Collaborative Graph Exploration

Christian Ortolf
Laboratory for Electrical Instrumentation,
University of Freiburg
Moving and Computing, La Maddalena 2017
Motivation

- Automated city cleaning
- Treasure hunt
- Missing person search

- Robot-Team
- Traverse Landscape
- Time essential
- Coordination critical
Model – Problem description

- Environment given as Graph $G$
- $k$ robots explore
- Round model (FSYNC)
- Visit all $n$ nodes
  - First Visit – exploration time
Model – Well equipped robots

- No …
  - … energy restriction
  - … restriction of computational power
  - … localization problems
  - … navigation problems
  - … failing robots
Model – Exploration scenario

- No Map
  - $G$ unknown
  - visited nodes
  - adjacent edges

- One starting node
- First visit interesting
- Trees, Grids with Convex obstacles
Model - Competitive Analysis

- Compare to unrestricted team

\[
\text{cost} = \frac{\text{Time Online}}{\text{Time Offline}}
\]
Offline Exploration - Perfect

- At least:
  \[ \max\left(\left\lceil \frac{n}{k} \right\rceil, d \right) \]

- NP hard
Offline Exploration - Simple

- DFS path
- k Subpaths

\[
\left\lceil \frac{2n}{k} \right\rceil + d
\]
Exploration - Milestones ’04

- Greedy
  \[ O\left(\frac{k}{\log k}\right) \]

Collective tree exploration,
Fraigniaud et al.,
LATIN 2004
Exploration - Milestones ’07

- Jellyfish

\[ \Omega \left( \frac{\log k}{\log \log k} \right) \]

Why Robots need maps
Dynia et al., SIROCCO 2007
Exploration - Our Results

- Lower bound for grids
  \[ \Omega \left( \frac{\log k}{\log \log k} \right) \]

- Upper bound for grids
  \[ O(\log^2 n) \]

- \ldots randomized algorithms

  - Grids
  \[ \Omega \left( \frac{\sqrt{\log m}}{\log \log m} \right) \]

  - Trees
  \[ \Omega \left( \frac{\log k}{\log \log k} \right) \]

Online Multi-Robot Exploration of Grid Graphs with Rectangular Obstacles, O. et al, SPAA 2012
Exploration - Our Results

- Yo*-Algorithmus

\[ O\left(2^{(2+o(1))\sqrt{(\log d)(\log \log k)}}(\log k)(\log k + \log n)\right) \]

- Simplified:

\[ n^{o(1)} \]

A recursive approach to multi-robot exploration of trees, O. et al, SIROCCO 2014
Lower Bound - Separation
Lower Bound - Poisons
Exploration Grid Graphs

- **Move to** \( P \)
  \[ O(\log m) \]

- „**Divide and Conquer“-Algorithm**
  \[ O(\log^2 n) \]
From Yo-yo to Yo*

- Explore each depth separately
- 4d-competitive
- Independend of k
From Yo-yo to Yo*

- Explore each depth separately
- 4d-competitive
- Independent of k
From Yo-yo to Yo*

- Explore each depth separately
- 4d-competitive
- Independent of $k$
From Yo-yo to Yo*  

- Assume $d$ known  
- Divide tree  
- Explore level before going on
Yo-1 - Algorithm

- Use DFS
  \[ k \leq |R| \]
- Else Yo-yo
- Robots get redistributed when half are done
- Balancing cost:
  \[ O(b \cdot d \cdot (\log n + \log k)) \]
Yo-1 - First Recursion

- Choose $c, a$ optimal

- Here: $c = a = d^{\frac{1}{2}}$

$$O\left(d^{\frac{1}{2}} \left(\log n + \log k\right)\right)$$
Yo* - More Recursions

- Call created Algorithm
  \[ Y_0 - (l + 1) \]
- Optimal \( l \) ?
Yo* - competitiveness

- Yo-  
  \[ O\left(20^l \frac{1}{d^{l+1}} (\log k)^l (\log k + \log n)(\log d)\right) \]

- Yo*  
  \[ l = \sqrt{\frac{\log d}{\log \log k}} \]

\[ \Rightarrow O\left(2^{(2+o(1))\sqrt{(\log d)(\log \log k)}} (\log k)(\log k + \log n)\right) \]

\[ \Rightarrow n^{o(1)}, k = n^c, d = n^{c'}, 0 < c, c' < 1 \]
Empirism - Comb

- Worst case
  - Greedy
    \( O\left(\frac{k}{\log k}\right) \)
  - Lazy
    \( O(k) \)
- Scaled with \( k \)
Empirism - Improvements

1. Redistribute without stopping
2. Do not wait with redistributing
3. Efficient base algorithm
4. Multiple levels in parallel
Empirism - Hedge Cutter
Conclusion

- Close bounds for Grids exist
- Subpolynomial competitiveness for tree exploration
- Hedge Cutter as open problem
## State of the Art - Exploration

<table>
<thead>
<tr>
<th>k</th>
<th>d</th>
<th>Graph</th>
<th>Competitive factor</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(O(\log^{c'} n))</td>
<td>any</td>
<td>any</td>
<td>(O(\log^{c'} n))</td>
<td>DFS</td>
</tr>
<tr>
<td>(O(\log^{c'} n))</td>
<td>any</td>
<td>Tree</td>
<td>(O(\log^{c'} n / \log \log n))</td>
<td>Greedy</td>
</tr>
<tr>
<td>any</td>
<td>(O(\sqrt{n}))</td>
<td>Grid w. O.</td>
<td>(O(\log^2 n))</td>
<td>D&amp;C alg</td>
</tr>
<tr>
<td>any</td>
<td>(\log^{c'} n)</td>
<td>Tree</td>
<td>(O(\log^{c'} n))</td>
<td>Yo-yo</td>
</tr>
<tr>
<td>(n^c)</td>
<td>(n^c)</td>
<td>Tree</td>
<td>(n^{o(1)})</td>
<td>Yo*</td>
</tr>
<tr>
<td>(n^c)</td>
<td>(n^c)</td>
<td>Comb</td>
<td>(O(\log n))</td>
<td>Hedge Cutter</td>
</tr>
<tr>
<td>(n)</td>
<td>(O(\sqrt{n}))</td>
<td>Grid w. O.</td>
<td>(O(\log n))</td>
<td>Online nav.</td>
</tr>
<tr>
<td>(dn^{1+c'})</td>
<td>any</td>
<td>any</td>
<td>(O(1/c'))</td>
<td>Fast Tree Expl.</td>
</tr>
<tr>
<td>(n^d)</td>
<td>any</td>
<td>any</td>
<td>1</td>
<td>Greedy/Flood</td>
</tr>
</tbody>
</table>

\(0 < c < 1, c' > 0\)

Christian Ortolf - Collaborative Graph Exploration
End

Thank You!