Lower bounds for maximal matchings and maximal independent sets

Jukka Suomela
Aalto University, Finland

arXiv:1901.02441
Joint work with

- **Alkida Balliu** · Aalto University
- **Sebastian Brandt** · ETH Zurich
- **Juho Hirvonen** · Aalto University
- **Dennis Olivetti** · Aalto University
- **Mikaël Rabie** · Aalto University and IRIF, University Paris Diderot

arXiv:1901.02441
Two classical graph problems

Maximal matching

Maximal independent set

Trivial linear-time centralized, sequential algorithm: add edges/nodes until stuck
Two classical graph problems

Maximal matching

Maximal independent set

Can be verified locally: if it looks correct everywhere locally, it is also feasible globally

Can these problems be solved locally?
Warmup: toy example

Bipartite graphs & port-numbering model
computer network with port numbering

bipartite, 2-colored graph

\(\Delta\)-regular (here \(\Delta = 3\))

output: maximal matching
Very simple algorithm

unmatched white nodes: send *proposal* to port 1
Very simple algorithm

unmatched white nodes:
send *proposal* to port 1

black nodes:
*accept* the first proposal you get, *reject* everything else
(break ties with port numbers)
**Very simple algorithm**

**unmatched white nodes:**
send *proposal* to port 1

**black nodes:**
*accept* the first proposal you get, *reject* everything else
(break ties with port numbers)
Very simple algorithm

unmatched white nodes: send proposal to port 2
Very simple algorithm

unmatched white nodes: send *proposal* to port 2

black nodes: accept the first proposal you get, *reject* everything else (break ties with port numbers)
Very simple algorithm

unmatched white nodes:
send *proposal* to port 2

black nodes:
*accept* the first proposal you get, *reject* everything else
(break ties with port numbers)
Very simple algorithm

unmatched white nodes:
send *proposal* to port 3
Very simple algorithm

unmatched white nodes:
send proposal to port 3

black nodes:
accept the first proposal you get, reject everything else
(break ties with port numbers)
Very simple algorithm

unmatched white nodes: send *proposal* to port 3

black nodes: *accept* the first proposal you get, *reject* everything else (break ties with port numbers)
Very simple algorithm

Finds a maximal matching in $O(\Delta)$ communication rounds

Note: running time does not depend on $n$
Bipartite maximal matching

• Maximal matching in very large 2-colored \( \Delta \)-regular graphs

• Simple algorithm: \( O(\Delta) \) rounds, independently of \( n \)

• Is this optimal?
  • \( o(\Delta) \) rounds?
  • \( O(\log \Delta) \) rounds?
  • 4 rounds??
Big picture
Bounded-degree graphs & LOCAL model
Distributed graph algorithms for maximal matching

• Maximal matching in general graphs
  • $n$ = number of nodes
  • $\Delta$ = maximum degree

• LOCAL model of distributed computing
  • “time” = number of synchronous communication rounds
    = how far do you need to see to choose your own part of solution
  • nodes are labeled with unique identifiers from $\{1, 2, \ldots, \text{poly}(n)\}$
  • $O(n)$ = trivial, $O(\text{diameter})$ = trivial

• Strong model — lower bounds widely applicable
Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

**Algorithms:**
- deterministic
- randomized

**Lower bounds:**
- deterministic
- randomized
Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

Algorithm:
- deterministic
- randomized

Lower bounds:
- deterministic
- randomized

Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

**Algorithms:**
- deterministic
- randomized

**Lower bounds:**
- deterministic
- randomized

Maximal matching, LOCAL model, \(O(f(\Delta) + g(n))\)

**Algorithms:**
- \(\bigcirc\) deterministic
- \(\bullet\) randomized

**Lower bounds:**
- \(\square\) deterministic
- \(\blacksquare\) randomized

\[O(\Delta + \log^* n)\] deterministic

\(\Delta\)

Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

Algorithms:
- Deterministic
- Randomized

Lower bounds:
- Deterministic
- Randomized


Kuhn et al. (2004, 2016)

Panconesi & Rizzi (2001)

Hanckowiak et al. (1998)

Hanckowiak et al. (2001)

Israeli & Itai (1986)
Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

**Algorithms:**
- Deterministic
- Randomized

**Lower bounds:**
- Deterministic
- Randomized


Kuhn et al. (2004, 2016)

Barenboim et al. (2012, 2016)

Fischer (2017)

Hanckowiak et al. (1998)

Hanckowiak et al. (2001)

Fischer (2017)

Israel & Itai (1986)

Panconesi & Rizzi (2001)
Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

**Algorithms:**
- deterministic
- randomized

**Lower bounds:**
- deterministic
- randomized

\[ O(\log \Delta + \log^* n) ??? \]
\[
\log^7 n - \text{Hanckowiak et al. (1998)} \\
\log^4 n - \text{Hanckowiak et al. (2001)} \\
\log^3 n - \text{Fischer (2017)} \\
\log n - \text{Israeli & Itai (1986)} \\
\sqrt{\frac{\log n}{\log \log n}} - \text{Barenboim et al. (2012, 2016)} \\
\log \log n - \text{Kuhn et al. (2004, 2016)} \\
\frac{\log n}{\log \log n} - \text{Panconesi & Rizzi (2001)} \\
\log^{\ast} n - \text{Linial (1987, 1992), Naor (1991)} \\
\frac{\log \Delta}{\log log \Delta} - \Delta - \text{New}
\]

\text{Maximal matching, LOCAL model, } O(f(\Delta) + g(n))

\text{Algorithms:}
\begin{itemize}
  \item deterministic
  \item randomized
\end{itemize}

\text{Lower bounds:}
\begin{itemize}
  \item deterministic
  \item randomized
\end{itemize}
Main results

Maximal matching and maximal independent set cannot be solved in

• \( o(\Delta + \log \log n / \log \log \log n) \) rounds
  with randomized algorithms

• \( o(\Delta + \log n / \log \log n) \) rounds
  with deterministic algorithms

Upper bound: \( O(\Delta + \log^* n) \)
Very simple algorithm

unmatched white nodes: send *proposal* to port 1

black nodes: accept the first proposal you get, reject everything else (break ties with port numbers)

This is optimal!
Proof techniques

Speedup simulation
Speedup simulation technique

• Given:
  • algorithm $A_0$ solves problem $P_0$ in $T$ rounds

• We construct:
  • algorithm $A_1$ solves problem $P_1$ in $T - 1$ rounds
  • algorithm $A_2$ solves problem $P_2$ in $T - 2$ rounds
  • algorithm $A_3$ solves problem $P_3$ in $T - 3$ rounds
    ...
  • algorithm $A_T$ solves problem $P_T$ in 0 rounds

• But $P_T$ is nontrivial, so $A_0$ cannot exist

• Given:
  • algorithm $A_0$ solves 3-coloring in $T = o(\log^* n)$ rounds

• We construct:
  • algorithm $A_1$ solves $2^3$-coloring in $T - 1$ rounds
  • algorithm $A_2$ solves $2^{2^3}$-coloring in $T - 2$ rounds
  • algorithm $A_3$ solves $2^{2^{2^3}}$-coloring in $T - 3$ rounds
    ...
  • algorithm $A_T$ solves $o(n)$-coloring in 0 rounds

• But $o(n)$-coloring is nontrivial, so $A_0$ cannot exist
Brandt et al. (2016): sinkless orientation

• **Given:**
  • algorithm \( A_0 \) solves *sinkless orientation* in \( T = o(\log n) \) rounds

• **We construct:**
  • algorithm \( A_1 \) solves *sinkless coloring* in \( T - 1 \) rounds
  • algorithm \( A_2 \) solves *sinkless orientation* in \( T - 2 \) rounds
  • algorithm \( A_3 \) solves *sinkless coloring* in \( T - 3 \) rounds
    ...
  • algorithm \( A_T \) solves *sinkless orientation* in 0 rounds

• But *sinkless orientation* is nontrivial, so \( A_0 \) cannot exist
Speedup simulation technique for maximal matching

• **Given:**
  • algorithm $A_0$ solves problem $P_0 = \text{maximal matching}$ in $T$ rounds

• **We construct:**
  • algorithm $A_1$ solves problem $P_1$ in $T - 1$ rounds
  • algorithm $A_2$ solves problem $P_2$ in $T - 2$ rounds
  • algorithm $A_3$ solves problem $P_3$ in $T - 3$ rounds
  ...
  • algorithm $A_T$ solves problem $P_T$ in 0 rounds

• But $P_T$ is nontrivial, so $A_0$ cannot exist

What are the right problems $P_i$ here?
Speedup simulation technique for maximal matching

• Given:
  • algorithm $A_0$ solves problem $P_0 = \text{maximal matching}$ in $T$ rounds

• We construct:
  • algorithm $A_1$ solves problem $P_1$ in $T - 1$ rounds
  • algorithm $A_2$ solves problem $P_2$ in $T - 2$ rounds
  • algorithm $A_3$ solves problem $P_3$ in $T - 3$ rounds
    ...
  • algorithm $A_T$ solves problem $P_T$ in 0 rounds

• But $P_T$ is nontrivial, so $A_0$ cannot exist
Representation for maximal matchings

white nodes “active”
output one of these:
· $1 \times M$ and $(\Delta - 1) \times O$
· $\Delta \times P$

black nodes “passive”
accept one of these:
· $1 \times M$ and $(\Delta - 1) \times \{P, O\}$
· $\Delta \times O$

M = “matched”
P = “pointer to matched”
O = “other”
We emphasize that the order of the elements does not matter here, and we could equally well write: 

"For a matched white node, one edge is labeled with an M, and 0 to indicate an edge in the matching. However, for brevity we will here represent multisets as regular expressions to represent them. When \( x = 00[01] \), or even \( 010 \). Now that \( x \) is a pair of labels, we can conveniently use black nodes to ensure that pointers do not point to unmatched black nodes (a maximal matchings)."

Here we will assume that feasibility of a solution does not depend on the port numbering. Hence, we have \( \{ \cdot \Delta \times \} = \{ O \} \). For an unmatched white node, all incident edges have to be labeled \( P \). The rules for the maximal matchings are: 

- Output one of these: \( 1 \times M \) and \( (\Delta-1) \times O \)
- \( \Delta \times P \)

\[
W = M O^{\Delta-1} | P^\Delta
\]

"For black nodes, the rules are: accept one of these: \( 1 \times M \) and \( (\Delta-1) \times \{ P, O \} \)
- \( \Delta \times O \)

\[
B = M[PO]^{\Delta-1} | O^\Delta
\]
Parameterized problem family

\[ W = \text{MO}^{\Delta-1} | \text{P}^\Delta, \]
\[ B = \text{M}[\text{PO}]^{\Delta-1} | \text{O}^\Delta \]

\[ W_\Delta(x, y) = \left( \text{MO}^{d-1} | \text{P}^d \right) \text{O}^y \text{X}^x, \]
\[ B_\Delta(x, y) = \left( [\text{MX}[\text{POX}]^{d-1} | [\text{OX}]^d \right) [\text{POX}]^y [\text{MPOX}]^x, \]

\[ d = \Delta - x - y \]
Main lemma

• Given: A solves $P(x, y)$ in $T$ rounds
• We can construct: $A'$ solves $P(x + 1, y + x)$ in $T - 1$ rounds

\[ W_\Delta(x, y) = \left( \text{MO}^{d-1} \mid P^d \right) O^y X^x, \]
\[ B_\Delta(x, y) = \left( \text{MX}[\text{POX}]^{d-1} \mid [\text{OX}]^d \right) [\text{POX}]^y [\text{MPOX}]^x, \]
\[ d = \Delta - x - y \]
Putting things together

Maximal matching in \( o(\Delta) \) rounds

→ “\( \Delta^{1/2} \) matching” in \( o(\Delta^{1/2}) \) rounds

→ \( P(\Delta^{1/2}, 0) \) in \( o(\Delta^{1/2}) \) rounds

→ \( P(O(\Delta^{1/2}), o(\Delta)) \) in \( 0 \) rounds

→ contradiction

What we really care about

k-matching: select at most \( k \) edges per node

Apply speedup simulation \( o(\Delta^{1/2}) \) times
Putting things together

• Basic version:
  • deterministic lower bound, *port-numbering model*

• Analyze what happens to local failure probability:
  • *randomized* lower bound, port-numbering model

• With randomness you can construct unique identifiers w.h.p.:
  • randomized lower bound, *LOCAL model*

• Fast deterministic $\rightarrow$ very fast randomized
  • stronger *deterministic* lower bound, LOCAL model

Proof technique does not work directly with unique IDs
Main results

Maximal matching and maximal independent set cannot be solved in

- $o(\Delta + \log \log n / \log \log \log n)$ rounds with randomized algorithms
- $o(\Delta + \log n / \log \log n)$ rounds with deterministic algorithms

Lower bound for MM implies a lower bound for MIS
Some open questions

• $\Delta \ll \log \log n$:
  • complexity of $(\Delta+1)$-vertex coloring or $(2\Delta-1)$-edge coloring?
  • example: are these possible in $O(\log \Delta + \log^* n)$ time?

• $\Delta \gg \log \log n$:
  • complexity of maximal independent set?
  • is it much harder than maximal matching in this region?
  • example: is it possible in deterministic polylog($n$) time?
Summary

• **Linear-in-Δ lower bounds** for maximal matchings and maximal independent sets

• Old: can be solved in $O(\Delta + \log^* n)$ rounds

• New: cannot be solved in
  • $o(\Delta + \log \log n / \log \log \log n)$ rounds with randomized algorithms
  • $o(\Delta + \log n / \log \log n)$ rounds with deterministic algorithms

• Technique: speedup simulation

arXiv:1901.02441
Speedup simulation

Given: white algorithm $A$ that runs in $T = 2$ rounds

- $v_1$ in $A$ sees $U$ and $D_1$

Construct: black algorithm $A'$ that runs in $T - 1 = 1$ rounds

- $u$ in $A'$ only sees $U$

$A'$: what is the set of possible outputs of $A$ for edge $\{u, v_1\}$ over all possible inputs in $D_1$?