What can be decided locally without identifiers?

Pierre Fraigniaud
Mika Göös
Amos Korman
Jukka Suomela

University Paris Diderot & CNRS
University of Toronto
University Paris Diderot & CNRS
University of Helsinki & HIIT
Input: graph $G$

Local decision

Fraigniaud et al.

Local decision without IDs

23rd July 2013
Input: graph $G$
Output: is $G \in \mathcal{P}$?
Local decision

Input: graph $G$
Output: is $G \in \mathcal{P}$?
Input: graph $G$
Output: is $G \in \mathcal{P}$?
Local decision

Input: graph $G$

Output: is $G \in \mathcal{P}$?
**Input:** graph $G$

**Output:** is $G \in \mathcal{P}$?
Input: graph $G$
Output: is $G \in \mathcal{P}$?
Input: graph $G$
Output: is $G \in \mathcal{P}$?

Local algorithm
\[ \equiv O(1) \text{ communication rounds} \]
\[ \equiv O(1) \text{ radius neighbourhood} \]
Input: graph $G$
Output: is $G \in \mathcal{P}$?

$\text{yes} / \text{no}$
Local decision

**Input:** graph $G$

**Output:** is $G \in \mathcal{P}$?

$G$ is accepted iff all nodes output *yes*
Local decision

Input: graph $G$
Output: is $G \in \mathcal{P}$?

Locally decidable $\mathcal{P}$:
- triangle-freeness
- Eulerian graphs
- line graphs
- Locally checkable labellings $(G, \ell)$
Our question

We ask: Do node identifiers help in local decision?
Our question

We ask: Do node identifiers help in local decision?

IDs do not seem useful…

- Graph properties do not depend on node labels
- Symmetry breaking is not needed for decision problems!
Our question—formalised

**LOCAL** model
(deterministic)

\[ V(G) \subseteq \{1,2,3,\ldots\} \]
Our question—formalised

**LOCAL** model (deterministic)

\[ V(G) \subseteq \{1, 2, 3, \ldots \} \]

**ID-oblivious model**

**Restriction:** Output is **invariant** under relabelling the nodes

(i.e., depends only on **topology**)

[FHK OPODIS’12]
Easy cases

Warm up!

Under some assumptions:

\[ \text{LOCAL} = \text{ID-obliviuous} \]

Proof by simulation…
Easy cases

Let $A$ be a $\text{LOCAL}$ decision algorithm

**ID-oblivious simulation of $A$**

**Input:** local neighbourhood $(H, v)$ of $G$

For each ID-assignment $f : V(H) \to \{1, 2, \ldots, n\}$:

- if $A(f(H, v)) = \text{no}$ then output $\text{no}$.

Otherwise output $\text{yes}$.

**Assumptions:** ● Nodes know $n$
Easy cases

Let \( A \) be a \texttt{LOCAL} decision algorithm

<table>
<thead>
<tr>
<th><strong>ID-oblivious simulation of ( A )</strong>*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> local neighbourhood ((H, v)) of ( G )</td>
</tr>
<tr>
<td>For each ID-assignment ( f : V(H) \to {1, 2, \ldots } ):</td>
</tr>
<tr>
<td>- if ( A(f(H, v)) = \text{no} ) then output ( \text{no} ).</td>
</tr>
<tr>
<td>Otherwise output ( \text{yes} ).</td>
</tr>
</tbody>
</table>

**Assumptions:** Nodes do not know \( n \)
Easy cases

Let $A$ be a $\textit{LOCAL}$ decision algorithm

**ID-oblivious simulation of $A$**

**Input:** local neighbourhood $(H,v)$ of $G$

For each ID-assignment $f : V(H) \to \{1, 2, \ldots \}$:

- if $A(f(H,v)) = \text{no}$ then output $\text{no}$.
- Otherwise, output $\text{yes}$.

**Assumptions:**
- Nodes do not know $n$
- Nodes are Turing computable
Our main result

<table>
<thead>
<tr>
<th>Main theorem*</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCAL $\neq$ ID-oblivious</td>
</tr>
<tr>
<td>(I.e., there is a locally decidable property that cannot be decided ID-obliviously)</td>
</tr>
</tbody>
</table>

**Assumptions:**
- Nodes do not know $n$
- Nodes are Turing computable
Our main result

Main theorem*

\[
\text{LOCAL} \neq \text{ID-obliviuous}
\]

(I.e., there is a locally decidable property that cannot be decided ID-obliviiously)

* Contrary to a conjecture of \[\text{[FHK'}12]\]

Assumptions:
- Nodes do not know \(n\)
- Nodes are Turing computable
Our main result

**Main theorem**

\[
\text{LOCAL} \neq \text{ID-oblivilous}
\]

(I.e., there is a locally decidable property that cannot be decided ID-oblivilously)

* Contrary to a conjecture of [FHK’12]

**Proof...**
Separation under promise

<table>
<thead>
<tr>
<th>Promise problem</th>
</tr>
</thead>
</table>
| **Input:** ● $G = (G, M)$ is a labelled $n$-cycle  
● $M$ is a Turing machine |
| **Promise:** ● If $M$ halts in $s$ steps, then $n \geq s$ |
| **Output:** ● *yes* if $M$ runs forever  
● *no* if $M$ halts |
## Separation under promise

### Promise problem

| Input:                      | • $G = (G, M)$ is a labelled $n$-cycle  
|                            | • $M$ is a Turing machine              |
| Promise:                   | • If $M$ halts in $s$ steps, then $n \geq s$ |
| Output:                    | • Yes if $M$ runs forever               
|                            | • No if $M$ halts                       |

**ID-oblivious** **Impossible:** Must solve the Halting Problem
Separation under promise

**Promise problem**

**Input:**
- $G = (G, M)$ is a labelled $n$-cycle
- $M$ is a Turing machine

**Promise:**
- If $M$ halts in $s$ steps, then $n \geq s$

**Output:**
- yes if $M$ runs forever
- no if $M$ halts

**ID-oblivious**

**Impossible:** Must solve the Halting Problem

**LOCAL**

**Possible:** Node $v$ simulates $M$ for $\text{ID}(v)$ steps
Getting rid of the promise

**Promise:** • If $M$ halts in $s$ steps, then $n \geq s$
Getting rid of the promise

Promise:  \bullet \text{If } M \text{ halts in } s \text{ steps, then } n \geq s

\Downarrow \text{ Replace!} \Downarrow

Computation table of \( M \)

\subseteq G

\text{yes instance}
Getting rid of the promise

**Promise:** • If $M$ halts in $s$ steps, then $n \geq s$

\[ \Downarrow \text{Replace!} \Downarrow \]

Computation table of $M$

**Interesting bit:** Table needs to be obfuscated!
Summary

For local decision, we proved:

\[ \text{LOCAL} \neq \text{ID-obliviuous} \]
For local decision, we proved:

\[ \text{LOCAL} \neq \text{ID-obliviuous} \]

<table>
<thead>
<tr>
<th>Decision</th>
<th>IDs help</th>
<th>IDs don’t help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This work</td>
<td>[FHK OPODIS’12]</td>
</tr>
<tr>
<td>Search</td>
<td>[HHRS SIROCCO’12]</td>
<td>[NS Sicomp’95] [GHS PODC’12]</td>
</tr>
</tbody>
</table>
For local decision, we proved:

\[ \text{LOCAL} \neq \text{ID-obliviuous} \]

Randomisation?

- Open problems in randomized decision [FKPP DISC'12]
Summary

For local decision, we proved:

\[ \text{LOCAL} \neq \text{ID-obliviuous} \]

Randomisation?

- Open problems in \textit{randomised} decision [FKPP DISC’12]

Cheers!