Towards Polynomial Lower Bounds for Dynamic Problems

Gilad Hoch
by an article of Mihai Pătrașcu

25.3.2014

- **dynamic reachability:** maintain a data structure representing a directed graph which supports the following actions:
  1. *insert*(u, v) - insert an edge between u and v
  2. *delete*(u, v) - delete the edge between u and v
  3. *reachability*(u, v) - is there a directed path from u to v?
- **dynamic shortest paths:** maintain a data structure representing an undirected graph which supports the following actions:
  1. *insert*(u, v) - insert an edge between u and v
  2. *delete*(u, v) - delete the edge between u and v
  3. *length*(u, v) - what is the length of the shortest path from u to v?
- **subgraph connectivity:** preprocess undirected graphs to support the actions:
  1. *toggle*(u) - turn node u on/off
  2. *connectivity*(u, v) - is there a path between u and v that passes only in "on" nodes?
- **Langerman's problems:** maintain an array $A[1, \ldots, n]$ of integers, under the following actions:
  1. *update*(i, x) - $A[i] \leftarrow x$
  2. *zeroSum* - is there a k such that $\sum_{i=1}^{k} A[i] = 0$?
- **Pagh's problem:** preprocess a family of sets $X_1, X_2, \cdots \subseteq [n]$ to support the following actions:
  1. *create*(i, j) - create a new set $X_{new} \leftarrow A_i \cap A_j$
  2. *query*(i, z) - determine if $z \in A_i$
- **Erickson's problem:** preprocess an integers matrix M to support:
  1. *incrementRow*(M, i) - increment all values in row i
  2. *incrementCol*(M, i) - increment all values in column i
  3. *max*(M) - return the maximum value in the matrix

THE MULTIPHASE PROBLEM

phase I pre-process an input of k sets, $S_1, \dots, S_k \subset [n]$ in time $O(nk \cdot \tau)$
(build a data structure for the next phases)

phase II process yet another input of a set $T \subseteq [n]$ in time $O(n \cdot \tau)$
(update the data structure from phase I)

phase III answer query: given an index $i \in [k]$, determine if $S_i \cap T = \varnothing$ in time $O(\tau)$

THE MULTIPHASE PROBLEM

phase I pre-process an input of k sets, $S_1, \ldots, S_k \subset [n]$ in time $O(nk \cdot \tau)$
(build a data structure for the next phases)

phase II process yet another input of a set $T \subseteq [n]$ in time $O(n \cdot \tau)$
(update the data structure from phase I)

phase III answer query: given an index $i \in [k]$, determine if $S_i \cap T = \varnothing$ in time $O(\tau)$

Conjecture (1)

$\exists$ *constants* $\gamma > 1$, *and* $\delta > 0$
such that: $k = \Theta(n^\gamma) \implies \tau = \Omega(n^\delta)$

THE MULTIPHASE PROBLEM

phase I pre-process an input of k sets, $S_1, \dots, S_k \subset [n]$ in time $O(nk \cdot \tau)$
(build a data structure for the next phases)

phase II process yet another input of a set $T \subseteq [n]$ in time $O(n \cdot \tau)$
(update the data structure from phase I)

phase III answer query: given an index $i \in [k]$, determine if $S_i \cap T = \varnothing$ in time $O(\tau)$

Conjecture (1)

*$\exists$ constants $\gamma > 1$, and $\delta > 0$
such that: $k = \Theta(n^\gamma) \implies \tau = \Omega(n^\delta)$*

Theorem (2)

if the multiphase problem is hard (in the sense of the conjecture) $\implies$ for every problem we saw previously $\exists \epsilon > 0$ such that the problem cannot be solved with $O(N^\epsilon)$ per operation and $O(N^{1+\epsilon})$ pre-processing time.

THE 3SUM PROBLEM

input: a set S of n numbers

output: find distinct $x, y, z \in S$ such that $x + y = z$

THE 3SUM PROBLEM

input: a set S of n numbers

output: find distinct $x, y, z \in S$ such that $x + y = z$

THEOREM (3)

Under the 3SUM conjecture, the multiphase problem with $k = \Theta(n^{2.5})$ requires $\tau \geq n^{0.5-o(1)}$ on the word RAM.

THE 3SUM PROBLEM

input: a set S of n numbers

output: find distinct $x, y, z \in S$ such that $x + y = z$

CONJECTURE (6: 3SUM-HARDNESS)

$\forall S \subset \{-n^3, \ldots, n^3\}, |S| = n$
$\forall$ algorithm A for 3SUM,
To determine whether $\exists x, y, z \in S$ such that $x + y = z$ and x, y, z are distinct,
$O(n^{2-o(1)})$ time is needed in expectation for A with the input S.

Convolution 3SUM

input: an Array $A[1..n]$ of n integers

output: find 2 indexes $i \neq j$ such that $A[i] + A[j] = A[i + j]$

Convolution 3SUM

input: an Array $A[1..n]$ of n integers

output: find 2 indexes $i \neq j$ such that $A[i] + A[j] = A[i + j]$

Theorem (10)

If 3SUM requires $\Omega(n^2/f(n))$ expected time, convolution-3SUM requires $\Omega(n^2/f^2(n \cdot f(n)))$ expected time.

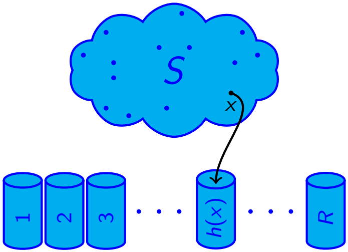
Convolution 3SUM

input: an Array $A[1..n]$ of n integers

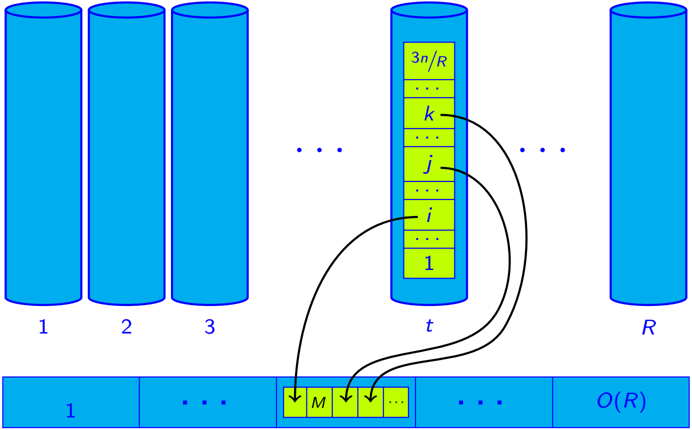
output: find 2 indexes $i \neq j$ such that $A[i] + A[j] = A[i + j]$

Theorem (10)

If 3SUM requires $\Omega({}^{n^2}/{}_{f(n)})$ expected time, convolution-3SUM requires $\Omega({}^{n^2}/{}_{f^2(n \cdot f(n))})$ expected time.



Convolution 3SUM



Hardness of Reporting Triangles

Lemma (11)

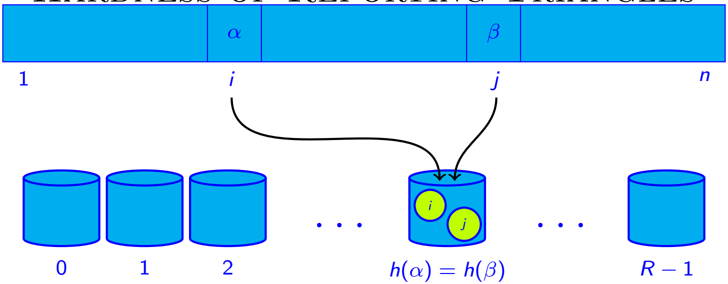
Assume Convolution-3SUM requires $\Omega(n^2/f(n))$ expected time, and let R be:

$$\omega(\sqrt{n \cdot f(n)}) < R < o(n/f(n))$$

. Then, $\Omega(n^2/f(n))$ expected time is needed to report $O(n^2/R)$ triangles in a tripartite graph where:

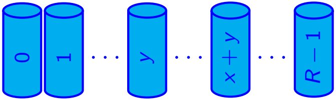
- *the three parts are A, B, C, of sizes $|A| = |B| = R\sqrt{n}$ and $|C| = n$;*
- *each vertex in $A \cup B$ has $O(n/R)$ neighbors in C;*
- *there are $O(nR)$ edges in $A \times B$.*

Hardness of Reporting Triangles



Hardness of Reporting Triangles

$x \leftarrow h(\alpha)$



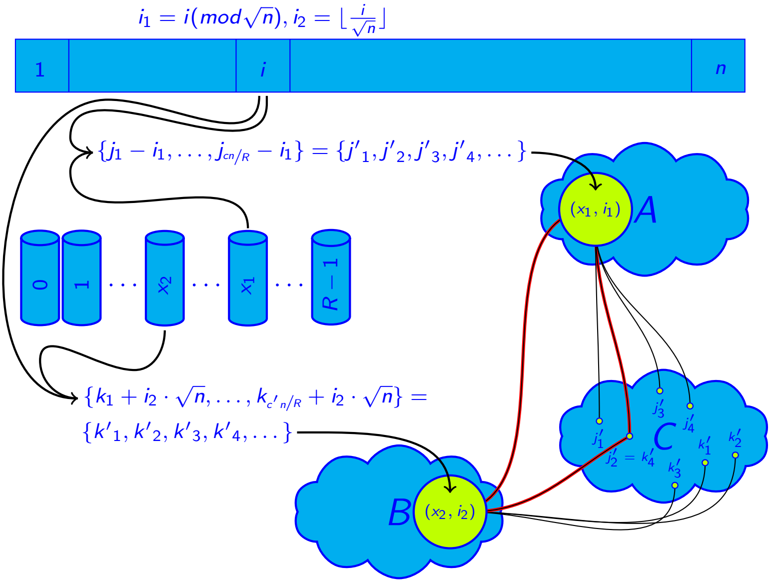
$\{j_1, \ldots, j_{cn/R}\}$

$T \leftarrow \{j_1 - i, \ldots, j_{cn/R} - i\}$

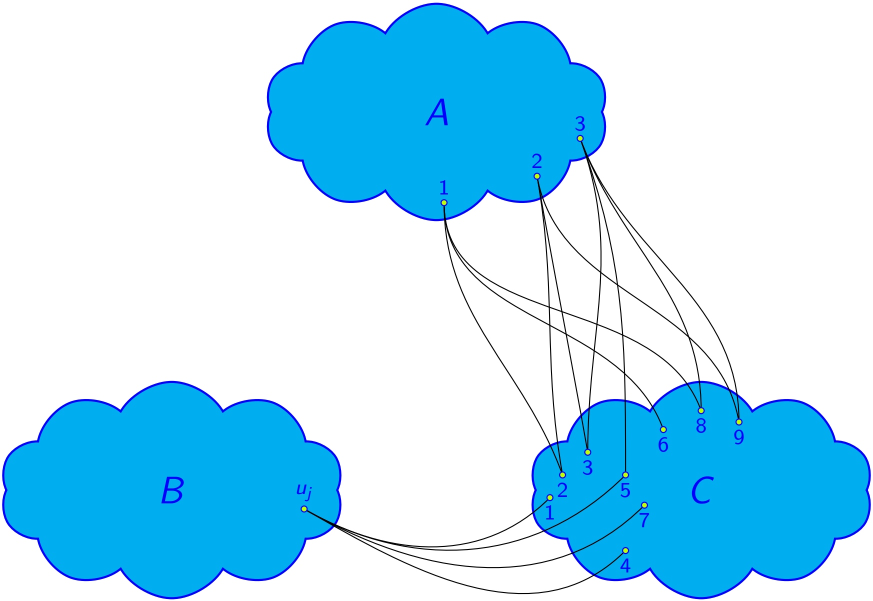
$\{j'_1, \ldots, j'_{c'n/R}\}$

$\{j'_1, \ldots, j'_{c'n/R}\} \cap T \neq \varnothing$

Hardness of Reporting Triangles

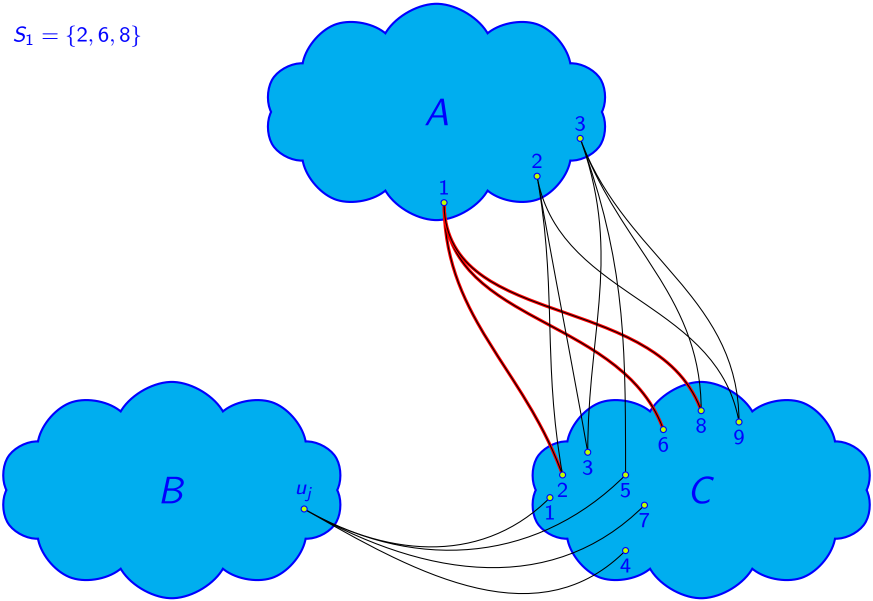


Reduction to the Multiphase Problem



Reduction to the Multiphase Problem

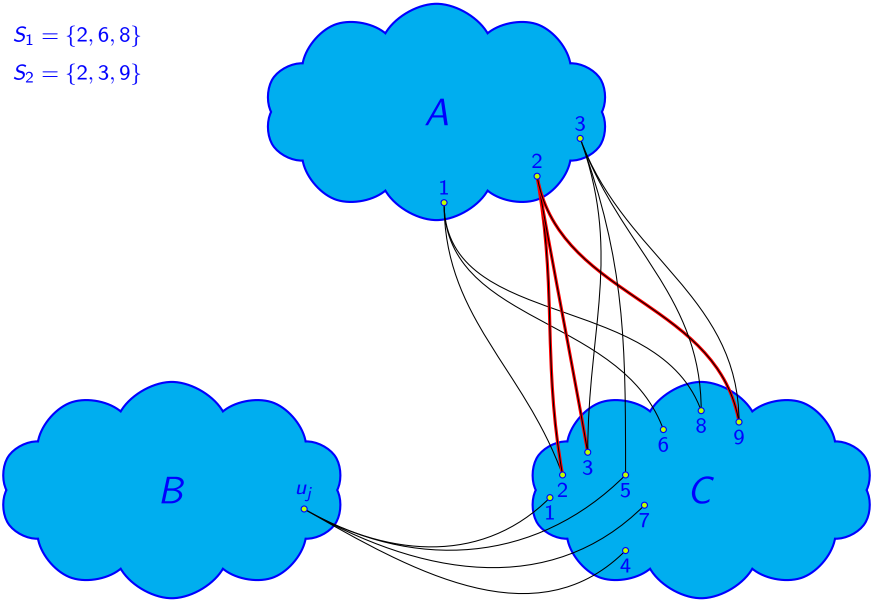
$S_1 = \{2, 6, 8\}$



Reduction to the Multiphase Problem

$S_1 = \{2, 6, 8\}$

$S_2 = \{2, 3, 9\}$



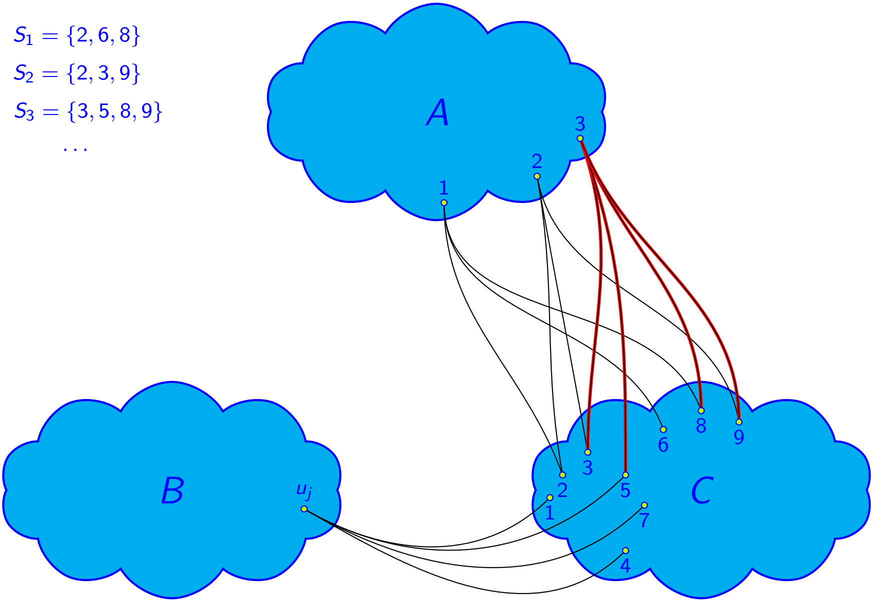
Reduction to the Multiphase Problem

$S_1 = \{2, 6, 8\}$

$S_2 = \{2, 3, 9\}$

$S_3 = \{3, 5, 8, 9\}$

$\dots$



Reduction to the Multiphase Problem

$S_1 = \{2, 6, 8\}$

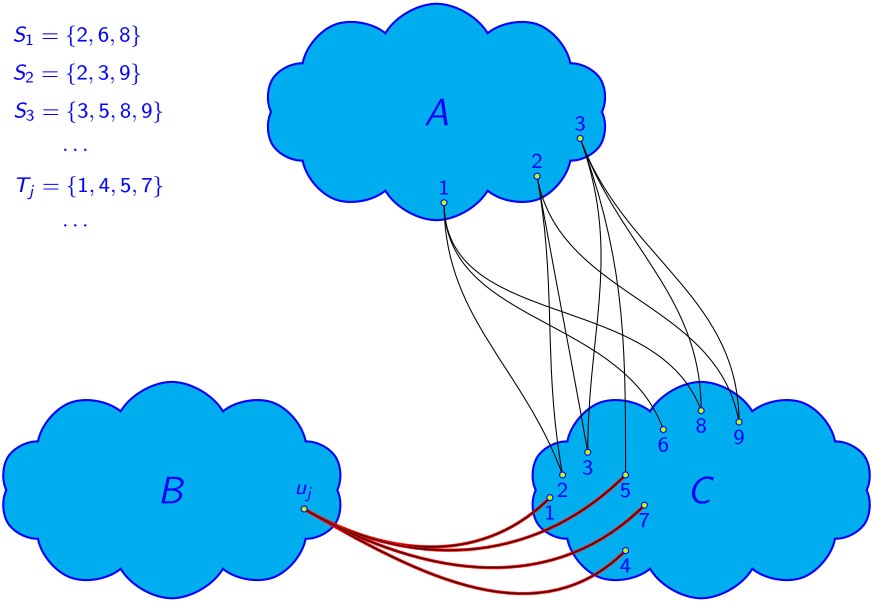
$S_2 = \{2, 3, 9\}$

$S_3 = \{3, 5, 8, 9\}$

$\dots$

$T_j = \{1, 4, 5, 7\}$

$\dots$



Reduction to the Multiphase Problem

$S_1 = \{2, 6, 8\}$

$S_2 = \{2, 3, 9\}$

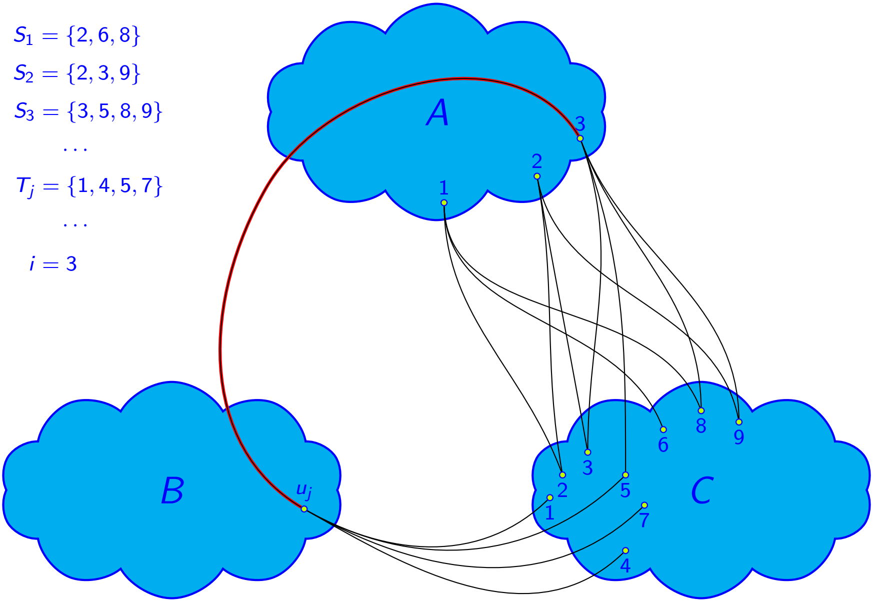
$S_3 = \{3, 5, 8, 9\}$

$\dots$

$T_j = \{1, 4, 5, 7\}$

$\dots$

$i = 3$



Reduction to the Multiphase Problem

$S_1 = \{2, 6, 8\}$

$S_2 = \{2, 3, 9\}$

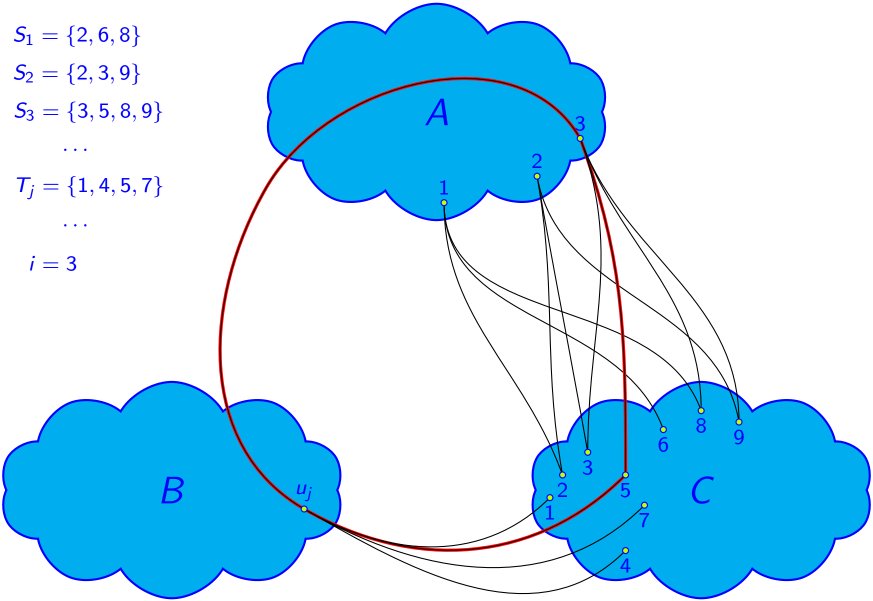
$S_3 = \{3, 5, 8, 9\}$

$\dots$

$T_j = \{1, 4, 5, 7\}$

$\dots$

$i = 3$



Questions?