Dynamic Graph Algorithms

CSL-531

2026

Introduction

Dr. Shahbaz Khan
Department of Computer Science and Engineering,
Indian Institute of Technology Roorkee
shahbaz.khan@cs.iitr.ac.in

Dynamic Graph Algorithms

Offline algo

Dynamic Graphs: Graphs that keeps **changing** with time.

Graph Updates: **Online** sequence of *insertion / deletion* of *edges / vertices.*

Dynamic Graph Algorithms

Design and analysis of algorithms for dynamic graphs.

Goal: Maintain a data structure or report query after each update.
Eg. DFS Tree, d(s,t)

Trivial: Recompute from scratch after every update using static algorithm.

Aim: Attempt to **reuse previous computation** for computing solution,
to **update much faster** than static.

Graph Input

Algo

Static

Output DFS Tree

$d(s,t)$

$G, e_1, e_2, e_3 \ldots e_k$

G

$G + \{e_1, e_2, e_3\}$

Static

Motivation

$d(s,t) \Rightarrow [O(m + n\log n)]$

Graph

Motivation

$d(s,H) \Rightarrow [O(m + n\log n)]$

$\log n$

- Most graphs in real world are **dynamic**.
  Eg. Social Network, Road Network, Internet Network, Genome Data

  Size of data

  TB
- Recomputing solutions on such graphs from scratch **impractical**.
  Eg. Shortest Paths, Connectivity, Centers
- **Theoretical SIgnificance**
  - Serve as black box in other algorithms
  - Related to Complexity of Parallel Algorithms
  - Related to Complexity of Hard problems in P

APSP $\Omega(n^3)$

Verifiable Poly

NP Hard

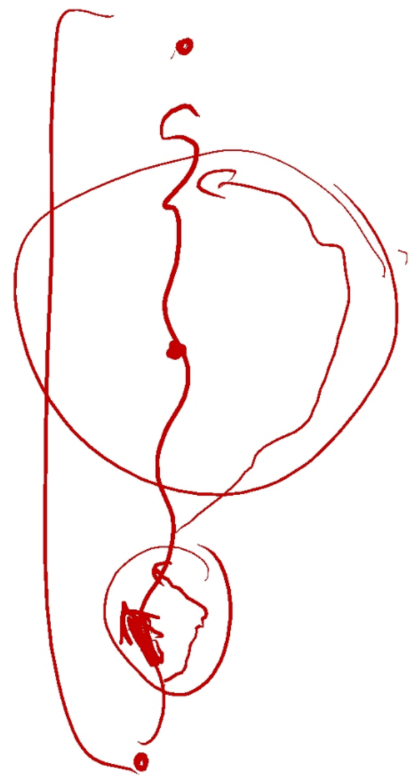
Poly

$O(n^3) \Rightarrow \Omega(n^3)$

Graph

nodes

relationships



Intended Learning Outcomes

Intended Learning Outcomes

Dynamic Graph Algorithms (CSL-531)

- ***Design*** dynamic graph algorithms using standard design techniques, and ***prove*** its correctness.

- ***Analyze*** dynamic graph algorithms using standard analysis techniques, and ***prove*** its tightness.

- ***Prove*** the hardness of a given dynamic graph problem, by reducing it to standard hard problems.

Analysis WC Eg.

$A \Rightarrow O(n^2)$

$O(m)$ $\Omega(m)$ $\Rightarrow \Theta(m)$

Course Modules

Sl. No.	Contents	Contact Hours
1.	**Introduction:** Terminology, Definitions, Models, classical problems	**2**
2.	**Amortizations Techniques:** Doubling, Monotonicity, Bounded Potential, Case studies: Partially dynamic MIS, BFS, Incremental Connectivity, Reachability.	**8**
3.	**Randomization Techniques:** Random Sampling, Hitting Sets, Probability tools, Case studies: Dynamic Matching, Decremental Connectivity, Fully Dynamic MIS.	**8**
4.	**Hierarchical Decomposition Techniques:** Decomposition, Multilevel decomposition, Case studies: Dynamic Matching, Fully Dynamic Connectivity, Shortest Paths.	**8**
5.	**Primal Dual Techniques:** Definition, Applications in Approximation algorithms, Case studies: Dynamic Vertex Cover, Fractional Matching, Coloring.	**8**
6.	**Fine Grained Complexity:** SETH, Polynomial Complexity Conjectures, Reductions, Case studies: Shortest Paths, Connectivity, Reachability, Matching, Diameter.	**4**
7.	**Miscellaneous Topics:** State of the Art in DFS Trees, Maximal Independent Sets.	**4**
	Total	**42**

Course Structure
and
Assessment Criteria

Assessments

- Mid Term Exam: 30% Marks
- End Term Exam: 40% Marks
  - Written Exam
- Class Work Sessional: 30% Marks
  - Quiz
  - Tutorials
  - Project

OPEN NOTES

WARNING

CHEATING and USE of UNFAIR MEANS

ENTIRE COMPONENT (Question/Assignment/Exam) marked ZERO for BOTH parties

REPEAT OFFENDER dealt with HARSHLY

Guidelines for interactions

- Accepting different accents, writing styles.
- Always be respectful and do not offend others.
- For disagreements, give constructive feedback.
- It is OKAY to give wrong answers.
- Inappropriate/disrespectful behaviour will have consequences.

Email: Start subject with ***[CSL531]***
MS Teams: Message me directly
Office Hours: Class days evening 3-4pm
(email in advance before coming)

Dynamic Graph Model

Given a graph **G(V,E)** that changes with time by an online sequence of updates**.**

- **Updates**
  - Insertion of Edges
  - Insertion of Vertices
  - Deletion of Edges
  - Deletion of Vertices

Inc. weight

Dec. "

e x g

Shortest Path

40

100

90

- **Goal**

  Report the **solution** or maintain the **data structure** after every update.
  - Connected, Distance(s,t), Shortest Path Tree, BFS Tree?

- **Classification:**
  - **Partially Dynamic:** Have either insertion or deletions

    Incremental (only insertions), *Decremental* (only deletions)
  - **Fully Dynamic:** Both insertions and deletions
  - **Others:** Fault Tolerant, Subgraph, Oracle

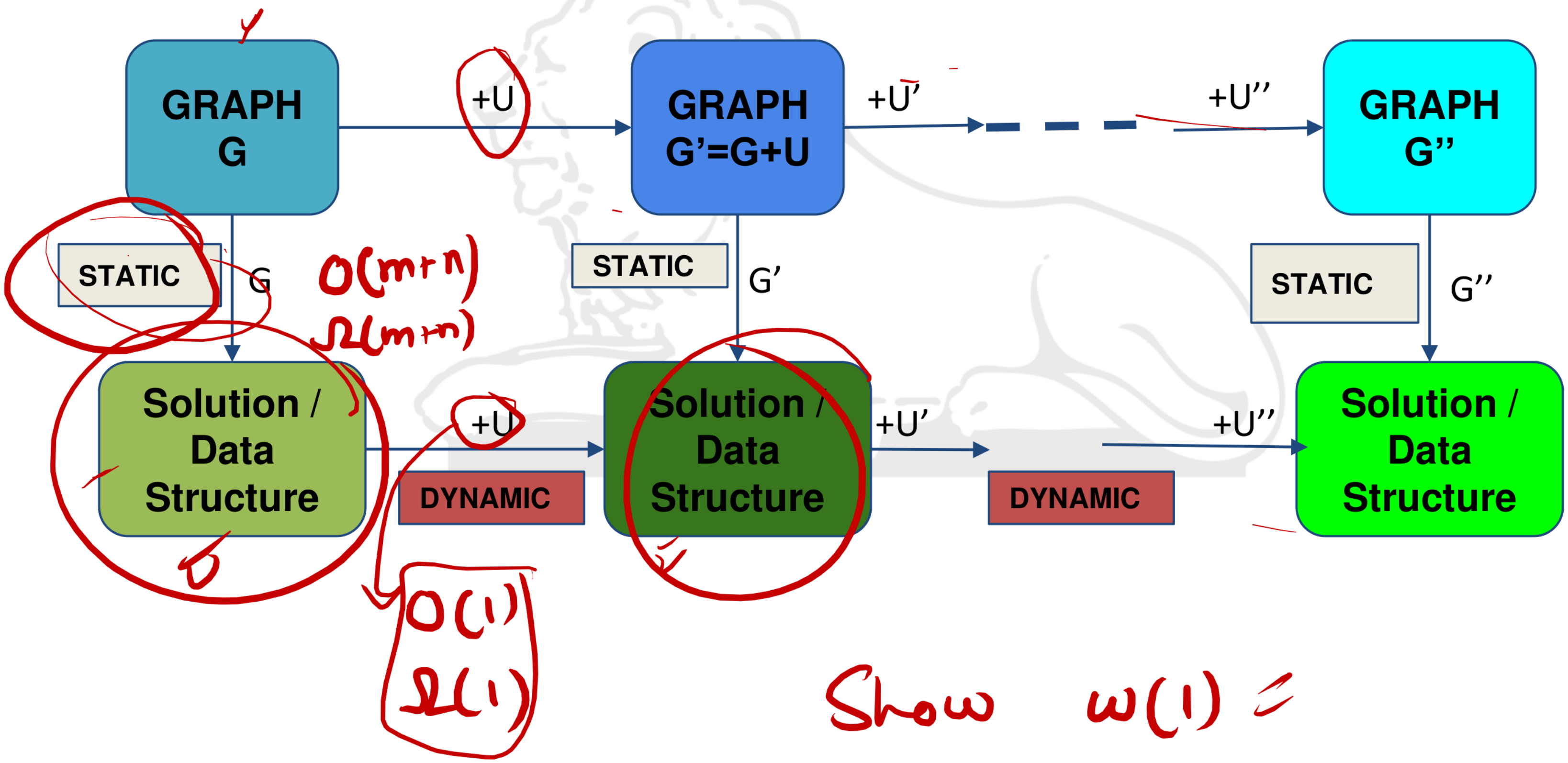
Inc. Connectivity DSU

Dynamic Graph Model

m edges n vertices

Given a graph **G(V,E)** that changes with time by an **online sequence** of updates**.**

- **Updates:** Insertions/Deletions of Edges/Vertices
- **Goal:** Report the **solution** or maintain the **data structure** after every update.



NOTE: Difference in size of input for static and dynamic algorithms.

Time Complexity

$G \xrightarrow{U}$ Partial

Q O(1)

Static

T

- **Preprocessing Time:** Initial time to preprocess the graph.
  - Shortest Path Tree from s
- **Update Time:** Time taken after each update to reconstruct data structure
  - ***Max:*** O(T)
  - ***Min:*** O(1)

Q O(1)

- **Query Time:** Time taken to answer query on the updated structure
  - ***Max:*** O(T)
  - ***Min:*** O(1)

$G \xrightarrow{U}$ Update O(n)

Query O(1)

Eager

Upd O(log n)

Query O(log n)

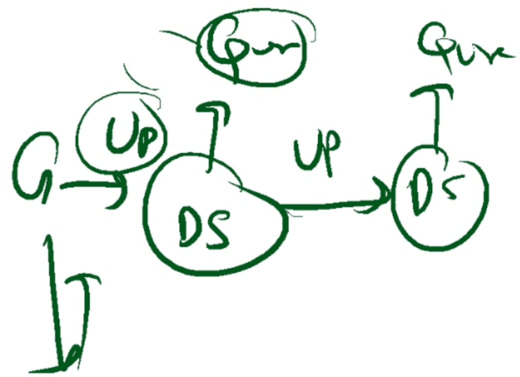
Upd O(1)

Qu O(n)

Lazy

Different tradeoffs for **Update** and **Query** time.

Worst-Case Bound vs **Amortized Bounds.**



S

DSU

$\boxed{m \text{ update}}$ $O(\alpha)$

$$\frac{\text{Total } T}{\# \text{ Updates}} = \frac{n\alpha}{m}$$

$O\left(\frac{n\alpha}{m}\right)$ Amortized $\Rightarrow O(1)$

$m = \Omega(n\alpha)$

Initially Disconnected

e_1

m

$n-1$ perform changes

$O(\alpha)$

$n-1$

Spanning tree.

Amortized time

$$\frac{\text{Total time}}{\text{Total Number of Updates}}$$

Limitations and Bounds

For a given problem P having best static algo time Ts(n) $\Rightarrow O(n^3)$
Dynamic Algorithm Td(n):

$O(mn)$

Upper bound:

max(update,query) = Td(n) $= O(T_s(n))$

$O(m^2)$ ×

Lower bound:

max(update,query) = Td(n) $\sqrt{n}$

Incremental, Decremental, Fully Dynamic?

$\Omega(T_s/m) \leftarrow \frac{mn}{m} = \Omega(n)$

WC ✓

Amortized ✓

Dynamic Algorithm for *NP Complete Problem* in **P**?

$G' = \phi$ $O(1)$

+e → $\sqrt{n}$

+e → $\sqrt{n}$

→ $\sqrt{n}$

m → $\sqrt{n}$

→ $\sqrt{n}$

G → $m\sqrt{n}$

Classical Problem

Incremental Reachability

Given a graph under edge insertions, maintain all vertices reachable from s

What is optimal update and query time?

Static

$O(m+n)$

Dynamic

$O(1)$ time ↳ Ω

Amortized?

WC. time?