit $f_1(A_1)$, pay cost $\leq \ell(A_1) + \ell(B_2)$.

How to use the shortcut edges? An easy case

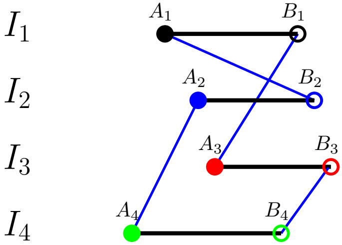
Add a matching of shortcut edges to the frozen segments ⇒ a set of bichromatic cycles ⇒ a cyclic solution.



No increase in latency: cost paid no greater than saving.

How to use the shortcut edges? An easy case

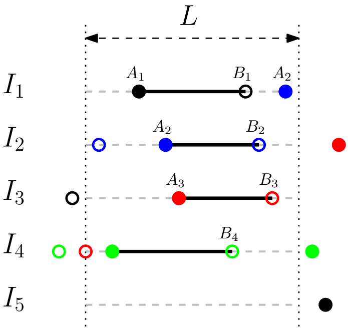
Add a matching of shortcut edges to the frozen segments ⇒ a set of bichromatic cycles ⇒ a cyclic solution.

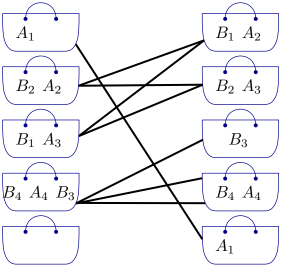


No increase in latency: cost paid no greater than saving.
But, can we find a matching of shortcut edges?

Shortcut graph and bag graph

Bag graph: capture potential shortcuts.

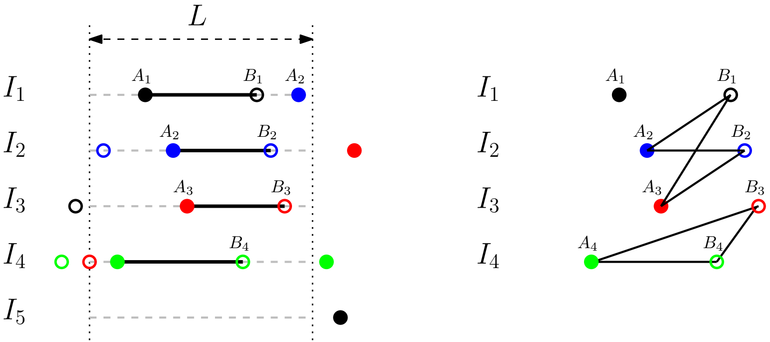




Place an edge between two bags if they share a common endpoint label.

Shortcut graph and bag graph

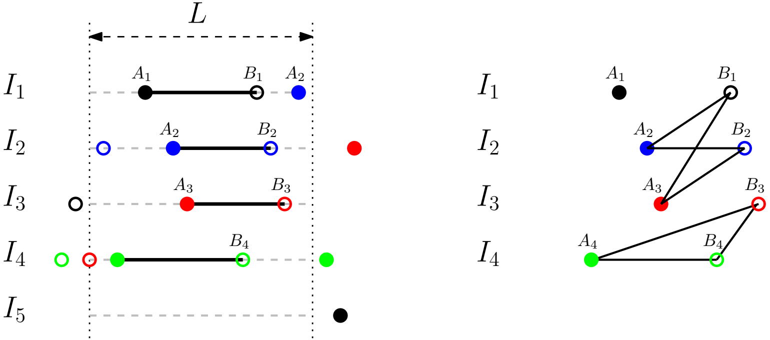
Shortcut graph: connect endpoints with potential shortcuts.



Shortcut graph is isomorphic to the line graph of the bag graph.

Shortcut graph and bag graph

Shortcut graph: connect endpoints with potential shortcuts.

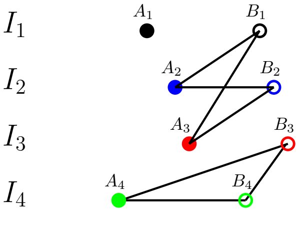


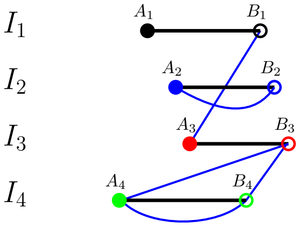
Shortcut graph is isomorphic to the line graph of the bag graph.
[Sumner'74] The line graph of a connected graph with an even number of edges has a perfect matching.

Modify into a cyclic solution: Add shortcuts

Case study on connected components of the bag graph:

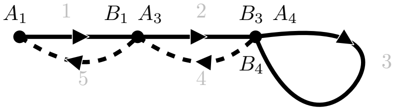
- even # vertices ⇒ a perfect matching.
- odd # vertices, no empty segment ⇒ a matching + a triangle.
- odd # vertices, w/ empty segment ⇒ a matching + a vertex.

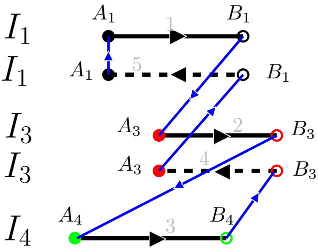




Modify into a cyclic solution: Eulerize

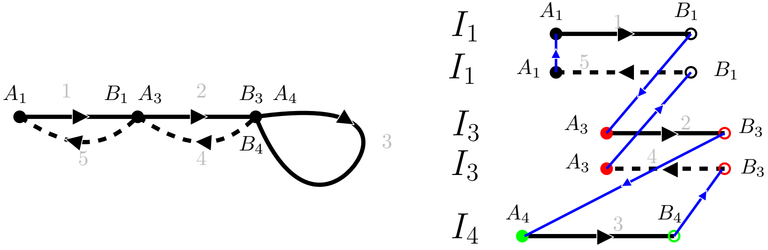
Duplicate certain frozen edges to make the graph Eulerian.





Modify into a cyclic solution: Eulerize

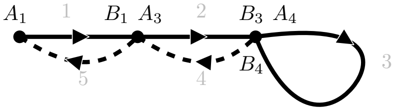
Duplicate certain frozen edges to make the graph Eulerian.

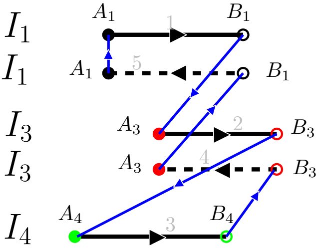


Issue: pay extra cost for duplicated black edges.

Modify into a cyclic solution: Eulerize

Duplicate certain frozen edges to make the graph Eulerian.





Issue: pay extra cost for duplicated black edges.
Analyze # duplicated edges & # "useless" robots.
Max lantency $\leq 2(1 - 1/k)L$.

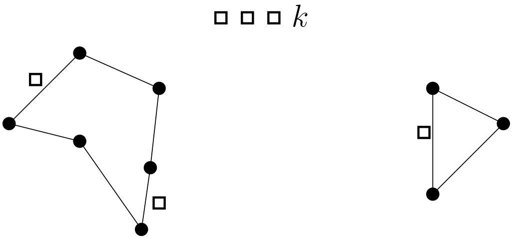
Outline

- Robot patrolling problem
- Cyclic solutions
- **Find a** $(1+\varepsilon)$**-approximate cyclic solution**
- Open problems

An optimal cyclic solution

An optimal cyclic solution:

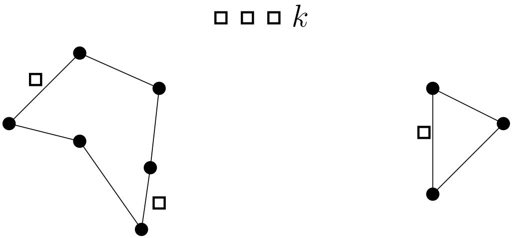
- Paritioning the sites into clusters $\Pi = \{P_1, P_2, \cdots, P_t\}$, $t \leq k$.
- Assign k robots to these clusters: $k_1 + k_2 + \cdots + k_t = k$.
- For each cluster, we place robots evenly along $TSP(P_i)$.



An optimal cyclic solution

An optimal cyclic solution:

- Paritioning the sites into clusters $\Pi = \{P_1, P_2, \cdots, P_t\}$, $t \leq k$.
- Assign k robots to these clusters: $k_1 + k_2 + \cdots + k_t = k$.
- For each cluster, we place robots evenly along $TSP(P_i)$.



Issue 1: how to find the partition Π?

Issue 2: how to assign robots to a partition Π?

How to assign robots to a partition?

Assume a γ-approximate TSP algorithm *tsp*.

How to assign robots to a partition?

Assume a γ-approximate TSP algorithm *tsp*.

- Assign one robot to each cluster. $k_i = 1$, $\forall i$.
- Max latency $\max_i tsp(P_i)/k_i$.
- Iteratively assign the next robot to the cluster w. the largest latency.

How to assign robots to a partition?

Assume a γ-approximate TSP algorithm *tsp*.

- Assign one robot to each cluster. $k_i = 1$, $\forall i$.
- Max latency $\max_i tsp(P_i)/k_i$.
- Iteratively assign the next robot to the cluster w. the largest latency.

This gives a γ-approximation to optimal solution with partitioning Π.

- General metric space: $\gamma = 1.5 - \delta$, $\delta > 10^{-36}$. [Karlin et.al. '21]
- Euclidean setting: $\gamma = 1 + \varepsilon$, [Arora'98, Mitchell'99]

How to assign robots to a partition?

Assume a γ-approximate TSP algorithm *tsp*.

- Assign one robot to each cluster. $k_i = 1$, $\forall i$.
- Max latency $\max_i tsp(P_i)/k_i$.
- Iteratively assign the next robot to the cluster w. the largest latency.

This gives a γ-approximation to optimal solution with partitioning Π.

- General metric space: $\gamma = 1.5 - \delta$, $\delta > 10^{-36}$. [Karlin et.al. '21]
- Euclidean setting: $\gamma = 1 + \varepsilon$, [Arora'98, Mitchell'99]

Issue 1: how to find the partition Π?

Issue 2: how to assign robots to a partition Π? ✓

A ‘well-separated’ $(1+\varepsilon)$-approximate cyclic solution

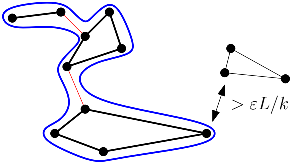
OPT cyclic solution has latency L.

- There exists a $(1+\varepsilon)$-approximate cyclic solution with a partition Π s.t. the min distance between clusters is $\geq \varepsilon L/k$.

A 'well-separated' $(1+\varepsilon)$-approximate cyclic solution

OPT cyclic solution has latency L.

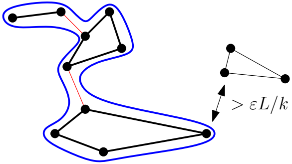
- There exists a $(1+\varepsilon)$-approximate cyclic solution with a partition Π s.t. the min distance between clusters is $\geq \varepsilon L/k$.



A 'well-separated' $(1+\varepsilon)$-approximate cyclic solution

OPT cyclic solution has latency L.

- There exists a $(1+\varepsilon)$-approximate cyclic solution with a partition Π s.t. the min distance between clusters is $\geq \varepsilon L/k$.

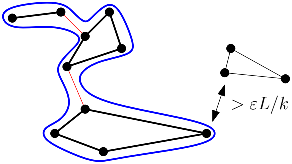


Add all short edges to an opt cyclic solution.

A ‘well-separated’ $(1+\varepsilon)$-approximate cyclic solution

OPT cyclic solution has latency L.

- There exists a $(1+\varepsilon)$-approximate cyclic solution with a partition Π s.t. the min distance between clusters is $\geq \varepsilon L/k$.

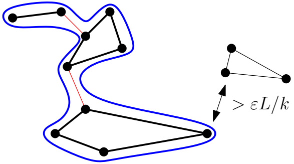


Add all short edges to an opt cyclic solution.
For each component keep the “minimum spanning” edges.

A 'well-separated' $(1+\varepsilon)$-approximate cyclic solution

OPT cyclic solution has latency L.

- There exists a $(1+\varepsilon)$-approximate cyclic solution with a partition Π s.t. the min distance between clusters is $\geq \varepsilon L/k$.

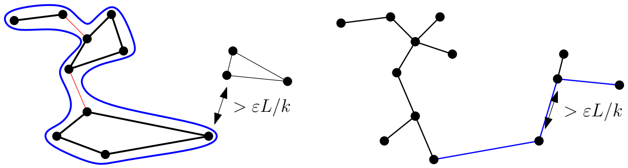


Add all short edges to an opt cyclic solution.
For each component keep the "minimum spanning" edges.
Turn each connected component to a new cycle.

Find a $(1+\varepsilon)$-approximate cyclic solution?

Start with the MST T.

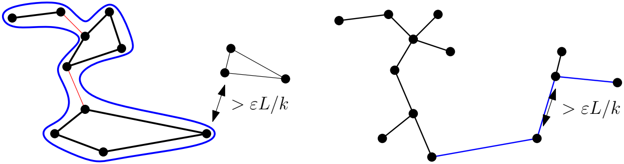
- MST has $\leq k(1 + k/\varepsilon)$ long edges ($\geq \varepsilon L/k$).
- Enumerate a subset of at most k long edges to remove.



Find a $(1+\varepsilon)$-approximate cyclic solution?

Start with the MST T.

- MST has $\leq k(1+k/\varepsilon)$ long edges ($\geq \varepsilon L/k$).
- Enumerate a subset of at most k long edges to remove.



Extra factor of $O((k/\varepsilon)^k)$ time to find the correct partition.

Open problems

- Improve approximation factor of $2(1 - 1/k)$ by the optimal cyclic solution.

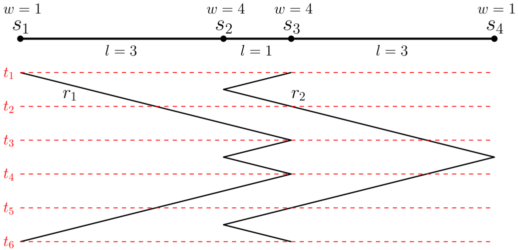
Open problems

- Improve approximation factor of $2(1 - 1/k)$ by the optimal cyclic solution.
- Conjecture: the optimal cyclic solution is overall optimal.

More open problems

Weighted version: minimize $\max_i w_i L_i$.

- $k = 1$, $O(\log n)$-approximation. [Alamdari et.al. '14]
- $k \geq 2$, $O(k^2 \log \frac{w_{\max}}{w_{\min}})$-approximation. [Afshani et.al. '20].



Even in 1D, optimal solution do not use disjoint cycles. Solutions with disjoint cycles are arbitrarily worse.

Questions and comments

- On Cyclic Solutions to the Min-Max Latency Multi-Robot Patrolling Problem, SoCG'2022.
- Approximation Algorithms for Multi-Robot Patrol-Scheduling with Min-Max Latency, WAFR'2020.