Lower Bounds for Maximal Matchings and Maximal Independent Sets

Alkida Balliu
Aalto University, Finland
Joint work with

Sebastian Brandt · ETH Zurich
Juho Hirvonen · Aalto University
Dennis Olivetti · Aalto University
Mikaël Rabie · LIP6 - Sorbonne University
Jukka Suomela · Aalto University
Overview

Maximal matching

Maximal independent set

We will talk about lower bounds for solving these problems in the distributed setting.
Distributed setting

Graph = communication network; synchronous rounds; time = number of communication rounds
Maximal matching problem

• **Matching**: edges in the matching do not share a node

• **Maximality**: if we add any other edge in the matching, than it is not a matching anymore

• We say that **a node is matched**: it is an endpoint of an edge in the matching
Maximal independent set problem

Input

Output

• **Independent set**: nodes in the IS do not share an edge

• **Maximality**: if we add any other node in the IS, than it is not independent anymore
Two classical graph problems

Maximal matching

Easy linear-time centralized algorithm: add edges/nodes until stuck

Maximal independent set
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?
Locality = how far do I need to see to produce my own part of the solution?
Locality = how far do I need to see to produce my own part of the solution?
Locality = how far do I need to see to produce my own part of the solution?

Local outputs form a globally consistent solution
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 Δ-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
LOCAL model

• Each node has a **unique identifier** from 1 to \( \text{poly}(n) \)

• **No bounds** on the computational power

• **No bounds** on the bandwidth

• **Synchronous** model

• **Everything** can be solved in **Diameter** time

**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, $O(f(\Delta) + g(n))$

Algorithms:
- 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:**
- Deterministic
- Randomized

**Lower bounds:**
- Deterministic
- Randomized

**Graphical Notes:**
- \( \Delta \) represents the maximum degree of a graph.
- \( \log^* n \) is the iterated logarithm.
- \( f(n) \) and \( g(n) \) are functions of \( n \).
- \( O(\Delta + \log^* n) \) deterministic.
- \( F(\log \log n, \log \log \log n) \) polylogarithmic.

**References:**
- Israeli & Itai (1986)
- Hanckowiak et al. (1998)
- Hanckowiak et al. (2001)
- Fischer (2017)
- Panconesi & Rizzi (2001)
Maximal matching, LOCAL model, \(O(f(\Delta) + g(n))\)

Algorithms:
- deterministic
- randomized

Lower bounds:
- deterministic
- randomized

Fischer (2017)
Hanckowiak et al. (1998)
Hanckowiak et al. (2001)
Israeli & Itai (1986)
Kuhn et al. (2004, 2016)
Panconesi & Rizzi (2001)
Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

**Algorithms:**
- deterministic
- randomized

**Lower bounds:**
- deterministic
- randomized


Kuhn et al. (2004, 2016)

Fischer (2017)

Barenboim et al. (2012, 2016)

Hanckowiak et al. (1998)

Hanckowiak et al. (2001)

Fischer (2017)

Israeli & Itai (1986)

Panconesi & Rizzi (2001)
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:**
- 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)

Israeli & Itai (1986)

Panconesi & Rizzi (2001)

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:
- deterministic
- randomized

Lower bounds:
- deterministic
- randomized

Kuhn et al. (2004, 2016)
Fischer (2017)
Barenboim et al. (2012, 2016)
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
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, in the LOCAL model

• $o(\Delta + \log n / \log \log n)$ rounds
  with deterministic algorithms, in the LOCAL model

Upper bound: $O(\Delta + \log^* n)$
**This is optimal!**

**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)
Lower bound for MM implies lower bound for MIS

An algorithm for MIS implies an algorithm for MM
Lower bound for MM implies lower bound for MIS

An algorithm for MIS implies an algorithm for MM
Lower bound for MM implies lower bound for MIS

An algorithm for MIS implies an algorithm for MM
Lower bound for MM implies lower bound for MIS

An algorithm for MIS implies an algorithm for MM

If we cannot solve MM in $o(\Delta)$, then we cannot solve MIS in $o(\Delta)$
Proof techniques

Round elimination
Round elimination 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

• 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 $2^{23}$-coloring in $T - 2$ rounds
  • algorithm $A_3$ solves $2^{223}$-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
Round elimination 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

$P_i$ can be found automatically

[Brandt, 2019]
Round elimination 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

Challenge: keep $P_i$ small
Round elimination technique for MM

• 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
General approach

Maximal matching in $o(\Delta)$ rounds

What we really care about
General approach

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

What we really care about

k-matching: select at most k edges per node
General approach

Maximal matching in $o(\Delta)$ rounds
$\rightarrow$ “$\Delta^{1/2}$ matching” in $o(\Delta^{1/2})$ rounds
$\rightarrow$ $P(\Delta^{1/2}, 0)$ in $o(\Delta^{1/2})$ rounds

What we really care about

$k$-matching: select at most $k$ edges per node
General approach

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

What we really care about

k-matching: select at most $k$ edges per node

Apply round elimination $o(\Delta^{1/2})$ times
**General approach**

Maximal matching in $o(\Delta)$ rounds

$\rightarrow$ "$\Delta^{1/2}$ matching" in $o(\Delta^{1/2})$ rounds

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

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

$\rightarrow$ contradiction

---

**What we really care about**

**k-matching:** select at most $k$ edges per node

Apply round elimination $o(\Delta^{1/2})$ times
Representation for maximal matchings

white nodes “active”

output one of these:

\[ 1 \times M \text{ and } (\Delta-1) \times O \]

\[ \Delta \times P \]

black nodes “passive”

accept one of these:

\[ 1 \times M \text{ and } (\Delta-1) \times \{P, O\} \]

\[ \Delta \times O \]

\[ W = MO^{\Delta-1} \mid P^\Delta \]

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

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

maximal matching

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

“weak” matching
Parametrized problem family

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

maximal matching

\[ W_{\Delta}(x, y) = \left( \text{MO}^{d-1} \mid \text{P}^{d} \right) \text{O}^y \text{X}^x, \]

“weak” matching

A node v can be matched with at most x neighbours
If v is not matched, at most y neighbours can be unmatched
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

\[
W_\Delta(x, y) = \begin{pmatrix} \text{MO}^{d-1} & P^d \end{pmatrix} O^y X^x, \\
B_\Delta(x, y) = \begin{pmatrix} [MX][POX]^{d-1} & [OX]^d \end{pmatrix} [POX]^y [MPOX]^x, \\
d = \Delta - x - y
\]
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
Summary

• **Linear-in-\(\Delta\) 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: *round elimination*
Round eliminator: example MM

• Round eliminator program link: https://users.aalto.fi/~olivetd1/round-eliminator

Some open questions

• Complexity of \((\Delta+1)\)-vertex coloring?
  • can be solved in \(\tilde{O}(\Delta^{1/2}) + O(\log^* n)\) rounds [Fraigniaud et al., 2016]
  • cannot be solved in \(o(\log^* n)\) rounds [Linial, 1987]
  • example: is it solvable in \(O(\log \Delta + \log^* n)\) time?
• Better understanding of the round elimination technique
Some open questions

• Complexity of $\text{(Δ+1)-vertex coloring}$?
  - can be solved in $\tilde{O}(Δ^{1/2}) + O(\log^* n)$ rounds [Fraigniaud et al., 2016]
  - cannot be solved in $o(\log^* n)$ rounds [Linial, 1987]
  - example: is it solvable in $O(\log Δ + \log^* n)$ time?

• Better understanding of the round elimination technique

Thank you!