Multiple Heterogeneous Data Ferry Trajectory Planning in Wireless Sensor Networks

Lirong Xue[1], Donghyun Kim[2], Yuqing Zhu[3]
Deying Li[1], Wei Wang[4], and Alade O. Tokuta[1]

[1]Renmin University of China, Beijing, China
[2]North Carolina Central University, Durham, USA
[3]The University of Texas at Dallas, Richardson, USA
[4]Xi'an Jiaotong University, Xi'an, China

Presented on May 1, 2014 at The 33rd Annual IEEE International Conference on Computer Communications (INFOCOM 2014), Toronto, Canada.

Agenda

- **Introduction**
- Preliminaries
- Problem Definitions & Algorithm – k-ATSPN
- Problem Definitions & Algorithm – k-ITSPN
- Simulation Results
- Conclusion

Networked Robotic Vehicles

- move on the ground, in the air, and undersea
  - such as mobile aerial vehicles and drones

- superior physical capabilities
  - wireless communication
  - computing resources

Image source: [1]

- execute challenging tasks by interacting with surrounding infrastructure

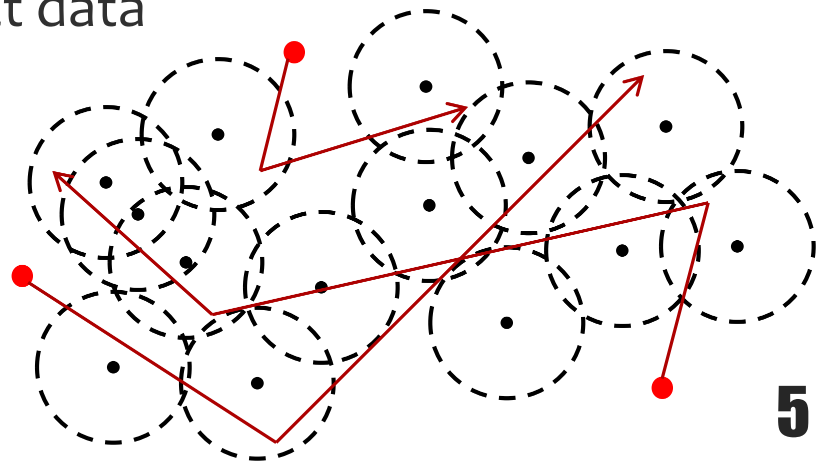
Multiple Networked Robotic Vehicle System

- accomplish a complicated task beyond the capability of a single robot through collaboration
- coordination is a crucial issue
- **trajectory planning** is important
  - related to latency, energy-efficiency
  - many trajectory planning optimizations are NP-hard (e.g. Traveling salesman problem)

Image source: [2]

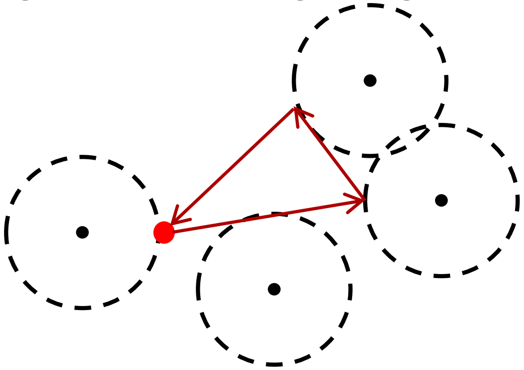
Application Scenario 1

- urgent data collection from stationary sensor nodes
  - e.g. survival search and damage assessment after an earthquake
- multiple fully-maneuverable mobile nodes
- limited communication range for ground sensor nodes
  - use mobile node to collect data
- suffers from huge latency
- trajectory can be
  - rooted tours
  - rooted paths



Application Scenario 2

- routine data collection from stationary sensors over long time period
  - e.g. scientific data collection of wildlife over a vast area
- multiple fully-maneuverable mobile nodes
  - establish a tour and collect data from ground sensors repeatedly along the tour (more like a un-rooted tour)
- limited communication range of ground sensor nodes
- suffers from huge data refreshment rate



Our Problems

1. k-Asynchronous Traveling Salesman Problem with Neighborhood (k-ATSPN) and Path Cover Problem with Neighborhood (k-PCPN)

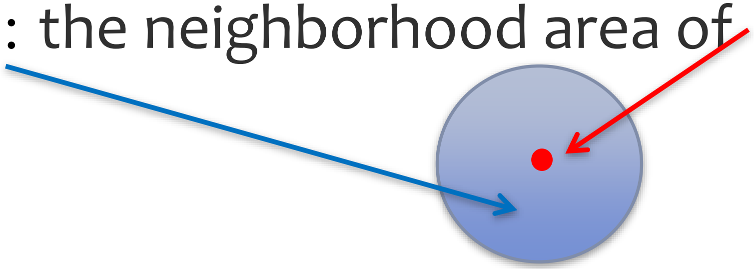
2. k-Inhomogeneous Traveling Salesman Problem with Neighborhood (k-ISTPN) and Weighted k-Tree Cover Problem (Wk-TCP)

Agenda

- Introduction
- **Preliminaries**
- Problem Definitions & Algorithm – k-ATSPN
- Problem Definitions & Algorithm – k-ITSPN
- Simulation Results
- Conclusion

Notations

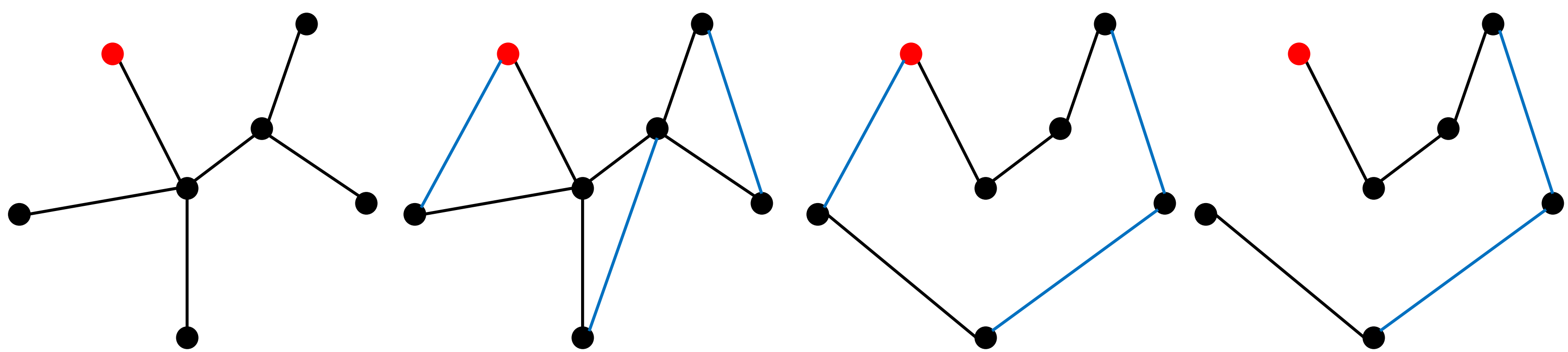
- $V = \{v_1, v_2, \ldots, v_n\}$: a set of n wireless sensor nodes
- $R = \{r_1, r_2, \ldots, r_k\}$: a set of k mobile nodes (roots)
- $N(v_i)$: the neighborhood area of v_i



- $G = (V, E)$: a graph with a vertex set V and edge set E
- $Len(v_i, v_j)$: the length of an edge between v_i, v_j
- $Len(G) = \sum_{(v_i, v_j) \in E(G)} Len(v_i, v_j)$

Traveling Salesman Problem

- Christofides' 1.5-approximation for TSP [3]
  - use an MST to find a tour
  - a path can be obtained from the tour (no cost increase)



MST

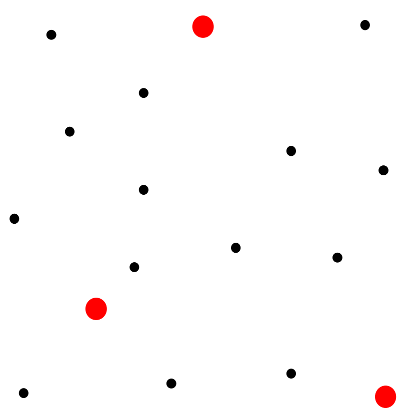
add perfect matching edges among odd degree nodes

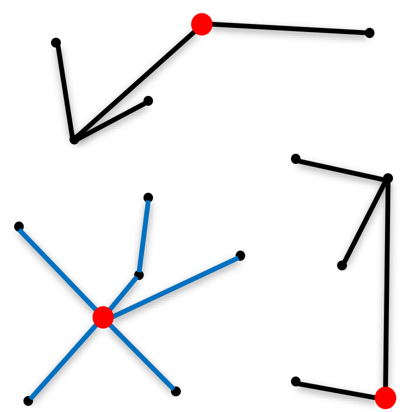
apply triangular inequality to have a tour

remove an edge to get a path

k-rooted Tree Cover Problem

- given a set V of nodes $\{v_1, v_2, \ldots, v_n\}$ and a set R of k roots $\{r_1, r_2, \ldots, r_k\}$, find k rooted trees such that
  - each node is in some tree
  - the total edge weight of the heaviest tree is minimum

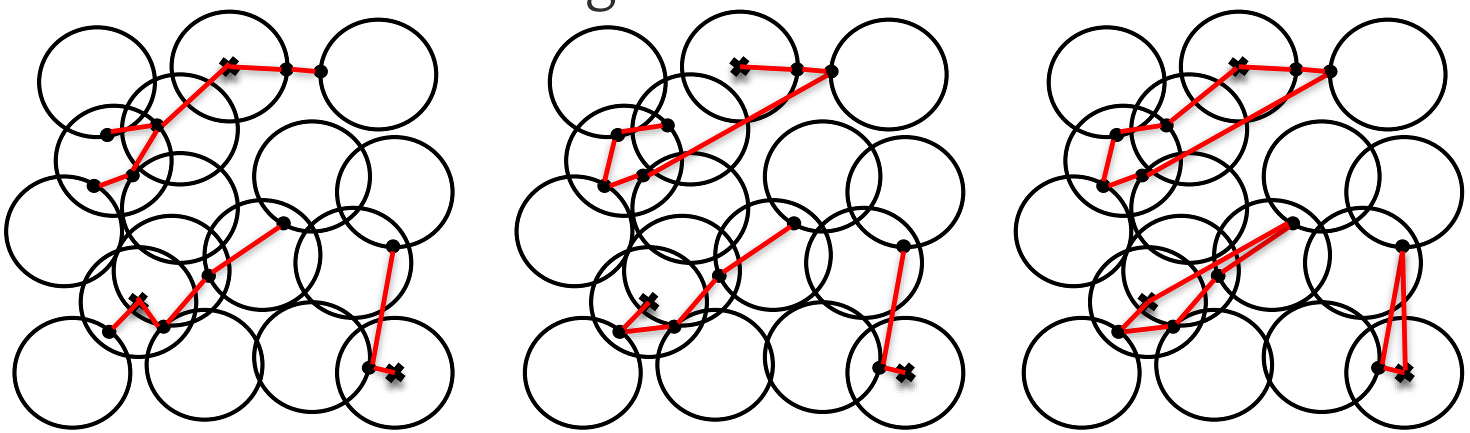




- Even et al. proposed (4+ε)-approximation algorithm [4]

k-rooted Tree Cover Problem with Neighborhood (k-TCPN) [5]

- given a set V of nodes $\{v_1, v_2, \ldots, v_n\}$ and a set R of k roots (mobile nodes) $\{r_1, r_2, \ldots, r_k\}$, find k trees such that
  - each tree is originated at one root
  - the neighborhood of each node is visited by some tree
  - the cost of the longest tree becomes minimum



k-trees k-paths k-tours

Constant Approximation for k-TCPN [5]

1. ignore neighborhood of each node and apply the $(4+\epsilon)$ -approximation algorithm for k-rooted tree cover problem
   - partition the graph with k subsets
   - each partition has a root and a subset of nodes

2. for each subset
   - compute an MST of
     - the root, and
     - the neighborhoods of the nodes in the subset

approximation factor multiplier: β

approximation ratio: $(4+\varepsilon)\ \beta$

3. output k-rooted trees (a feasible solution of k-TSPN)

k-rooted Path Cover Problem with Neighborhood (k-PCPN) [5]

- given a set V of nodes $\{v_1, v_2, \ldots, v_n\}$ and a set R of k roots $\{r_1, r_2, \ldots, r_k\}$, find k paths such that
  - each path starts from one root
  - the neighborhood of each node is visited by some path
  - the cost of the longest path becomes minimum

- convert a solution of k-TCPN to a solution of k-PCPN
  - use Christofides' 1.5-approximation for TSP
  - approximation ratio: $1.5\,(4+\varepsilon)\,\beta$

k-traveling Salesperson Problem with Neighborhood (k-TSPN) [5]

- given a set V of nodes $\{v_1, v_2, \ldots, v_n\}$ and a set R of k roots $\{r_1, r_2, \ldots, r_k\}$, find k tours such that
  - each tour is originated at one root
  - the neighborhood of each node is visited by some tour
  - the cost of the longest tour becomes minimum

- convert a solution of k-TCPN to a solution of k-TSPN
  - use Christofides' 1.5-approximation for TSP
  - approximation ratio: 1.5 (4+ε) β

Agenda

- Introduction
- Preliminaries
- **Problem Definitions & Algorithm – *k*-ATSPN**
- Problem Definitions & Algorithm – *k*-ITSPN
- Simulation Results
- Conclusion

k-Asynchronous TSPN and PCPN (k-ATSPN and k-APCPN)

- related to k-TSPN and k-PCPN
- definition
  - each root (mobile node) is mobilized (available) after $\{t_1, t_2, \ldots, t_k\}$ time units
  - given a set V of nodes $\{v_1, v_2, \ldots, v_n\}$ and a set R of k roots $\{r_1, r_2, \ldots, r_k\}$, find k tours (or paths) such that
    - each tour starts from one root
    - the speed of each mobile node is uniformly s
    - the neighborhood of each node v is visited by some tour
    - one time data collection latency from every sensor node is minimized

k-Asynchronous TSPN and PCPN (*k*-ATSPN and *k*-APCPN) – cont

- objective
  - *k*-ATSPN and *k*-APCPN

slow starting node needs more time to finish the assigned task

$$\min \max_{1 \le i \le k} [\frac{Len(U_i)}{s} + t_i]$$

cost is in terms of unit time (latency)

uniform speed: *s*

Constant Approximation for k-ATSPN and k-APCPN

- induction to k-TSPN (or k-PCPN) :
  - for nodes ($V = \{v_1, \ldots, v_n\} \cup \{r_1, \ldots, r_k\}$) and delay ($\{t_1, \ldots, t_k\}$)
    - construct auxiliary graph $\hat{G} = (V, \hat{E}, C_E)$
    - for each pair (v_i, v_j), add edge to $\hat{E}$, and the cost is the Euclidean distance $Euc(v_i, v_j)$
    - for each pair (v_i, r_j), add edge to $\hat{E}$, and the cost the their Euclidean distance add their delay (converted to distance):

$$Euc(v_i, r_j) + Delay(r_j) * Speed$$

  - then the delays are incorporated as distance between the roots and other nodes
  - this induction is true since each root is visited only once
- apply Kim's constant approximation algorithm for k-TSPN (or k-PCPN) [5] on $\hat{G}$

Agenda

- Introduction
- Preliminaries
- Problem Definitions & Algorithm – k-ATSPN
- **Problem Definitions & Algorithm – k-ITSPN**
- Simulation Results
- Conclusion

Weighted k Trees Cover Problem ($\mathrm{W}k$-TCP) – problem definition

- related to k-ISTPN
- definition
  - given an edge-weighted graph $G = (V, E, w_E)$
  - given k different weights $w_1, w_2, \ldots, w_k$
  - find k trees $T = \{T_1, \ldots, T_k\}$ spanning G, such that:
    - each node in V is covered by some tree
    - cost becomes minimum:

$$\min[Cost(T)] = \min \max_{1 \le i \le k} [\frac{Len(T_i)}{w_i}]$$

Weighted k Trees Cover Problem (Wk-TCP) – problem definition

- objective:

$$\min[Cost(T)] = \min \max_{1\le i\le k}\left[\frac{Len(T_i)}{w_i}\right]$$

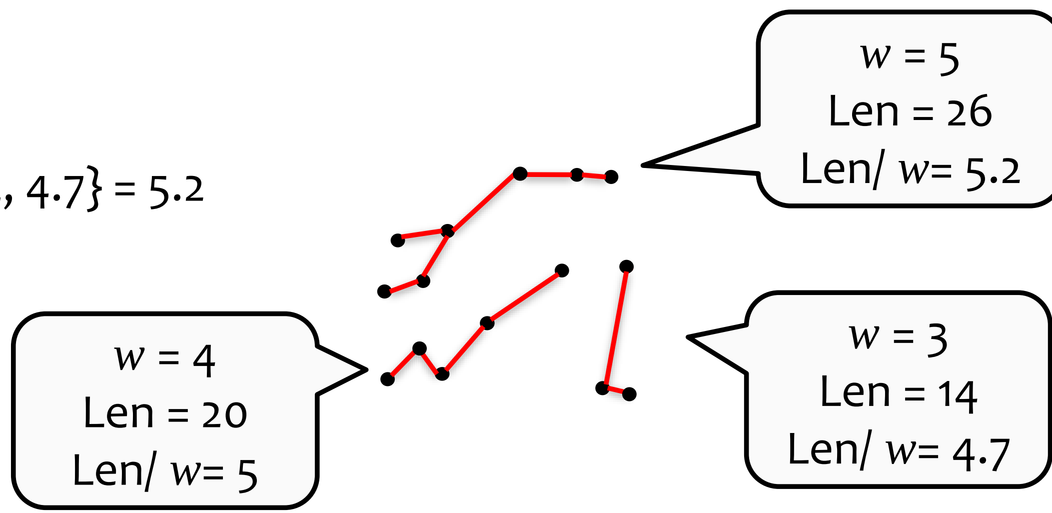
cost of all edges in a tree, divided by weight

aim to minimize the cost of T

T's cost is the cost of the its heaviest tree

eg. Wk-TCP

- $k = 3$
- weights (w) = {3,4,5}
- total cost = max{5, 5.2, 4.7} = 5.2



2α Approximation for Wk-TCP

- $\alpha = \frac{k-1+w_1+w_2+\ldots+w_k}{w_1+w_2+\ldots+w_k}$

- a generalization of Even's k-rooted tree cover algorithm [4]
- it is NP-hard: special case k-TCP with $w_i = w_j$ for each i, j pairs, is NP-hard

We do not consider a team of $200 UAV and a fighter jet!

- assumptions (generalized):
  - variance between normalized weights is "non-extreme":

$$1/\alpha \leq w_1 \leq w_2 \leq \cdots \leq w_k = 1$$

- 2α – approximation ratio

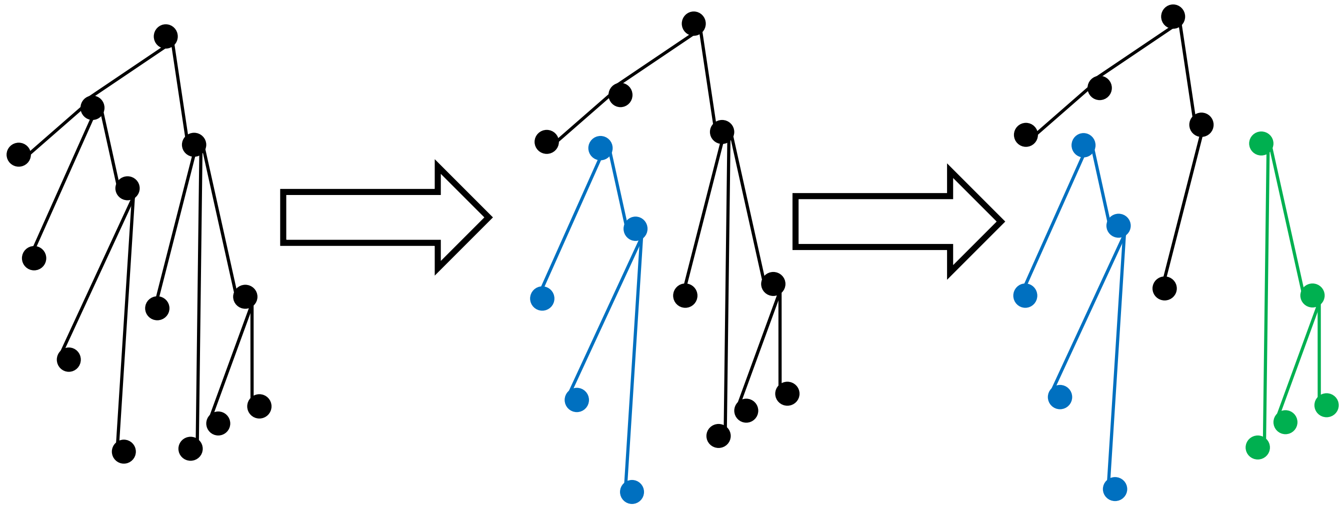
2α Approximation for Wk-TCP

1. first [edge decomposition], decompose the MST of G into k trees, given b (an arbitrary value)
   - suppose the optimal cost of Wk-TCP is b^*
   - cost of the decomposed trees is no greater than $2\alpha b$
   - for $b \geq b^*$, such procedure can be done

2. then, use binary search to find the optimal b (i.e. b_0)
   - $b_0 \leq b^*$, therefore $Cost(T) \leq 2\alpha b^*$
   - approximation ratio is at most 2α

2α Approximation for Wk-TCP: edge decomposition

- procedure:
  - for a given value b
  - for each weight w_i, decompose a tree from the MST of G
  - therefore, get k trees $\{T_1, \dots, T_k\}$
  - if can't decompose, return "fail", else return "succ"



2α Approximation for Wk-TCP: edge decomposition

- if success:

$$\alpha b w_i \leq Len(T_i) \leq 2\alpha b w_i$$

- as long as $b \leq b^*$, decomposition will success (Lemma 2)
    - the optimal cost of Wk-TCP is b^*

- when success, the cost of the result tree: (by definition)

$$Cost(T) \leq 2\alpha b$$

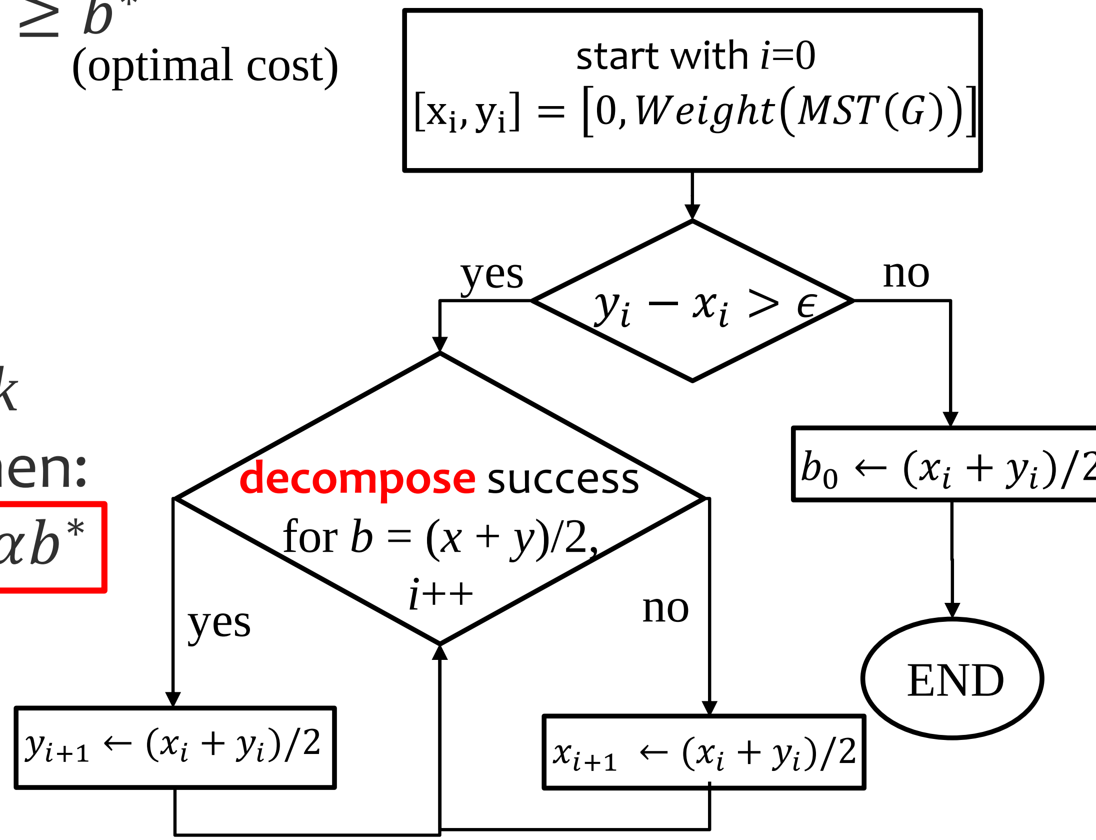
2α Approximation for Wk-TCP: binary search

- return "succ" when $b \geq b^*$ (optimal cost)
- so:

$$b_0 = \lim_{i \to \infty} x_i \leq b^*$$

- set b as b_0 to get the k decomposed trees, then:

$$Cost(T) \leq 2\alpha b_0 \leq 2\alpha b^*$$



k-Inhomogeneous TSPN (k-ITSPN)

- related to k-TSPN and k-PCPN
- definition
  - given edge-weighted graph $G = (V, E, c_E)$
  - given a set S of k distinct speeds $S = \{s_1, s_2, \ldots, s_k\}$, find k tours such that
  - the neighborhood of each node is visited by some tour
  - the time to complete one round trip is minimized (related to worst case data refreshment rate)
- objective

$$\min \max_{1 \le i \le k} [\frac{Len(U_i)}{s_i}]$$

k-Inhomogeneous TSPN (k-ITSPN)

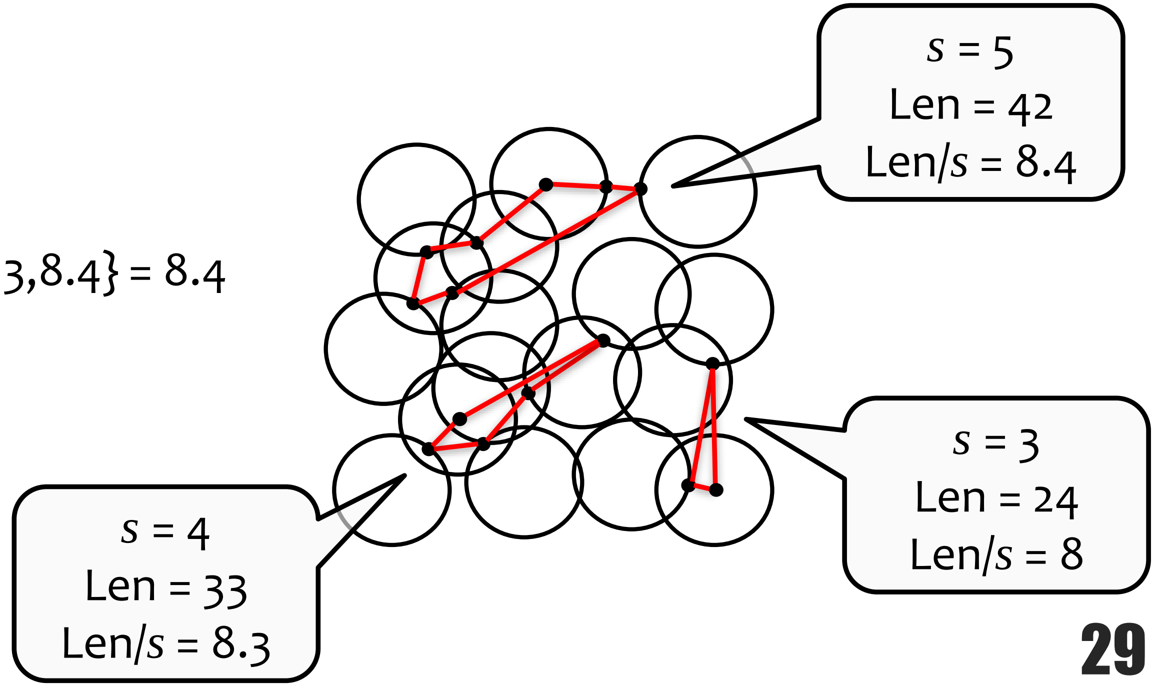
- objective
  - k-ITSPN

$$\min \max_{1\le i\le k}\left[\frac{Len(U_i)}{s_i}\right]$$

faster node finish task earlier

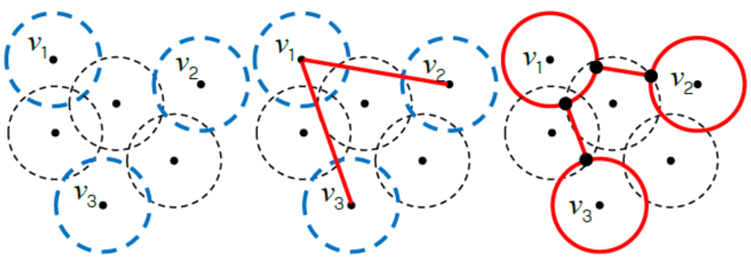
eg. k-ISTPN

- $k = 3$
- speed = {3,4,5}
- total cost = max{8,8.3,8.4} = 8.4



An Approximation for k-ITSPN

1. apply two coloring strategy to obtain a pairwise disjoint subset of disks of V
2. apply Wk-TCP's algorithm to get k trees
3. for each tree, convert each tree into neighborhood tour (using Kim et. al.'s γ-approximation [5])



- approximation ratio is at most:

$$(2\alpha) \cdot (1.5) \cdot \gamma$$

Wk-TCP

convert to tour

k-TSPN

Agenda

- Introduction
- Preliminaries
- Problem Definitions & Algorithm – k-ATSPN
- Problem Definitions & Algorithm – k-ITSPN
- **Simulation Results**
- Conclusion

Simulation Results

- performed over 30×30 virtual space.
- the number of nodes (n) varies to 30, 40, … , 90.
- the number of weights (k) varies to 3, 6 and 9.
- for each parameter setting, we created 100 instances and compute the average.
- compare Wk-TCP with MST without (k -1) heaviest edges
- comparison is done in terms of Tree (also Tour and Path after the conversion)
- metrics

$$\text{ratio} = \frac{\text{cost of output of W}k - \text{TCP}}{\text{cost of MST without } k - 1 \text{ heaviest edges } \div \text{sum of weights}}$$

Simulation Results – cont’

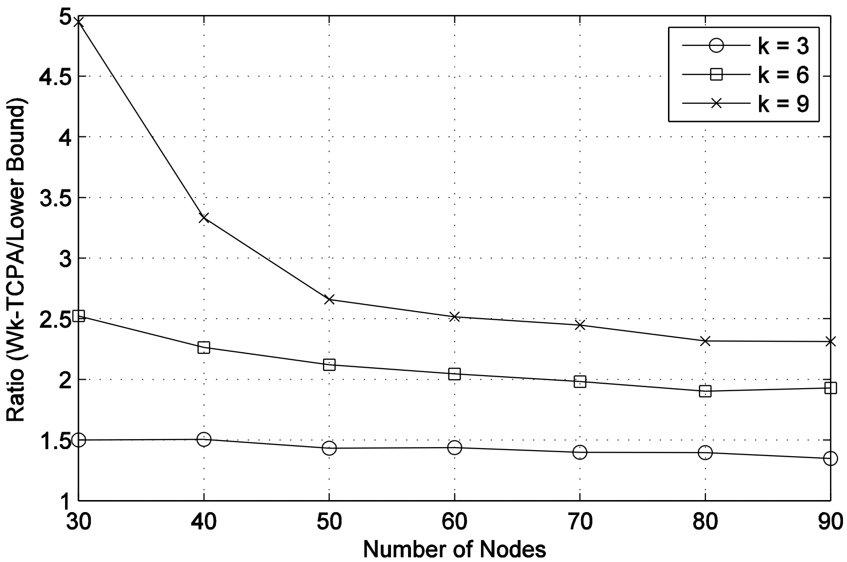


fig. 1. effect of k and n with $w_i \in (0, 5]$

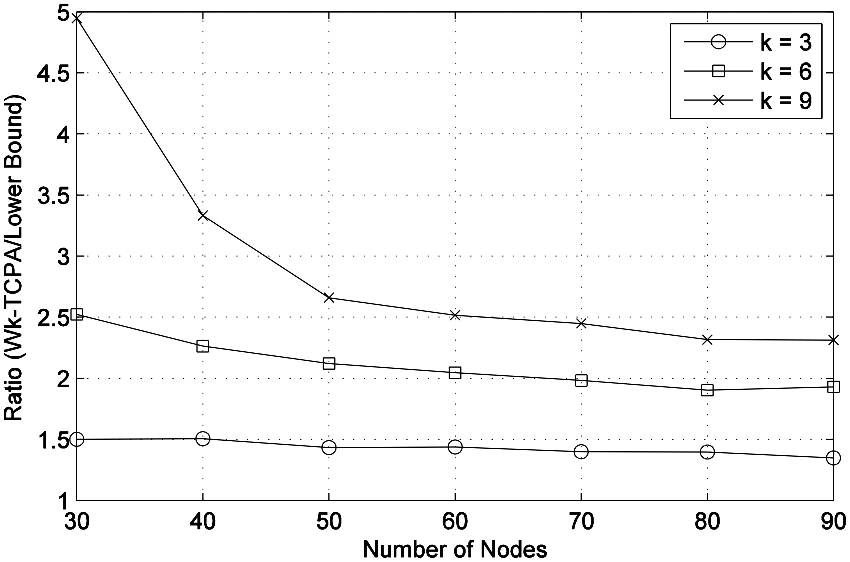


fig. 2. effect of k and n with $w_i \in (0, 10]$

ratio
- lower ratio for more nodes
- ratio stabilizes as node increases

Simulation Results – cont'

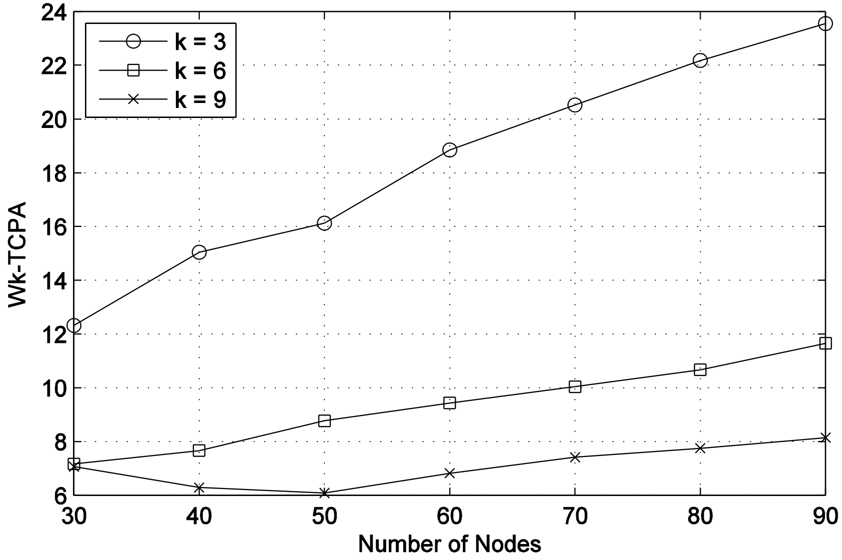


fig. 3. performance of Wk-TCP with $w_i \in (0, 5]$

performance
$k = 3, 6, 9$

- cost increases for more nodes
- almost linear

Simulation Results – cont'

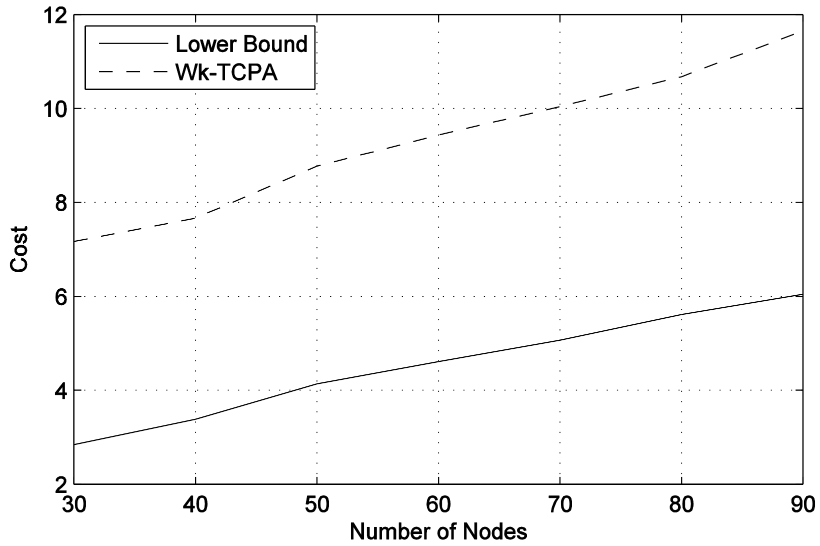


fig. 4. Comparison between Wk-TCP and the lower bound with $w_i \in (0, 10]$

performance (compare)

- both cost increases for more nodes
- both almost linear

Agenda

- Introduction
- Preliminaries
- Problem Definitions & Algorithm –k-ATSPN
- Problem Definitions & Algorithm – k-ITSPN
- Simulation Results
- **Conclusion**

Conclusion

- introduce new problems
  - k-Asynchronous TSPN and PCPN (k-ATSPN & k-APCPN)
  - k-Inhomogeneous TSPN (k-ITSPN)
    - Weighted k-Tree Cover Problem

- approximation algorithms for
  - k-Asynchronous TSPN and PCPN (k-ATSPN & k-APCPN)
  - k-Inhomogeneous TSPN (k-ITSPN)
    - Weighted k-Tree Cover Problem

References

[1] http://www-bgr-com.vimg.net/wp-content/uploads/2012/01/parrot_ar.drone_12-645x336.jpg

[2] http://www.ros.org/news/resources/2010/parrot_ardrone3.jpg

[3] N. Christofides, "Worst-case Analysis of a New Heuristic for the Travelling Salesman Problem," Report 388, Graduate School of Industrial Administration, CMU, 1976.

[4] G. Even, N. Garg, J. Konemann, R. Ravi, and A. Sinha, "Min-Max Tree Covers of Graphs," Operations Research Letters, vol. 32, issue 4, pp.309-315, 2004.

[5] D. Kim, R.N. Uma, B.H. Abay, W. Wu, W. Wang, and A.O. Tokuta, "Minimum Latency Multiple Data MULE Trajectory Planning in Wireless Sensor Networks," IEEE Transactions on Mobile Computing (TMC), vol. 13, no. 4, pp. 838-851, April 2014.

Thank you
Question?