Self-Adjusting Linear Networks

with Chen Avin* and Stefan Schmid†

*Ben Gurion University, †University of Vienna

Ingo van Duijn

Department of Computer Science
Aalborg University
Denmark
Baby Steps towards
Self-Adjusting Linear Networks
with Chen Avin* and Stefan Schmid†
*Ben Gurion University, †University of Vienna

Ingo van Duijn
Department of Computer Science
Aalborg University
Denmark
Overview

Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures
Overview

Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

Case study:
- List Access
- List Communication
Overview

Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

Case study:
- List Access
- List Communication

Results:
- Grid Network Bounds
Overview

Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

Case study:
- List Access
- List Communication

Results:
- Grid Network Bounds

Lower Bound Proof Sketch
Overview

Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

Case study:
- List Access
- List Communication

Results:
- Grid Network Bounds

Lower Bound Proof Sketch

Future Work
Self-Adjusting Networks
Self-Adjusting Networks

The Setting:
- Network
Self-Adjusting Networks

The Setting:
- Network
- Communication
Self-Adjusting Networks

The Setting:
- Network
- Communication

Buongiorno
Self-Adjusting Networks

The Setting:
- Network
- Communication
Self-Adjusting Networks

The Setting:
- Network
- Communication
- Adjustments
Self-Adjusting Networks

The Setting:
- Network
- Communication
- Adjustments
- Online

Buonasera
Self-Adjusting Networks

The Setting:
- Network
- Communication
- Adjustments
- Online

\{ \text{Cost!} \}

The Goal:
- Minimise Cost
Self-Adjusting Networks

The Setting:
- Network
- Communication
- Adjustments
- Online

\{ Cost! \}

The Goal:
- Minimise Cost
- (Distributed Algorithm)
Self-Adjusting Networks

The Setting:
- Network
- Communication
- Adjustments
- Online

\{ Cost! \}

The Goal:
- Minimise Cost
- (Distributed Algorithm)

The Questions:
- Which Model?
The Setting:
- Network
- Communication
- Adjustments
- Online

Cost!

The Goal:
- Minimise Cost
- (Distributed Algorithm)

The Questions:
- Which Model?
- Formal Guarantees?
Dynamic Dictionaries
Dynamic Dictionaries

The Setting:
- Keys
Dynamic Dictionaries

The Setting:
- Keys (with data)
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
The Setting:
- Keys
- Pointer Structure
- Key Access

Access 3
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:
- Dynamic Optimality
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:
- Dynamic Optimality
- Minimise: Best Online Cost
- Best Offline Cost
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:
- Dynamic Optimality
- Minimise: Best Online Cost, Best Offline Cost

Towards Networks:
- Pairwise Access
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:
- Dynamic Optimality
- Minimise: Best Online Cost
- Best Offline Cost

Towards Networks:
- Pairwise Access

Access 6
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:
- Dynamic Optimality
- Minimise: Best Online Cost
- Best Offline Cost

Towards Networks:
- Pairwise Access
Dynamic Dictionaries

The Setting:
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:
- Dynamic Optimality
- Minimise: Best Online Cost
  Best Offline Cost

Towards Networks:
- Pairwise Access
- Ignore Searching/Routing
A Simple? Network
A Simple Network

The Dictionary:
- Linked List
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
A Simple Network

The Dictionary:
- Linked List
- Access in Front

Access Cost

Access 5

Ingo van Duijn
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm

Access 5

Move Cost
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm

Access 5
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:
- Pairwise Access
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access

Communicate (2,7)
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:
- Pairwise Access

Communicate (2,7)

Move Cost
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Communicate (2,7)
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Communicate (2,7)
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Communicate (2,7)
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

(5, 8)
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

(5, 8)
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

(5, 7)

(5, 8)
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

(5, 7)

(5, 8)
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

(5, 6)
(5, 7)
(5, 8)
A Simple Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

- (5, 4)
- (5, 6)
- (5, 7)
- (5, 8)
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

(5, 3)
(5, 4)
(5, 6)
(5, 7)
(5, 8)
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

1. (5, 2)
2. (5, 3)
3. (5, 4)
4. (5, 6)
5. (5, 7)
6. (5, 8)
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire

Communication:

5
- (5, 1)
- (5, 2)
- (5, 3)
- (5, 4)
- (5, 6)
- (5, 7)
- (5, 8)
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire
- Wheel can be Everywhere
A Simple? Network

The Dictionary:
- Linked List
- Access in Front
- Move-to-Front Algorithm → Dynamically Optimal!

The Network:
- Pairwise Access
- Move-to-Where?

Not So Simple:
- Co-locating can Backfire
- Wheel can be Everywhere
- Same Conclusion by Olver et al.
Results
Results

Bounds

Communication Complexity
Results

Grid Networks:
■ Bounds on Competitive Ratio

Communication Complexity

Bounds

$\log n$
Results

Grid Networks:
- Bounds on Competitive Ratio
- Restricted Communication

Communication Complexity

Linear Demand

Bounds

\[ \log n \]

Communication Complexity
Results

Grid Networks:
- Bounds on Competitive Ratio
- Restricted Communication

Communication Complexity

Bounds

Log $n$

Linear Demand

Lower Bound

Communication Complexity
Results

Grid Networks:
- Bounds on Competitive Ratio
- Restricted Communication

![Diagram showing bounds on communication complexity and linear demand](image-url)
Results

Grid Networks:
- Bounds on Competitive Ratio
- Restricted Communication

![Bounds vs Communication Complexity](image)

- Linear Demand
  - Upper Bound
  - Lower Bound

Log $n$ vs Communication Complexity
Results

Grid Networks:
- Bounds on Competitive Ratio
- Restricted Communication

![Graph]
- Linear Demand
- Linear Demand
- Upper Bound
- Lower Bound
- Communication Complexity
- \( \log n \)
Results

Grid Networks:
- Bounds on Competitive Ratio
- Restricted Communication

![Diagram showing bounds on linear demand as a function of communication complexity. The x-axis represents communication complexity, ranging from 0 to log n, and the y-axis represents bounds. There are two distinct regions: an upper bound and a lower bound. The graph shows a transition point labeled with a question mark.](image-url)
Results

Grid Networks:
- Bounds on Competitive Ratio
- Restricted Communication

Line Networks:
- Distributed Implementation

Diagram:
- Linear Demand vs. Communication Complexity
  - Upper Bound
  - Lower Bound
  - Log n
Lower Bound
Lower Bound
Lower Bound
Lower Bound

\[ \sigma = \]

\begin{align*}
&1 & 2 \\
&6 & 7 & 8 \\
&3 & 9 \\
&5 & 4
\end{align*}
Lower Bound

$\sigma = (7, 8)$
Lower Bound

$\sigma = (7, 8)(4, 9)$
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6) \]
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6) \]
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6) \]
\( \sigma = (7,8)(4,9)(1,6)(4,9) \)
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6)(4, 9) \]
Lower Bound

\[ \sigma = (7,8)(4,9)(1,6)(4,9) \]
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6)(4, 9)(1, 7) \]
Lower Bound

\[ \sigma = (7,8)(4,9)(1,6)(4,9)(1,7) \]
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6)(4, 9)(1, 7) \]
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6)(4, 9)(1, 7)(7, 8) \]
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6)(4, 9)(1, 7)(7, 8) \]
Lower Bound

\[ \sigma = (7, 8)(4, 9)(1, 6)(4, 9)(1, 7)(7, 8) \]
Lower Bound

The Strategy:
- Exploit Bad Edges

\[ \sigma = (7, 8)(4, 9)(1, 6)(4, 9)(1, 7)(7, 8) \]
Lower Bound

The Strategy:
- Exploit Bad Edges
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching

- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching

The Basic Math:
- $\log n$ Bad Matchings
- $n^2$ Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$
- Competitive Ratio: $\Omega(\log n)$
The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:
- $\log n$ Bad Matchings
- $n^2$ Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$
  → Competitive Ratio: $\Omega(\log n)$
Lower Bound

The Strategy:
■ Exploit Bad Edges
■ Introduce Bad Matching
■ Maintain List Graph

The Basic Math:
■ $\log n$ Bad Matchings
■ $n^2$ Cost per Matching
■ Online Cost: $\Omega(n^2 \log n)$
■ Offline Cost: $\Theta(n^2)$
→ Competitive Ratio: $\Omega(\log n)$
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:
- \( \log n \) Bad Matchings
- \( n^2 \) Cost per Matching
- Online Cost: \( \Omega(n^2 \log n) \)
- Offline Cost: \( \Theta(n^2) \)
  \[ \rightarrow \text{Competitive Ratio: } \Omega(\log n) \]

\( n^2 \) per Matching:
- Quantify Distortion of Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:
- \( \log n \) Bad Matchings
- \( n^2 \) Cost per Matching
- Online Cost: \( \Omega(n^2 \log n) \)
- Offline Cost: \( \Theta(n^2) \)
  \( \rightarrow \) Competitive Ratio: \( \Omega(\log n) \)

\( n^2 \) per Matching:
- Quantify Distortion of Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:
- \( \log n \) Bad Matchings
- \( n^2 \) Cost per Matching
- Online Cost: \( \Omega(n^2 \log n) \)
- Offline Cost: \( \Theta(n^2) \)
  \( \rightarrow \) Competitive Ratio: \( \Omega(\log n) \)

\( n^2 \) per Matching:
- Quantify Distortion of Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:
- $\log n$ Bad Matchings
- $n^2$ Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$
  → Competitive Ratio: $\Omega(\log n)$

$n^2$ per Matching:
- Quantify Distortion of Matching
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:
- \( \log n \) Bad Matchings
- \( n^2 \) Cost per Matching
- Online Cost: \( \Omega(n^2 \log n) \)
- Offline Cost: \( \Theta(n^2) \)
  \( \rightarrow \) Competitive Ratio: \( \Omega(\log n) \)

\( n^2 \) per Matching:
- Quantify Distortion of Matching

\[
\begin{align*}
2 \times 1 + 0 \times 0 &= 2 \\
\uparrow \\
(2, 0) \\
\downarrow \\
1 \times 1 + 0 \times 1 &= 1
\end{align*}
\]
Lower Bound

The Strategy:
■ Exploit Bad Edges
■ Introduce Bad Matching
■ Maintain List Graph

The Basic Math:
■ \( \log n \) Bad Matchings
■ \( n^2 \) Cost per Matching
■ Online Cost: \( \Omega(n^2 \log n) \)
■ Offline Cost: \( \Theta(n^2) \)
→ Competitive Ratio: \( \Omega(\log n) \)

\( n^2 \) per Matching:
■ Quantify Distortion of Matching

\[
\begin{align*}
2 \times 1 + 0 \times 0 &= 2 \\
1 \times 1 + 0 \times 1 &= 1 \\
\sum &= 3
\end{align*}
\]
Lower Bound

The Strategy:
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:
- $\log n$ Bad Matchings
- $n^2$ Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$
  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

$n^2$ per Matching:
- Quantify Distortion of Matching
- Compute Sum of Distortion of all Matchings
- Average Distortion is Sufficient

\[2 \times 1 + 0 \times 0 = 2\]
\[\uparrow\]
\[1 \times 1 + 0 \times 1 = 1\]
\[\downarrow\]
\[\sum = 3\]
Open Questions & Future Work

Better Bounds for Line Networks:
- Nontrivial Upper Bound?
- Better Lower Bound?
- Dynamic Offline Algorithms?
Better Bounds for Line Networks:
- Nontrivial Upper Bound?
- Better Lower Bound?
- Dynamic Offline Algorithms?

$O(n/\log n)$?
Open Questions & Future Work

Better Bounds for Line Networks:
- Nontrivial Upper Bound?
- Better Lower Bound?
- Dynamic Offline Algorithms?

Different Networks:
- Binary (Splay) Trees
- No BST Property

\[ O(n / \log n) \]