Distributed Maximal Matching: Greedy is Optimal

Juho Hirvonen and Jukka Suomela

Helsinki Institute for Information Technology HIIT
University of Helsinki
Finland

17 July 2012
PODC
Maximal Matchings

Input

Output
Distributed Algorithms

- Graph $G = \text{input} = \text{communication network}$
  - node = computer
  - edge = communication link
  - synchronous communication rounds, *deterministic* algorithms
Distributed Algorithms

• Graph $G =$ input = communication network

• Each node has to stop and output its own part of the solution
  • am I matched?
  • with whom?
Distributed Algorithms

- Time = number of communication rounds
- Equivalent:
  - running time is $T$
  - all nodes stop after $T$ communication rounds
  - output of node $v = f(\text{radius-}T\text{ neighbourhood of } v)$

- How fast can we find a maximal matching?
Time = \( f(n, \Delta) \)

- \( n \) = number of nodes
- \( \Delta \) = maximum degree
- Focus: \( \Delta \ll n \)
<table>
<thead>
<tr>
<th>problem</th>
<th>upper bound</th>
<th>lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximal matching</td>
<td>$\Delta + \log^* n$</td>
<td>$\text{polylog}(\Delta) + \log^* n$</td>
</tr>
<tr>
<td>$(\Delta+1)$-vertex colouring</td>
<td>$\Delta + \log^* n$</td>
<td>$\log^* n$</td>
</tr>
<tr>
<td>$(2\Delta-1)$-edge colouring</td>
<td>$\Delta + \log^* n$</td>
<td>$\log^* n$</td>
</tr>
<tr>
<td>maximal edge packing</td>
<td>$\Delta$</td>
<td>$\text{polylog}(\Delta)$</td>
</tr>
<tr>
<td>vertex cover 2-approx.</td>
<td>$\Delta$</td>
<td>$\text{polylog}(\Delta)$</td>
</tr>
</tbody>
</table>


Time = $f(n, \Delta)$

- Fairly well-understood as a function of $n$
  - tight upper and lower bounds if $\Delta = O(1)$
- Wide open as a function of $\Delta$
  - exponential gap
- **Linear-in-$\Delta$ lower bounds missing**
Plan

1. Study a simpler model
   • unique node identifiers make things complicated
   • but *what is the right model?*

2. Generalise
   • future work...
   • but see the next talk for a promising technique!
Simpler Model?

• Unique identifiers
  • standard model
  • complicated to analyse directly...

• Node colouring
  • weaker than unique identifiers
Simpler Model?

• Unique identifiers
  • standard model
  • complicated to analyse directly...

• Node colouring
  • weaker than unique identifiers
  • *too weak — cannot find a maximal matching!*
Simpler Model?

• Port numbering
  • another popular model
  • too weak...

• Edge colouring
  • stronger than port numbering
  • *just right for our purposes!*
Model

• Given: $k$-edge-coloured graph
  - proper edge colouring: adjacent edges have different colours
  - colour palette: $\{1, 2, \ldots, k\}$
  - anonymous nodes
  - nodes can use edge colours to refer to their neighbours
Greedy Algorithm

- Greedily add edges of colour 1, ...

Input

Greedy algorithm
Greedy Algorithm

- Greedily add edges of colour 1, 2, ...

Input

Greedy algorithm
Greedy Algorithm

- Greedily add edges of colour 1, 2, 3, ...

Input

Greedy algorithm
Greedy Algorithm

- Greedily add edges of colour 1, 2, ..., $k$
Greedy Algorithm

- That’s it – we have a maximal matching
Greedy Algorithm

• Running time is exactly $k - 1$ rounds
  • initially each node knows the colours of incident edges
  • analysis is tight

• But *is there a faster algorithm?*
Contributions

• Maximal matchings in $k$-edge-coloured graphs
• General graphs: $\geq k - 1$ rounds
  • matching upper bound: greedy
• Bounded-degree graphs: $\Omega(\Delta + \log^* k)$
  • matching upper bound: adaptation of Panconesi–Rizzi (2001)
Lower Bound

• $d$-regular, $k$-edge-coloured graphs

• $d = k$:
  • trivial to find a maximal matching in constant time (pick a colour class)

• $d = k - 1$:
  • as difficult as the general case!
  • we show that we need at least $d$ rounds
Lower Bound

• Given an algorithm \( A \)

• Construct two \( d \)-regular trees \( T_1 \) and \( T_2 \):
  • root nodes have *identical* \((d - 1)\)-neighbourhoods
  • root nodes produce *different outputs*

• Running time of \( A \) is at least \( d = k - 1 \)
Node Colours

- Node colour = the unique “missing colour”
Templates

- Degree < $d$
Templates

- Degree $< d$: add loops
Templates

- Degree $< d$: add loops, unfold loops
Templates

- Unfolding preserves traversals
Templates

• Compact representations of trees
Templates

\[ 1 \ 3 \ 4 \ = \ 4 \ 1 \ 2 \ = \ 1 \ 3 \ 4 \]
Templates

\[ 1 \quad = \quad \begin{array}{c}
\begin{array}{c}
1 \\
3
\end{array}
\end{array} \quad = \quad \begin{array}{c}
\begin{array}{c}
1 \\
2 \\
3 \\
4
\end{array}
\end{array} \]
What is the output of $A$ here?

Definition!
Induction

- Degree $i$ templates:
  - root nodes produce different outputs
  - identical neighbourhoods up to distance $i - 1$
- $i = 1$: base case
- $i > 1$: by induction
- $i = d$: main result
Base Case

- Edge of colour $y$ exists, in matching

- Edge of colour $y$ exists, but not in matching
Base Case

\[ x \leftrightarrow y \longrightarrow x \rightarrow z \quad K \]

\[ z \leftrightarrow y \longrightarrow z \rightarrow x \quad L \]
Base Case

output in $X$ cannot be copied from $K$ & $L$ – something must change!
Base Case

degree 1 templates, same radius-0 view, different output
Base Case

degree 1 templates, same radius-0 view, different output
Inductive Step

Given:
degree $i$ templates, same radius-$(i-1)$ view, different output

Construct:
degree $i+1$ templates, same radius-$i$ view, different output

(here $i = 1$)
Inductive Step

Choose one loop per node

Prefer loops that are matched in $T$

Then unfold these loops...
Inductive Step
Inductive Step

... again, something must change in the output!
Inductive Step
Inductive Step

same radius-o view

same radius-o view
Inductive Step

same radius-1 view
Conclusions

• By induction, we can construct:
  • two degree-$d$ trees
  • same radius-$(d-1)$ view
  • different output
Conclusions

• Maximal matching in \textit{k-edge-coloured} graphs requires:
  • \(k - 1\) communication rounds in general
  • \(\Theta(\Delta + \log^* k)\) rounds in graphs of degree \(\leq \Delta\)

• What if we have \textbf{unique identifiers}?
  • \textit{in progress}: tight bounds for \textit{maximal edge packings}...
  • \textit{still open}: tight bounds for \textit{maximal matchings}?