Minimizing Data Collection Latency in Wireless Sensor Network with Multiple Mobile Elements

Donghyun Kim¹, Baraki H. Abay¹, R.N. Uma¹
Weili Wu², Wei Wang³, and Alade O. Tokuta¹

¹North Carolina Central University, Durham, NC
²The University of Texas at Dallas, Richardson, TX
³Xi’an Jiaotong University, China


http://www.cs.nccu.edu/~advnet/
Agenda

• Motivations

• Problem Definitions

• Preliminaries

• Main Results – Approximation Algorithms

• Simulation Results

• Conclusion

http://www.cs.nccu.edu/~advnet/
Motivations

• Data Collection using Multiple SenCars in Wireless Sensor Networks.
  • Sencar [5]: a fully controllerable mobile sink.
    • Extend sensor network lifetime.
    • Collect data from disconnected sensor network.
  • Suffers from huge latency.

• What is the best trajectories to minimize the latency?
Motivations – cont’

• Exploration of unknown areas using multiple mobile robot agents.
  • Each mobile robot collects the knowledge of an unknown area.

• How to minimize the time to collect the knowledge of all of points of interest using multiple mobile agents?

http://www.cs.nccu.edu/~advnet/
Agenda

• Motivations

• Problem Definitions

• Preliminaries

• Main Results – Approximation Algorithms

• Simulation Results

• Conclusion
\(k\)-traveling Salesperson Problem with Neighborhood (\(k\)-TSPN)

- 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 starts from one root,
  - the neighborhood of each node is visited by some tour,
  - the cost of the longest tour becomes minimum.

http://www.cs.nccu.edu/~advnet/
**k-rooted Path Cover Problem with Neighborhood (k-PCPN)**

- 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.
$k$-rooted Tree Cover Problem with Neighborhood ($k$-TCPN)

- Given a set $V$ of nodes $\{v_1, v_2, \cdots, v_n\}$ and a set $R$ of $k$ roots $\{r_1, r_2, \cdots, r_k\}$, 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.
Agenda

• Motivations

• Problem Definitions

• Preliminaries

• Main Results – Approximation Algorithms

• Simulation Results

• Conclusion

http://www.cs.nccu.edu/~advnet/
Tree, Tour, and Path

- Christofides’ 1.5-approximation for TSP [17]
  - Use an MST to find a tour.
  - A path can be obtained from the tour (no cost increase).

1. **MST**
2. Add perfect matching edges among odd degree nodes
3. Apply triangular inequality to have a tour
4. Remove an edge to get a path

[Diagram showing the steps]
**$k$-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

Cost of the solution

- Evan et al. proposed $(4+\varepsilon)$-approximation algorithm [13]
Minimum Spanning Tree with Neighborhood (MSTN) Problem

- Given a set \( V \) of nodes \( \{v_1, v_2, \ldots, v_n\} \), find an MST over the neighborhood areas of the nodes in \( V \).

- Yang et al. [16]
  - Assumptions:
    - the neighborhood area of a node is a circle centered at the node with radius 1.
    - the neighborhood areas of nodes do NOT overlap. (i.e. distance is at least 2)
  - Proposed two approximation algorithms and PTAS.
  - analysis is heavily dependent on the “pair-wise disjointness” of neighborhoods.
Agenda

• Motivations

• Problem Definitions

• Preliminaries

• **Main Results – Approximation Algorithms**

• Simulation Results

• Conclusion

http://www.cs.nccu.edu/~advnet/
General Minimum Spanning Tree with Neighborhood (GMSTN) Problem

• An extension of MSTN
• Assumptions (generalized):
  • the radius of the neighborhood area of each node is \( d \).
  • the neighborhood areas may overlap with each other.
General Minimum Spanning Tree with Neighborhood Algorithm (GMSTNA)

• Utilize Mitchell’s approximation strategy for TSP.
  (a) Compute a maximal “pair-wise disjoint” disk set.
  (b) Compute an MST of the centers of the disjoint disks.
  (c) Merge the borders of the disjoint disks and the MST excluding the segments inside the disks

Feasible Solution
Further Optimization

- Find a set of points representing disks.
- Calculate the MST of the subset of points (approx. set-cover)
- Cost never increases!
Performance Analysis of GMSTNA

• Lemma 4.1

• $$\text{Len}(T_{out}) \leq (d(2\pi - 1) + 1)\text{Len}(T_{mst-I}^{\text{center}}) + 2\pi d$$
  
  • $$I$$ is the set of “pair-wise disjoint” disks.
  
  • $$T_{mst-I}^{\text{center}}$$ is the MST of the centers of the disks in $$I$$.
  
  • $$T_{out}$$ is an output of GMSTNA.
  
  • $$d$$ is the radius of the circular neighborhood area.
  
  • $$\text{Len}(\cdot)$$: the total length (cost) of the tree (or path, tour).
Performance Analysis of GMSTNA – cont’

• Lemma 4.1
  \[ \text{Len}(T_{out}) \leq (d(2\pi - 1) + 1)\text{Len}(T_{mst-I}^{\text{center}}) + 2\pi d \]

• Theorem 4.2 \cite{16}
  \[ \text{Given a set } V \text{ of nodes } \{V_1, V_2, \ldots, V_n\}, \text{ let } T^{\text{center}}_{mst-I} \text{ and } T^{\text{disk}}_{mst-I} \text{ be an MST of } V \text{ and an MST of a set of “pair-wise disjoint” disks } \{N(v_1), N(v_2), \ldots, N(v_n)\}. \text{Then, } \text{Len}(T^{\text{center}}_{mst-I}) \leq (1 + \frac{20}{\pi})\text{Len}(T^{\text{disk}}_{mst-I}) + 2d. \]
Theorem 4.3
(from Lemma 4.1 and Theorem 4.2)

\[ \text{Len}(T_{out}) \leq (d(2\pi - 1) + 1)\text{Len}(T_{mst-I}^{\text{center}}) + 2\pi d \]
\[ \leq (d(2\pi - 1) + 1)((1 + 20/\pi)\text{Len}(T_{mst-I}^{\text{disk}}) + 2d) + 2\pi d \]
\[ \leq (d(2\pi - 1) + 1)((1 + 20/\pi)\text{Len}(T_{mst}^{\text{disk}}) + 2d) + 2\pi d \]

Output of GMSTNA (before optimization)

Optimal

http://www.cs.nccu.edu/~advnet/
$k$-rooted Tree Cover Problem with Neighborhood Algorithm ($k$-TCPNA)

1) Select the disks centered at the roots.
2) Find the maximal pair-wise disjoint disks.
$k$-TCPNA: Example – cont’

3) Apply the $(4+\varepsilon)$-approximation [13] for the centers of the pair-wise disjoint disks.
$k$-TCPNA: Example – cont’

4) Apply the **optimization technique**
$k$-TCPNA: Example – cont’

5) Compute $k$ MSTs

http://www.cs.nccu.edu/~advnet/
$k$-Traveling Salesperson Problem with Neighborhood ($k$-TSPN)

6-1) Convert each MST into a tour
\(k\)-rooted Path Cover Problem with Neighborhood (\(k\)-PCPN)

6-2) Convert each MST into a path
Performance Analysis

- Theorem 4.6
  - There exists a constant factor approximation algorithm for $k$-PCPN.

$$\leq 1.5 \cdot ((d(2\pi - 1) + 1)((1 + 20/\pi)\text{Cost}(OPT_p) + 2d) + (2\pi + 1)d) \cdot (4 + \varepsilon)$$

(2) GMSTNA: from $k$-rooted (center) trees to $k$-rooted neighborhood spanning trees

(3) Christofides’ 1.5-approximation for TSP [17]: To convert MSTs to Paths

(1) Evan et al.’s the $(4+\varepsilon)$-approximation algorithm for $k$-rooted tree cover problem [13]: To partition the nodes and have $k$-rooted (center) trees.
Performance Analysis – cont’

• Theorem 4.6
  • There exists a constant factor approximation algorithm for $k$-PCPN.

\[ \leq 1.5 \cdot \left( (d(2\pi - 1) + 1)((1 + 20 / \pi)\text{Cost}(OPT_P) + 2d) + (2\pi + 1)d \right) \cdot (4 + \varepsilon) \]

• Corollary 4.1
  • There exists a constant factor approximation algorithm for $k$-TSPN.

\[ \leq 1.5 \cdot \left( (d(2\pi - 1) + 1)((1 + 20 / \pi)\text{Cost}(OPT_U) + 2d) + (2\pi + 1)d \right) \cdot (4 + \varepsilon) \]
Agenda

• Motivations

• Problem Definitions

• Preliminaries

• Main Results – Approximation Algorithms

• Simulation Results

• Conclusion
Simulation Results

• Performed over $10 \times 10$, $15 \times 15$, and $30 \times 30$ virtual space.
• The number of nodes ($n$) varies to 30, 50, and 70.
• The number of roots ($k$) varies to 3 and 6.
• For each parameter setting, we created 100 instances and compute the average.
• Compare $k$-TCPNA with $k$-TCPA [13]
• Comparison is done in terms of Tree (also Tour and Path after the conversion)
• Metrics
  1) Improvement Ratio (IR)
     \[
     \frac{\text{Cost of output of } k\text{-TCPA} - \text{cost of output of } k\text{-TCPNA}}{\text{Cost of output of } k\text{-TCPA}}
     \]
  2) Direct Ratio (DR)
     \[
     \frac{\text{Cost of output of } k\text{-TCPNA}}{\text{Cost of output of } k\text{-TCPA}}
     \]
Simulation Results – cont’

1) As the number of nodes increases, $k$-TCPNA outperforms $k$-TCPA.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Tree-IR</th>
<th>Tree-DR</th>
<th>Tour-IR</th>
<th>Tour-DR</th>
<th>Path-IR</th>
<th>Path-DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.313</td>
<td>0.687</td>
<td>0.251</td>
<td>0.749</td>
<td>0.263</td>
<td>0.737</td>
</tr>
<tr>
<td>50</td>
<td>0.355</td>
<td>0.645</td>
<td>0.308</td>
<td>0.692</td>
<td>0.320</td>
<td>0.680</td>
</tr>
<tr>
<td>70</td>
<td>0.435</td>
<td>0.565</td>
<td>0.381</td>
<td>0.619</td>
<td>0.389</td>
<td>0.611</td>
</tr>
</tbody>
</table>

2) As the size of the virtual space decreases, $k$-TCPNA outperforms $k$-TCPA.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Tree-IR</th>
<th>Tree-DR</th>
<th>Tour-IR</th>
<th>Tour-DR</th>
<th>Path-IR</th>
<th>Path-DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.182</td>
<td>0.818</td>
<td>0.132</td>
<td>0.868</td>
<td>0.135</td>
<td>0.865</td>
</tr>
<tr>
<td>50</td>
<td>0.188</td>
<td>0.812</td>
<td>0.147</td>
<td>0.853</td>
<td>0.177</td>
<td>0.823</td>
</tr>
<tr>
<td>70</td>
<td>0.270</td>
<td>0.730</td>
<td>0.239</td>
<td>0.761</td>
<td>0.255</td>
<td>0.745</td>
</tr>
</tbody>
</table>

3) As we have less number of mobile elements, $k$-TCPNA outperforms $k$-TCPA.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Tree-IR</th>
<th>Tree-DR</th>
<th>Tour-IR</th>
<th>Tour-DR</th>
<th>Path-IR</th>
<th>Path-DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.263</td>
<td>0.737</td>
<td>0.210</td>
<td>0.790</td>
<td>0.215</td>
<td>0.785</td>
</tr>
<tr>
<td>50</td>
<td>0.276</td>
<td>0.724</td>
<td>0.241</td>
<td>0.759</td>
<td>0.26</td>
<td>0.74</td>
</tr>
<tr>
<td>70</td>
<td>0.322</td>
<td>0.678</td>
<td>0.273</td>
<td>0.727</td>
<td>0.282</td>
<td>0.718</td>
</tr>
</tbody>
</table>

http://www.cs.nccu.edu/~advnet/
Agenda

• Motivations

• Problem Definitions

• Preliminaries

• Main Results – Approximation Algorithms

• Simulation Results

• Conclusion
Conclusion

• Introduce three new problems
  • $k$-traveling Salesperson Problem with Neighborhood ($k$-TSPN)
  • $k$-rooted Path Cover Problem with Neighborhood ($k$-PCPN)
  • $k$-rooted Tree Cover Problem with Neighborhood ($k$-TCPN)

• Constant Factor Approximation Algorithms for
  • General Minimum Spanning Tree with Neighborhood Algorithm (GMSTNA)
  • $k$-TCPN, $k$-PCPN, and $k$-TSPN

• Future Work

http://www.cs.nccu.edu/~advnet/
Thank you
Question?