UNIVERSITÉ FRANÇOIS - RABELAIS TOURS

LI Laboratoire d'Informatique EA 2101

AnyTime and Distributed Approaches for Graph Matching

Presented by: Zeina Abu-Aisheh

Wednesday, May 25 2016

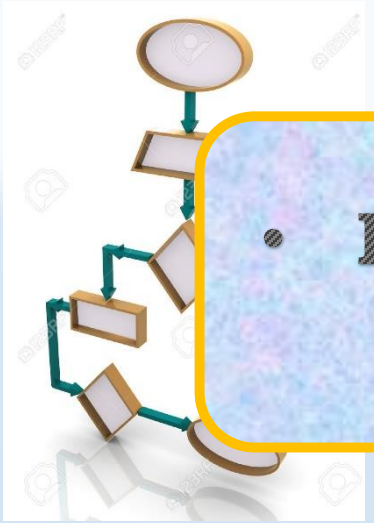
Supervised by:
Prof. Jean-Yves Ramel, Dr. Romain Raveaux and Prof. Patrick Martineau

External Jury Members:

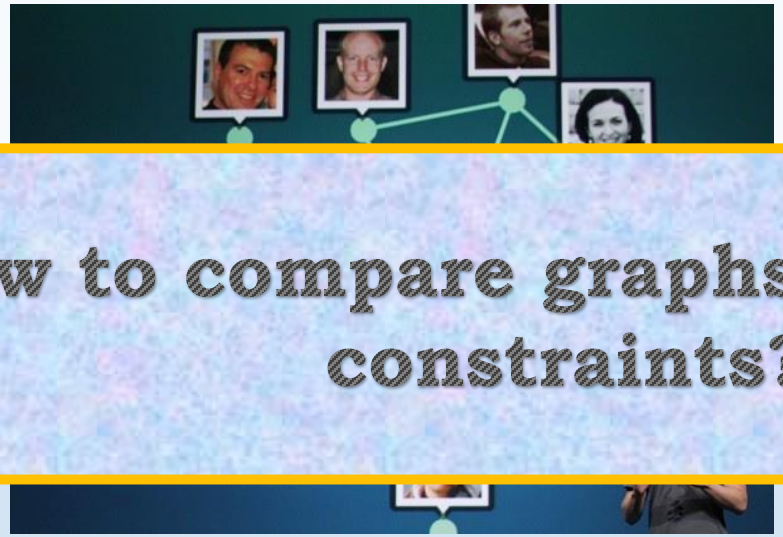
Prof. Christine Solnon — Prof. Luc Brun

Prof. Francesc Serratosa — Dr. Kasper Riesen

Surrounded by Graphs



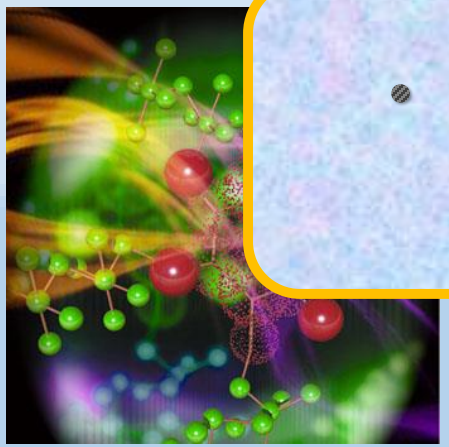
Program Flows



Social Networks

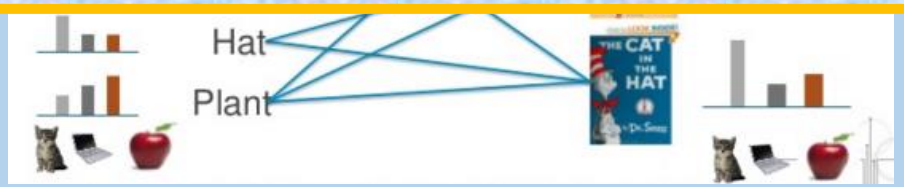
Images

- How to compare graphs under time constraints?

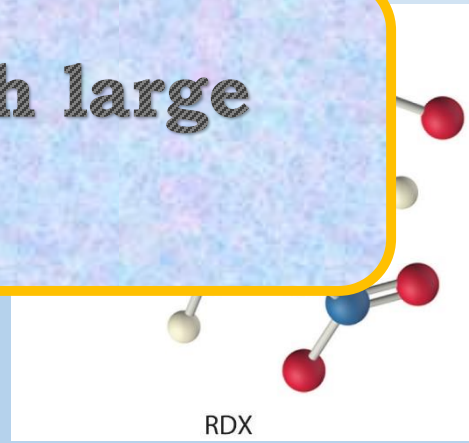


Biological Networks

- How to scale up to match large graphs?



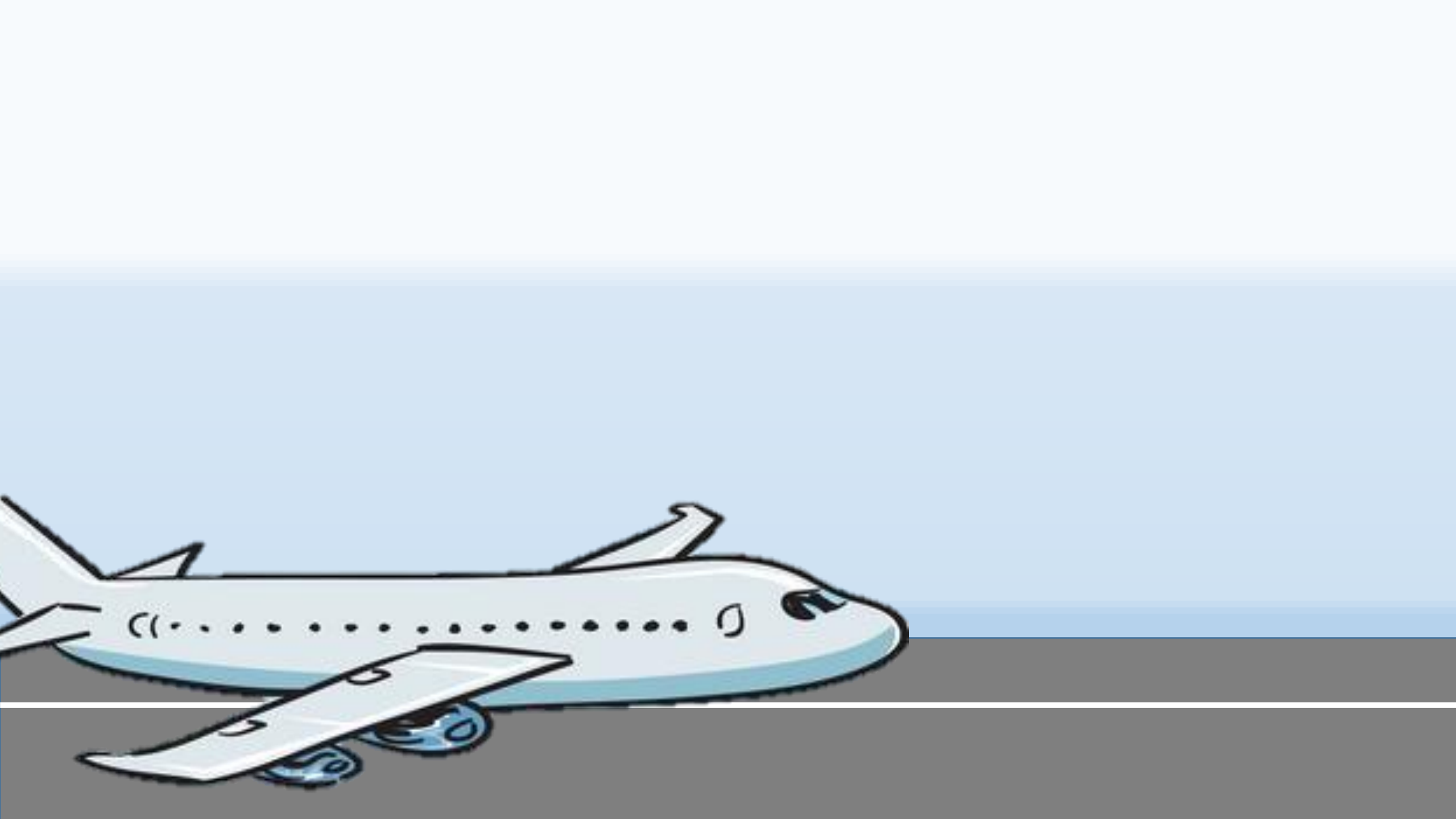
Machine Learning



Chemical Compounds

Outline

- **Introduction**
- **Strength and Weakness of Graph Matching Methods**
- **Contribution1: Optimization Techniques**
- **Contribution2: Metrics and Datasets for Performance Evaluation**
- **Experiments**
- **Conclusion**



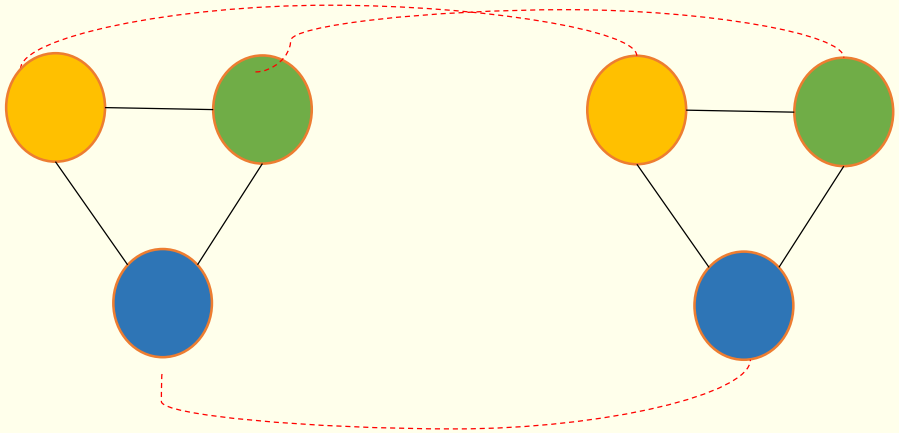
Outline

- **Introduction**
  - **Graph Matching Problems**
- **Strength and Weakness of Graph Matching Methods**
- **Contribution 1: Optimization Techniques**
- **Contribution 2: Metrics and Datasets for Performance Evaluation**
- **Experiments**
- **Conclusion**

Exact Graph Matching

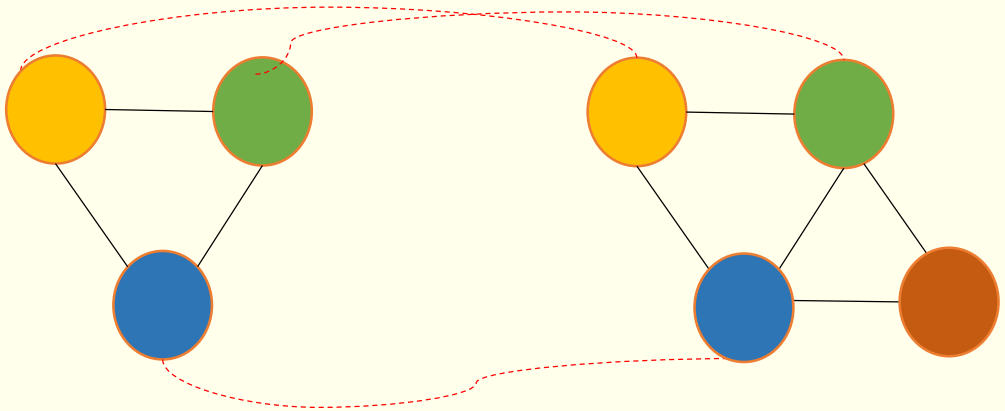
Graph Isomorphism

Graph G_1 Graph G_2



Subgraph Isomorphism

Graph G_1 Graph G_2



- Strict correspondances
  - ✓ Edge-preserving
- Too strict not used in Pattern Recognition

Error-Tolerant Graph Matching

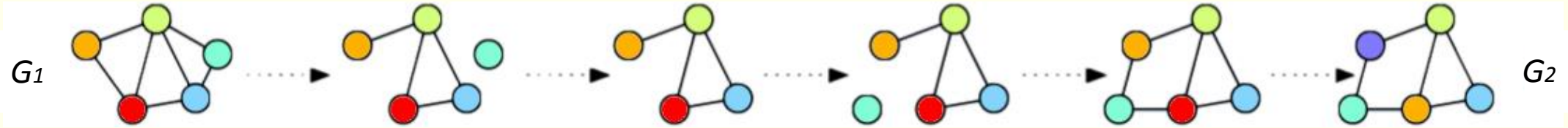
- Graphs of different topology/structures and attributes can be matched
- Attributes penality
- Structure penality

- NP-complete
- Graph Isomorphism "open problem"

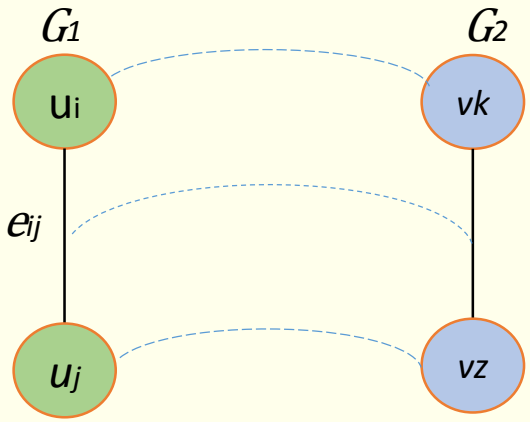
Graph Edit Distance (GED)

Let $G_1 = (V_1, E_1, L_{V_1}, L_{E_1}, \mu_1, \zeta_1)$ and $G_2 = (V_2, E_2, L_{V_2}, L_{E_2}, \mu_2, \zeta_2)$ be two graphs, the graph edit distance between G_1 and G_2 is defined as:

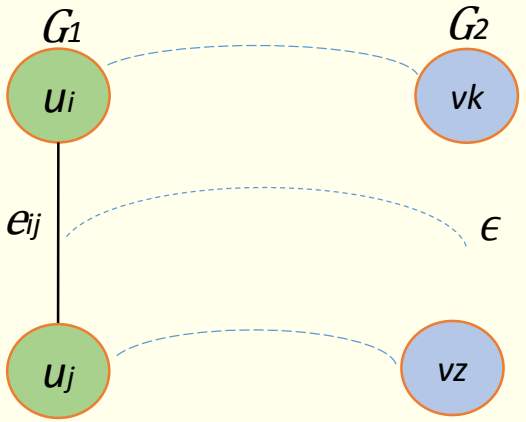
$$d_{\lambda_{min}}(G_1, G_2) = \min_{\lambda \in \gamma(G_1, G_2)} \sum_{ed_i \in \lambda} c(ed_i)$$



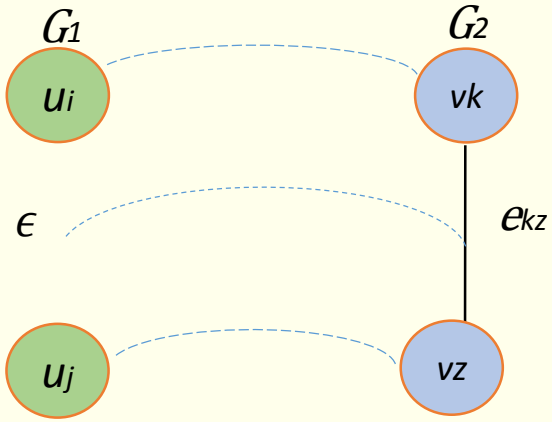
Edge Mappings based on their Adjacent Vertices



Edges Substitution



Edges Deletion



Edges Insertion

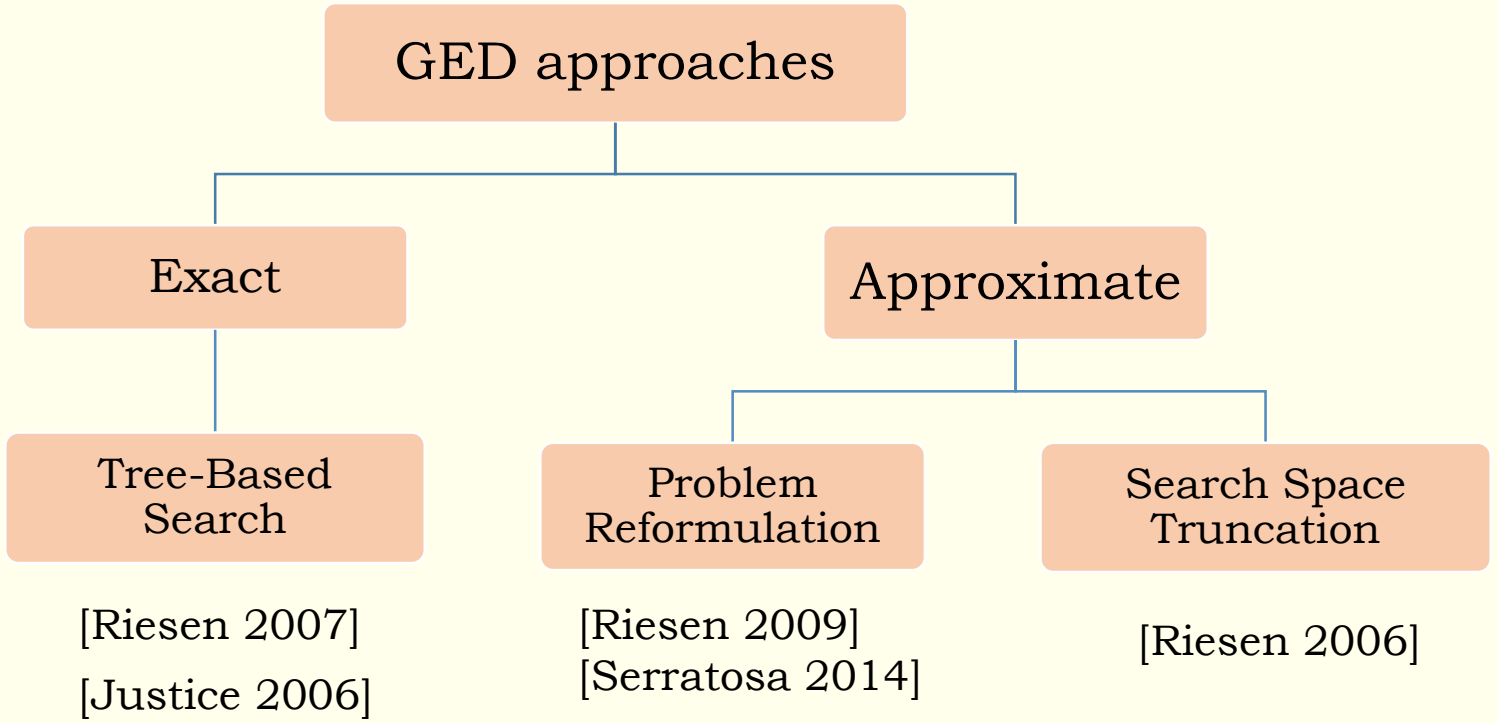
Focus on Graph Edit Distance (GED)

- **Cost Functions:**
  - ✓ GED is generic and compatible with graphs labeled with numeric and/or symbolic attributes
  - ✓ GED can be a metric (under constraints)
  - ✓ GED can be turned into the Maximum Common Subgraph problem

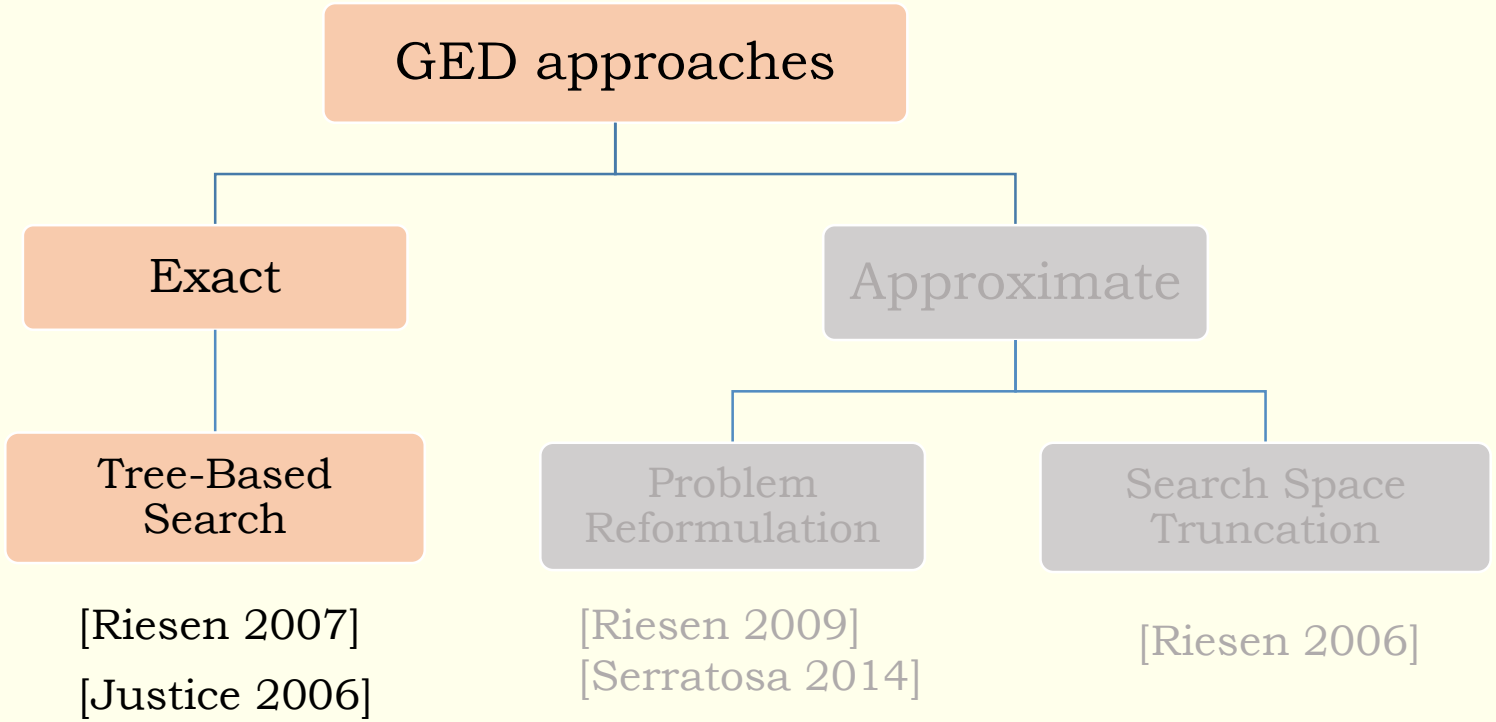
Outline

- Introduction
- **Strength and Weakness of Graph Matching Methods**
  - **Graph Edit Distance Methods**
  - **Graph Matching Repositories and their performance evaluation metrics**
  - **Limitations of existing methods**
- Strength and Weakness of Graph Matching Methods
- Optimization Techniques
- Metrics and Datasets for Performance Evaluation
- Experiments
- Conclusion

Graph Edit Distance Approaches



Graph Edit Distance Approaches

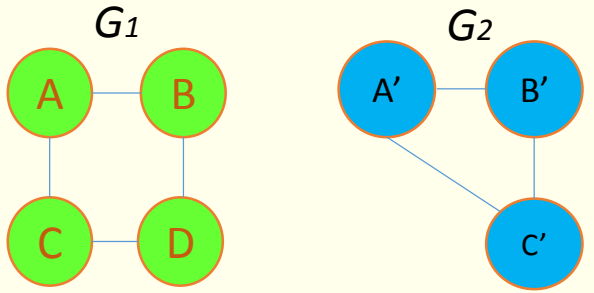


Exact Graph Edit Distance Approaches

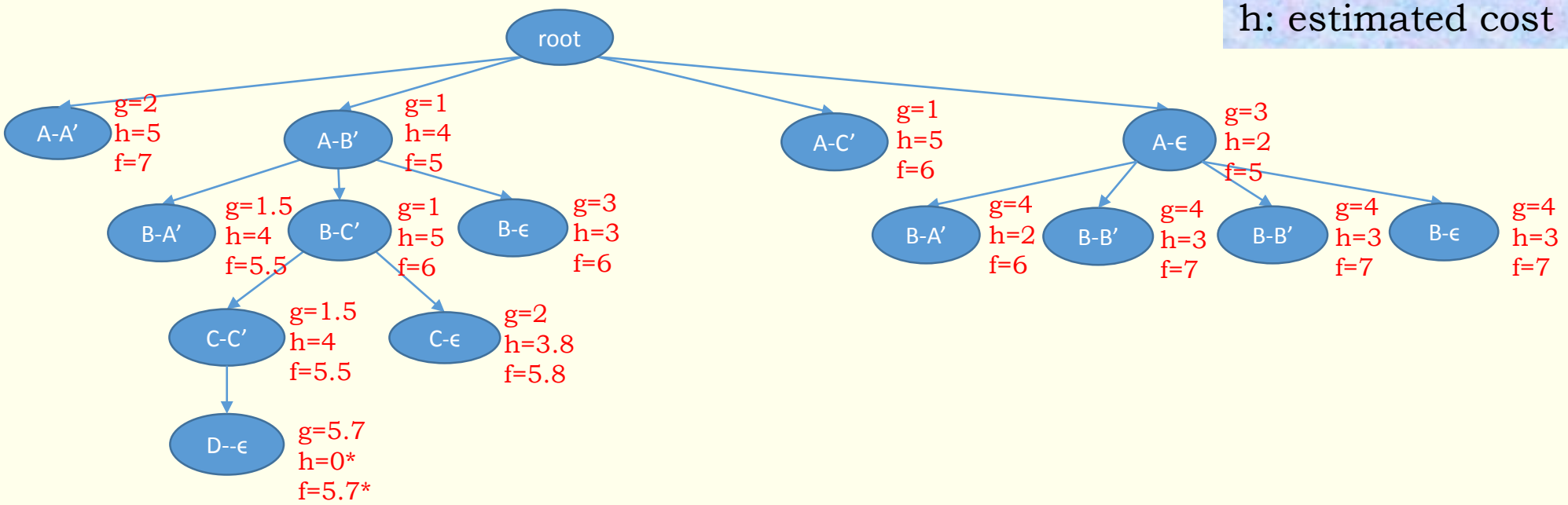
- **Tree-Based Search Approaches**

Exact Graph Edit Distance Approaches

- **Tree-Based Search Approaches**
  - ✓ A* algorithm: [Riesen 2007]

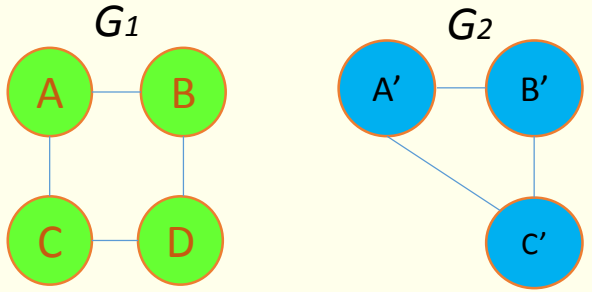


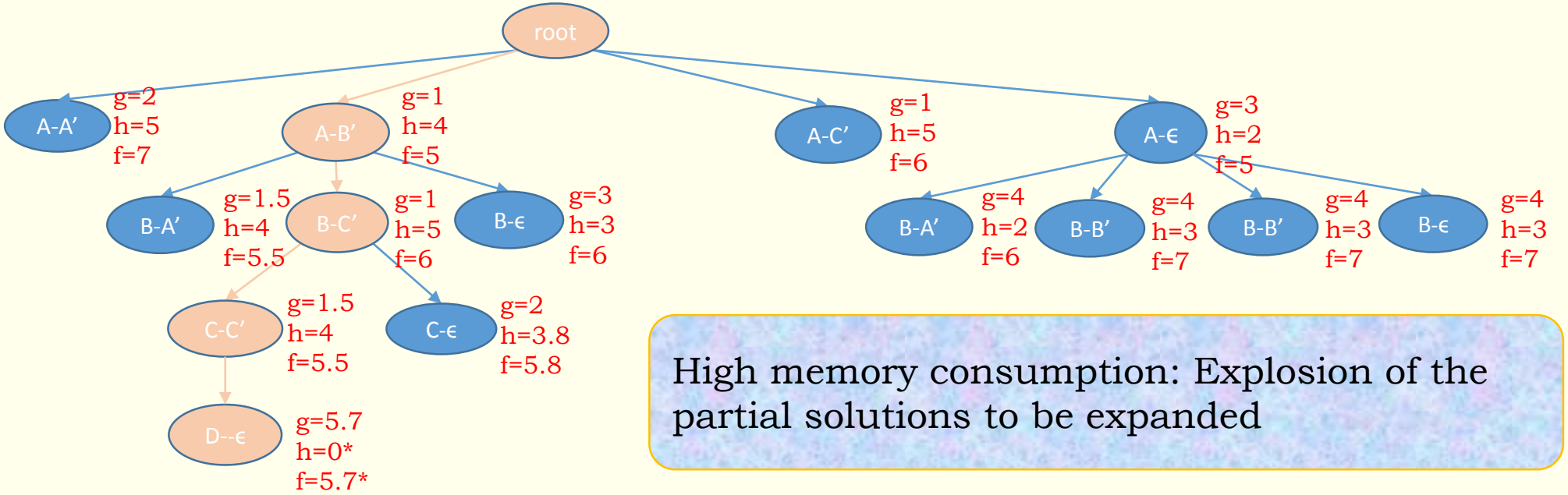
g: current cost
h: estimated cost



Exact Graph Edit Distance Approaches

- **Tree-Based Search Approaches**
  - ✓ A* algorithm: [Riesen 2007]

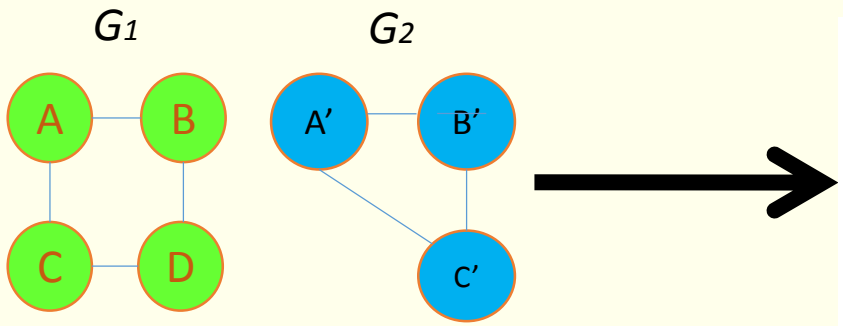




High memory consumption: Explosion of the partial solutions to be expanded

Exact Graph Edit Distance Approaches

- **Tree-Based Search Approaches**
  - ✓ Mathematical Formulation [Justice and Hero 2006]

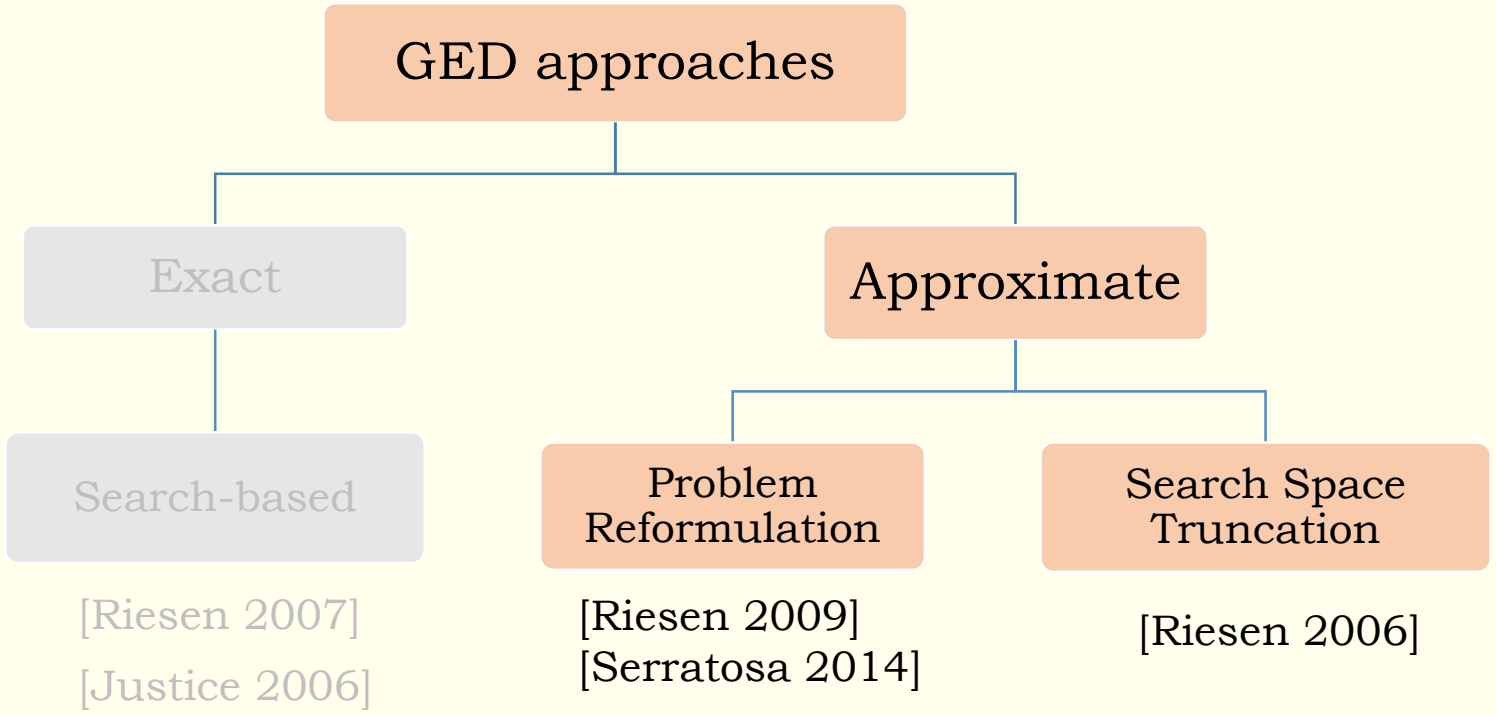


$$GED(G_1, G_2) = \min_{P,S,T \in \{0,1\}^{NXN}} \sum_{i=1}^{N} \sum_{j=1}^{N} c(l(A_1^i), l(A_2^j)) P^{ij} + \frac{1}{2} c(0,1)(S+T)^{ij}$$

$$s.t. \begin{cases} (A_1 P - P A_2 + S - T)^{ij} = 0, \ \forall i, j & \textbf{(1)} \\ \sum_i P^{ik} = \sum_j P^{kj} = 1, \forall k & \textbf{(2)} \end{cases}$$

It can only match unattributed edges

Graph Edit Distance Approaches



Approximate Graph Edit Distance Approaches

- **Search Space Truncation**
- **Problem Reformulation**

Approximate Graph Edit Distance Approaches

- **Search Space Truncation**
  - ✓ **Beam Search [Riesen 2006]**
- **Problem Reformulation**

- At each iteration, we keep the x most promising solutions
  - ✓ Suboptimal solution

Approximate Graph Edit Distance Approaches

- Search Space Truncation
- **Problem Reformulation**
  - ✓ **Bipartite Graph Matching: [Riesen 2009]**

Graph Matching Problem (Quadratic Assignment Problem) → Reformulation → **Linear Sum Assignment Problem**

$$C_{ve} = \left[\begin{array}{cccc|cccc} c_{1,1} & \dots & \dots & c_{1,|V_2|} & c_{1,\epsilon} & \infty & \dots & \infty \\ \dots & \dots & \dots & \dots & \infty & c_{2,\epsilon} & \dots & \infty \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{|V_1|,1} & \dots & \dots & c_{|V_1|,|V_2|} & \infty & \dots & \infty & c_{|V_1|,\epsilon} \\ \hline c_{\epsilon,1} & \infty & \dots & \infty & 0 & \dots & \dots & 0 \\ \infty & c_{\epsilon,2} & \dots & \infty & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \infty & \dots & \infty & c_{\epsilon,|V_2|} & 0 & \dots & \dots & 0 \end{array}\right]$$

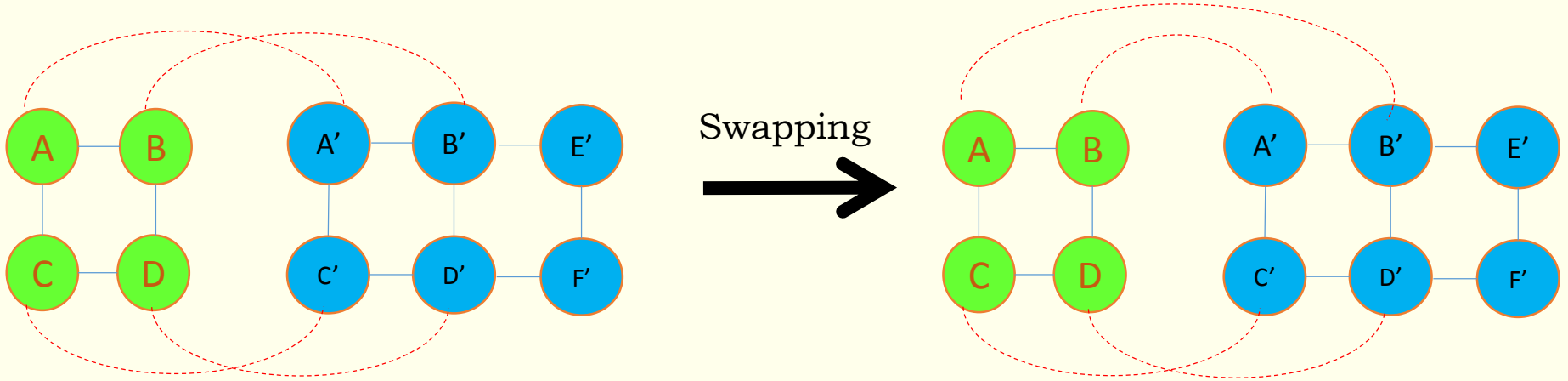
Approximate Graph Edit Distance Approaches

- Search Space Truncation
- **Problem Reformulation**
  - ✓ **Fast Bipartite Graph Matching; [Serratosa 2014]**

$$C_{ve} = \left|\begin{array}{cccc||cccc} c_{1,1} & \dots & \dots & c_{1,|V_2|} & c_{1,\epsilon} & \infty & \dots & \infty \\ \dots & \dots & \dots & \dots & \infty & c_{2,\epsilon} & \dots & \infty \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{|V_1|,1} & \dots & \dots & c_{|V_1|,|V_2|} & \infty & \dots & \infty & c_{|V_1|,\epsilon} \\ \hline\hline c_{\epsilon,1} & \infty & \dots & \infty & 0 & \dots & \dots & 0 \\ \infty & c_{\epsilon,2} & \dots & \infty & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \infty & \dots & \infty & c_{\epsilon,|V_2|} & 0 & \dots & \dots & 0 \end{array}\right| \rightarrow C_{ve} = \left|\begin{array}{ccc} c_{1,1} - (c_{1,\epsilon} + c_{\epsilon,1}) & \dots & c_{1,|V_2|} - (c_{1,\epsilon} + c_{\epsilon,|V_2|}) \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ c_{|V_1|,1} - (c_{|V_1|,\epsilon} + c_{\epsilon,1}) & \dots & c_{|V_1|,|V_2|} - (c_{|V_1|,\epsilon} + c_{\epsilon,|V_2|}) \end{array}\right|$$

Approximate Graph Edit Distance Approaches

- Search Space Truncation
- **Problem Reformulation**
  - ✓ **Bipartite Graph Matching: [Riesen 2009]**
    - ➢ Local rather than global structures
  - ✓ **Improvements by Local Search**



Approximate Graph Edit Distance Approaches

- Search Space Truncation
- **Problem Reformulation**
  - ✓ **Bipartite Graph Matching: [Riesen 2009]**
    - ➢ Local rather than global structures
  - ❖ **Improvements via Local Search Strategies**
    - ➢ Genetic algorithm: [Riesen 2014]
    - ➢ Beam Search algorithms:
      - ❑ BP-Beam: [Riesen 2014] in Artifical Neural Network in PR
      - ❑ IBP-Beam: [Ferrer 2015] in GBR2015
      - ❑ SBP-Beam: [Ferrer 2015] in PRL

Limitations of Graph Edit Distance Approaches

- **No intense study on the exact graph edit distance side.**
  - ✓ Long run time and high memory consumption
  - ✓ Good to evaluate accuracy
- **Approximate methods are fast enough, BUT:**
  - ✓ No deep study on the quality of the solution as a function of time
- **No algorithms work under time and memory constraints**
  - ✓ Tree-based algorithms can be of great interest "anytime algorithms"

Contribution: Anytime GED that can find a good compromise between quality and speed

Graph Matching Repositories and their Metrics

	Problem Type	Graph Type	Database Type	Metrics Type	Purpose
[Santo2003]	Exact GM	Non-attributed	Synthetic	Accuracy and scalability	Matching
CMU	Error-tolerant GM	Attributed	Real-world	Memory consumption, accuracy and matching	Matching
IAM DB [Riesen 2008]	Error-tolerant GM	Attributed	Real-world	Accuracy and running time	Classification
[Foggia 2001]	Exact GM	Attributed	Synthetic	Accuracy and scalability	Matching
[Carletti 2014]	Exact GM	(Non) attributed	Real-world	Scalability	Matching
[Pascal 2007]	Error-Tolerant GM	Attributed	Real-world	Accuracy and matching	Classification/ Matching
Tarragona in [Cortes 2015]	Error-Tolerant GM	Attributed	Real-World	Accuracy and matching	Matching

Limitations of Existing Repositories

- **Lack of metrics for GM methods:**
  - ✓ None of them studied the scalability of methods
  - ✓ Lack of diversity of graph datasets (dense, large graphs, etc.)

Contribution: An annotated repository dedicated to scalability with performance evaluation metrics

Outline

- Introduction
- Strength and Weakness of Graph Matching Methods
- **Contribution 1: Optimization Techniques**
  - **Toward an Anytime Graph Matching Approach**
  - **Parallel and Distributed Approaches for Graph Edit Distance**
- Contribution 2: Metrics and Datasets for Performance Evaluation
- Experiments
- Conclusion

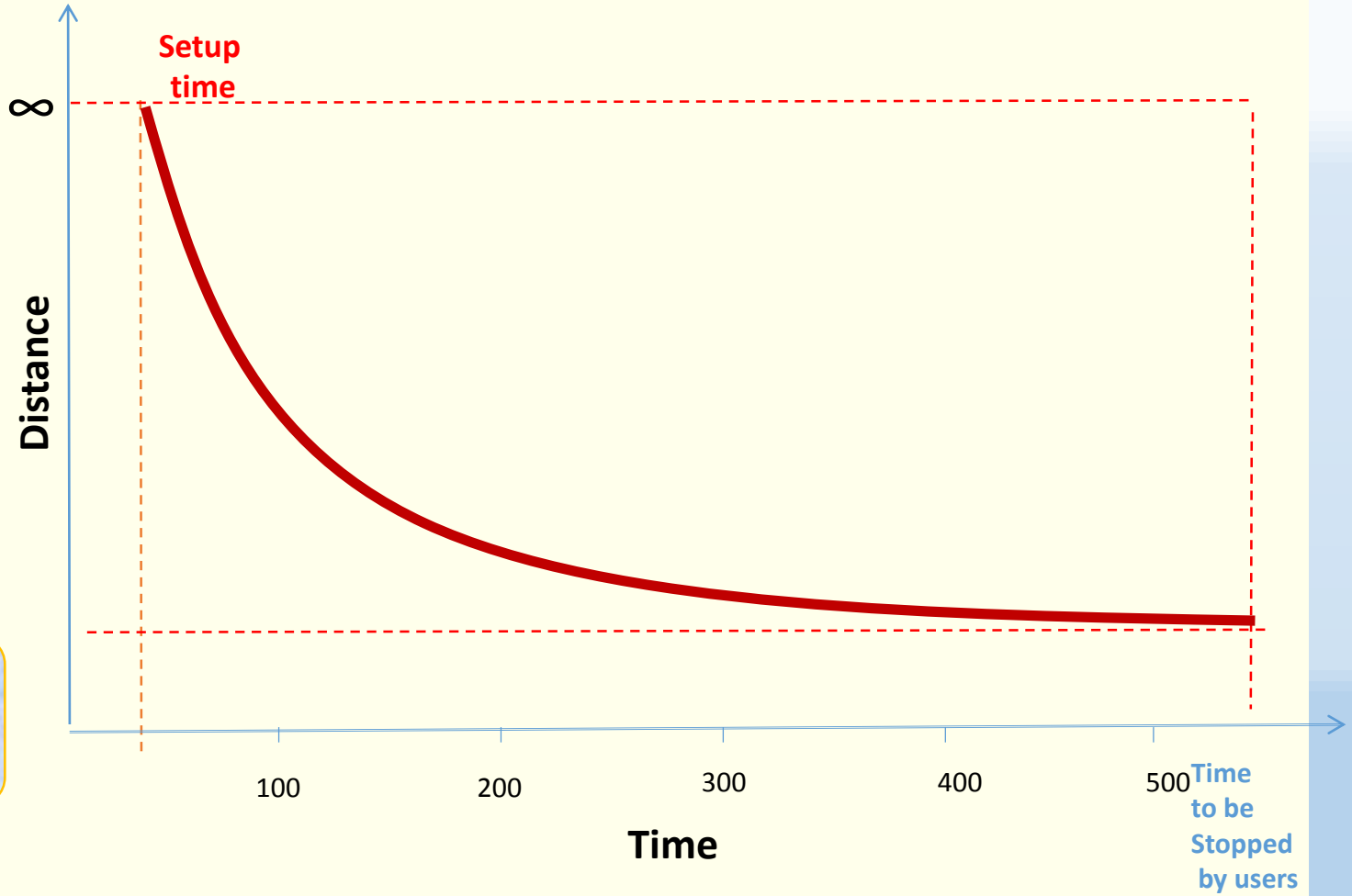
From Exact to Anytime Algorithms

- **Anytime Algorithms**

  [Zilberstein 1996]

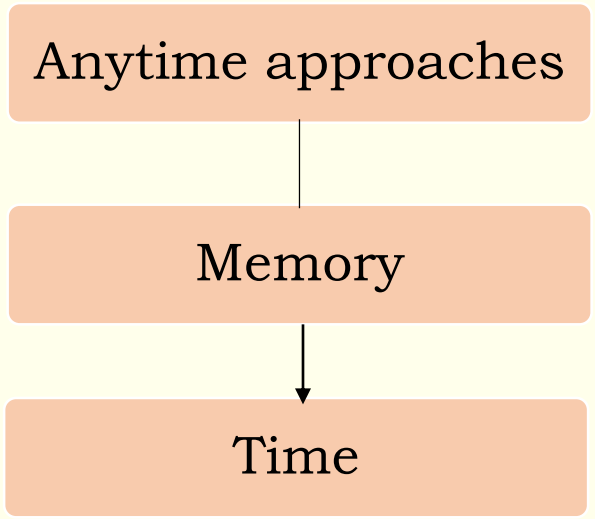
  - ✓ **Monotonicity**
  - ✓ **Diminishing returns**
  - ✓ **Interruptability**

- Setup time and quality of the first solution



From Exact to Anytime Algorithms

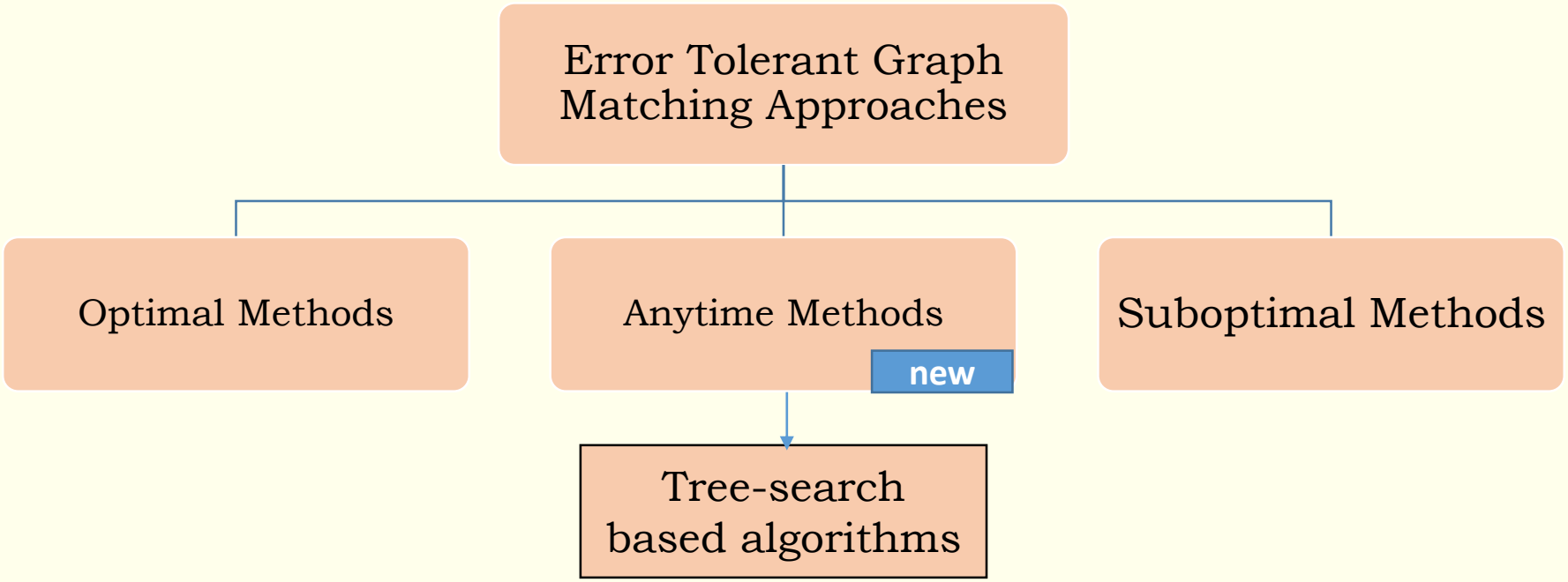
- **Anytime Algorithms:**
  - ✓ Search-tree based algorithms are anytime algorithms by nature



Recursive Best-First Search [Korf 1993]
Learning Real-Time [Shimbo 2003]

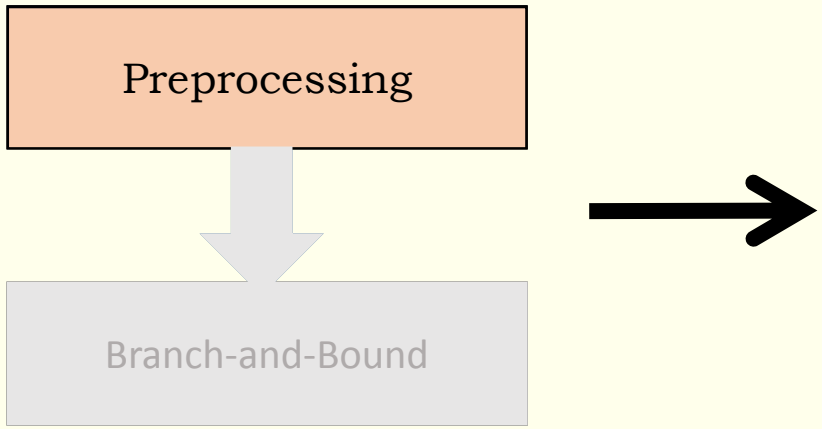
A* [Zhou 2002]
Bandwidth heuristic search [Hansen 2007]
Weighted A* [Likhachev 2008]

Anytime Graph Matching



From Exact Graph Edit Distance to Anytime One

- **Overcoming the memory bottleneck of A***
  - ✓ Depth-First Graph Edit Distance (*DF*)



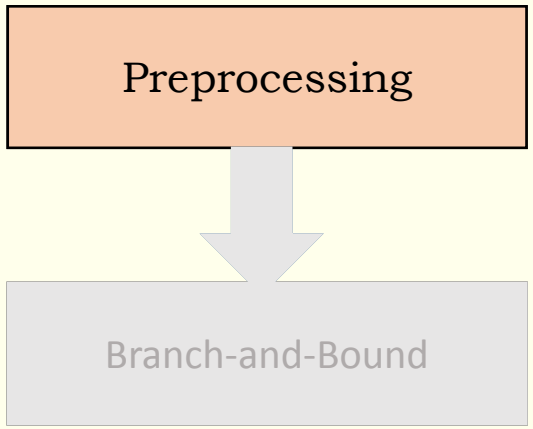
Cost Matrices

$$C_e \text{ and } C_V = \left[\begin{array}{ccccc|cc} c_{1,1} & \cdots & \cdots & c_{1,m} & & c_{1\leftarrow\epsilon} & c_{1\rightarrow\epsilon} \\ \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots \\ c_{n,1} & \cdots & \cdots & c_{n,m} & & c_{n\leftarrow\epsilon} & c_{n\rightarrow\epsilon} \\ \hline c_{\epsilon\rightarrow 1} & \cdots & \cdots & c_{\epsilon\rightarrow m} & & \infty & \infty \\ c_{\epsilon\leftarrow 1} & \cdots & \cdots & c_{\epsilon\leftarrow m} & & \infty & \infty \end{array}\right]$$

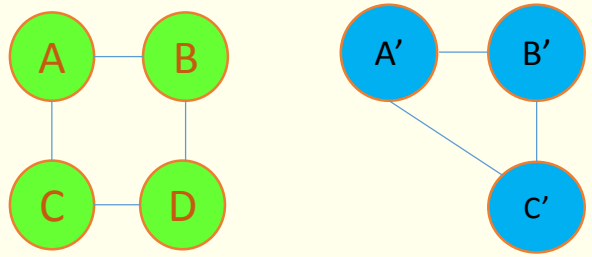
Z. Abu-Aisheh, R. Raveaux, J-Y Ramel and P. Martineau : An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems. ICPRAM (1) 2015 : 271-278.

From Exact Graph Edit Distance to Anytime One

- **Overcoming the memory bottleneck of A***
  - ✓ Depth-First Graph Edit Distance (*DF*)



❖ To prune the search tree



Bipartite Graph Matching

$$d_{UB} = d_{p_{min}}$$

$$p_{min} = \{\underbrace{u_1 \rightarrow v_{i_1}}_{\omega_1}, \underbrace{u_2 \rightarrow v_{i_2}}_{\omega_2}, \ldots, \underbrace{u_k \rightarrow v_{i_k}}_{\omega_k}\}$$

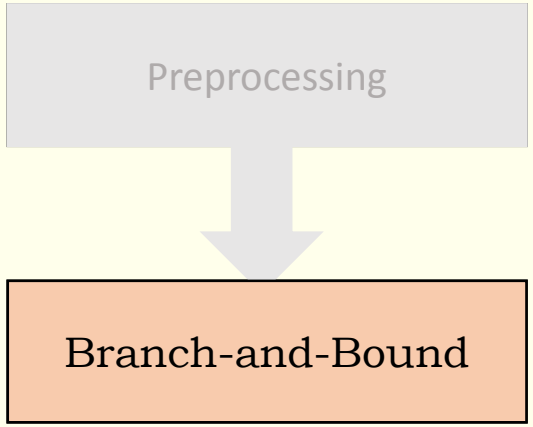
❖ To start with the promising vertices of G1

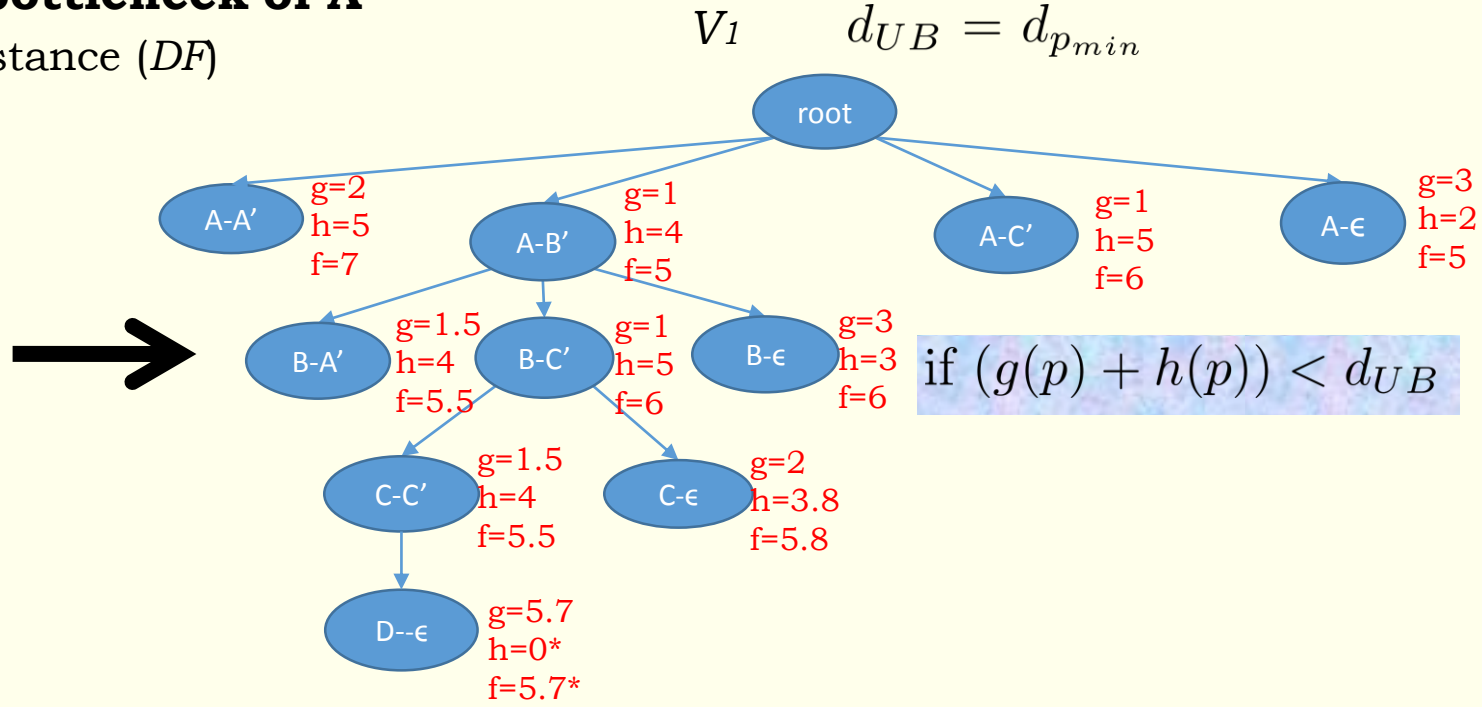
Sort V_1 in ascending order according to ω's

$$V_1 = \{u_k, u_2, u_1, \ldots.\}$$

From Exact Graph Edit Distance to Anytime One

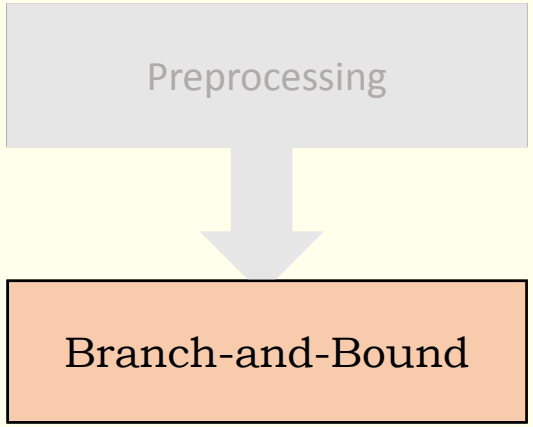
- **Overcoming the memory bottleneck of A***
  - ✓ Depth-First Graph Edit Distance (*DF*)

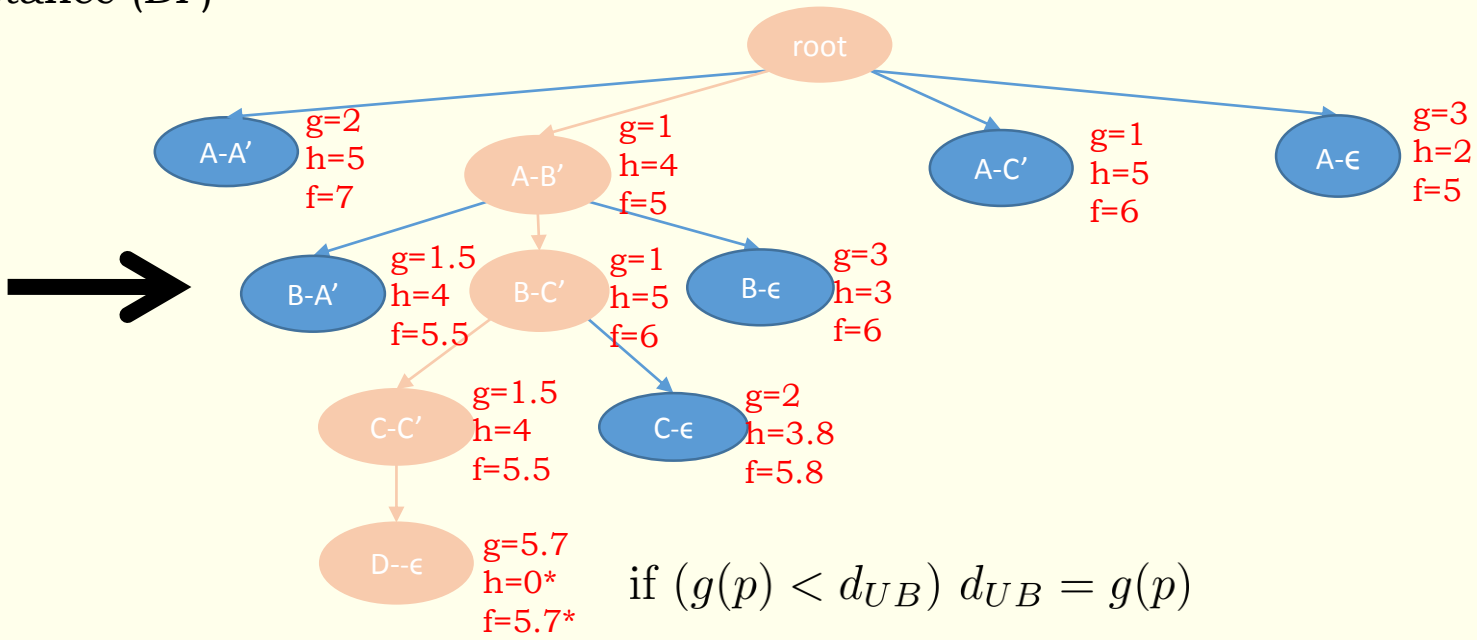




From Exact Graph Edit Distance to Anytime One

- **Overcoming the memory bottleneck of A***
  - ✓ Depth-First Graph Edit Distance (*DF*)

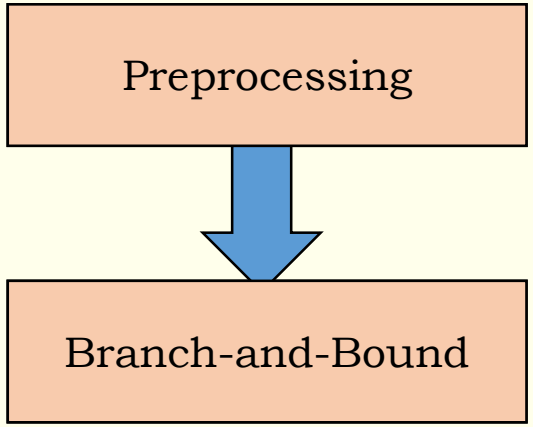


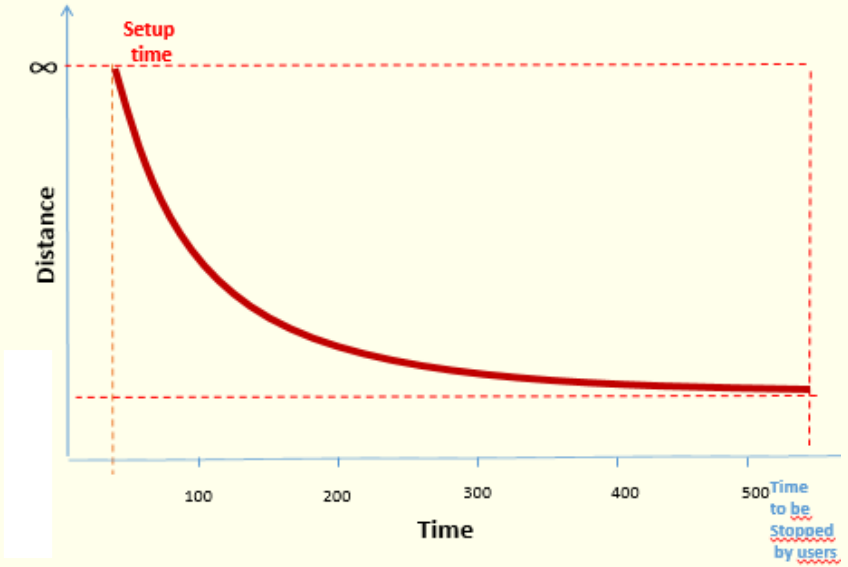


$$\text{if } (g(p) < d_{UB})\ d_{UB} = g(p)$$

From Exact Graph Edit Distance to Anytime One

- From *DF* to Anytime *DF* (***ADF***)



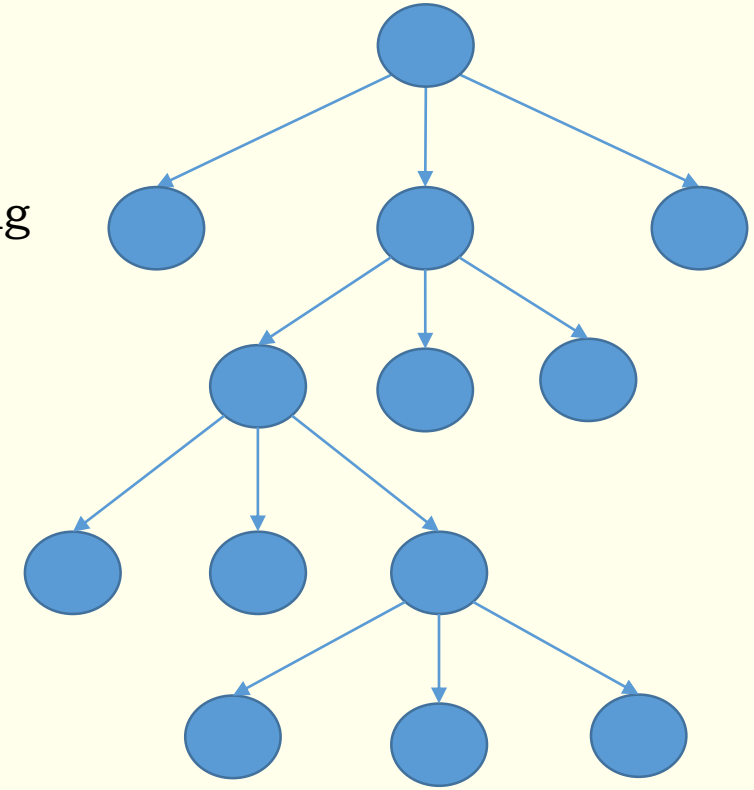


- Export all the list of d_{UB}
- **Interruptability:** At each iteration, we can stop the algorithm if it exceeds C_T

Z. Abu-Aisheh, R. Raveaux and J-Y Ramel: Anytime Graph Matching. Special issue of Pattern Recognition Letters (revision: 2nd round).

ADF: Theoretical Observations

- **ADF memory complexity is O(|V_1|.|V_2|)**
  - ✓ Encountering a local minimum while exploring
  - ✓ **To have more diversity + scaling up**
    - ➢ Parallel and Distributed versions of *ADF*



Raised Questions

1. What are the subtasks that should be generated before the parallelism starts?

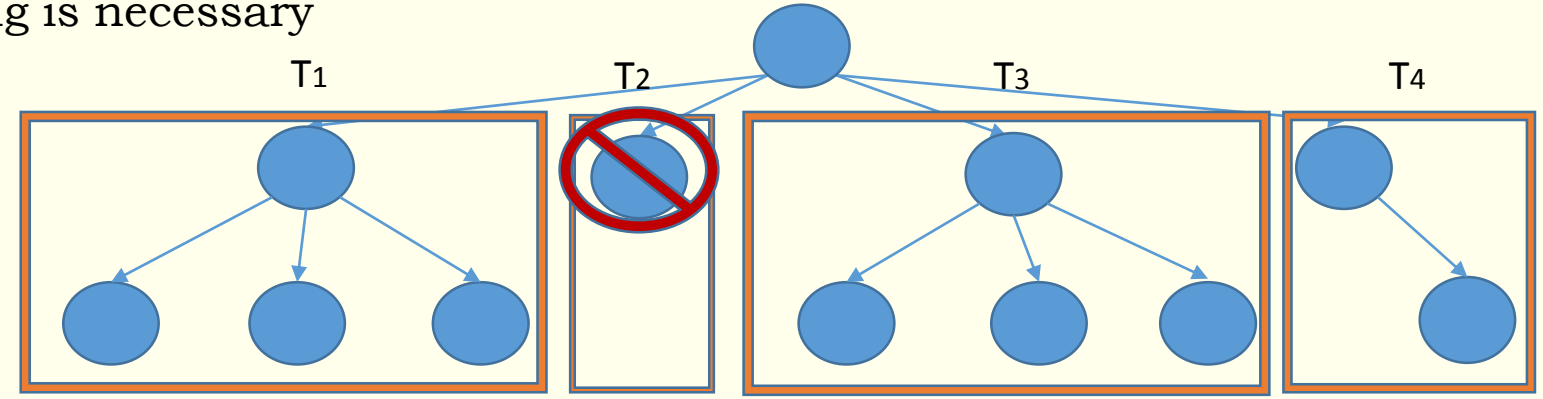
2. How to distribute the tasks?

3. How to keep all threads busy?

4. How to prune the search space as fast as possible?

Coming back to Anytime Depth-First (ADF)

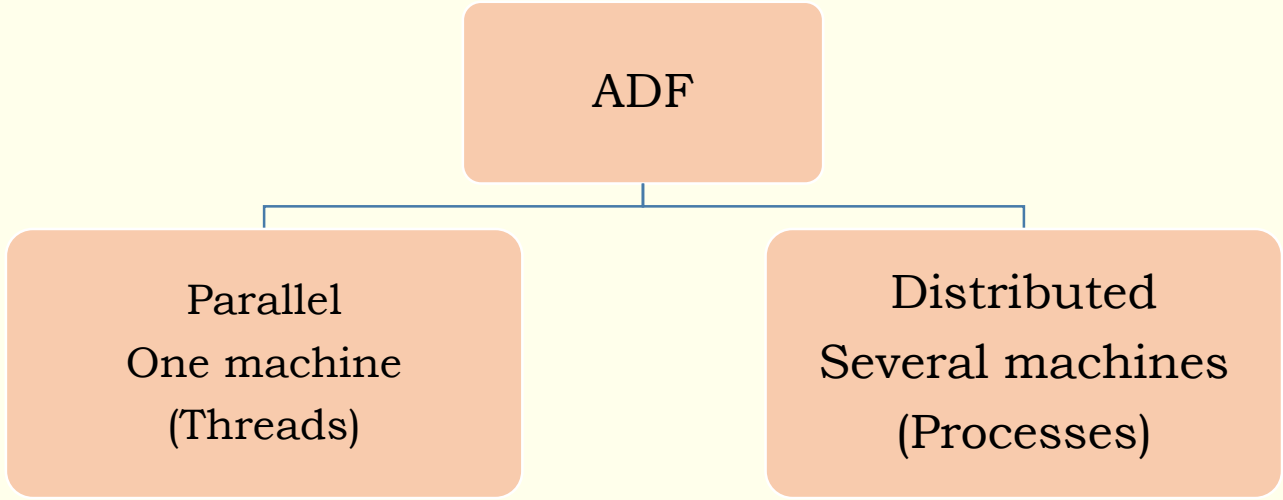
- Tasks = exploration of partial solutions is more time consuming than g(p) and h(p)
- Search tree is irregular
  - ✓ Load Balancing is necessary



- Prune the search tree: sharing the upper bound
- Number of CPU/machines
  - ✓ No rule (experiments)

Parallel and Distributed Branch-and-Bound Algorithms

- None of branch-and-bound algorithms was dedicated to solving error-tolerant graph matching

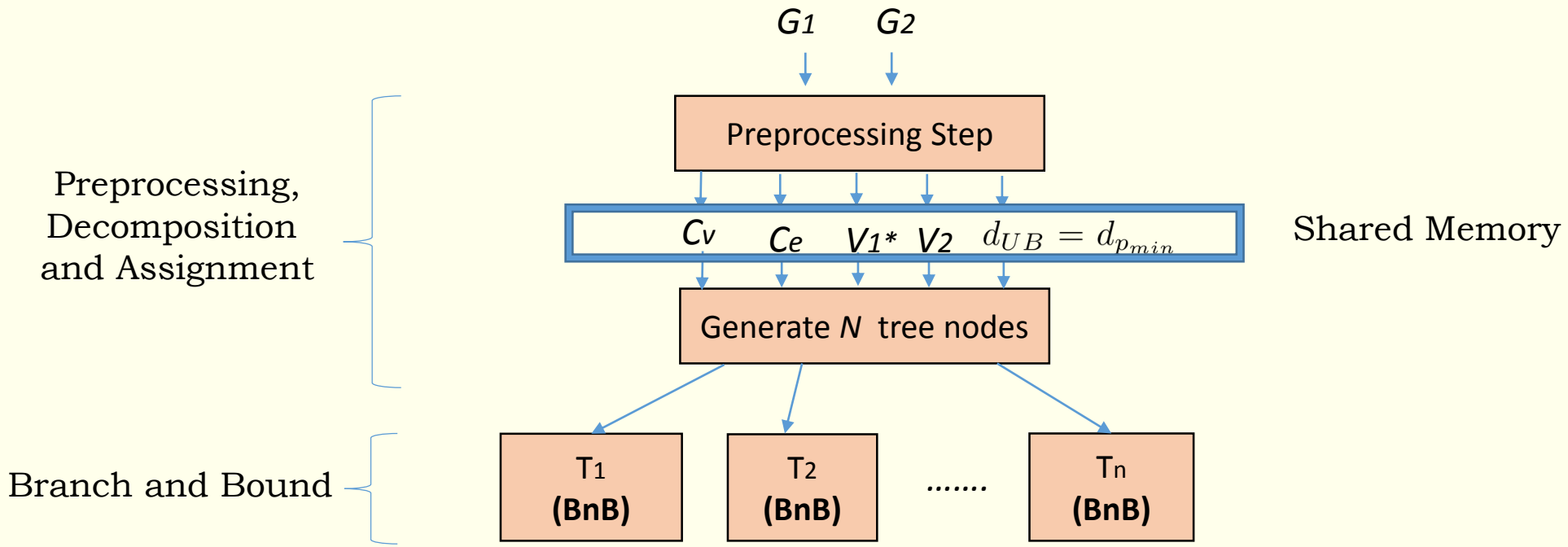


[Rao 1987] [Neary 2005]

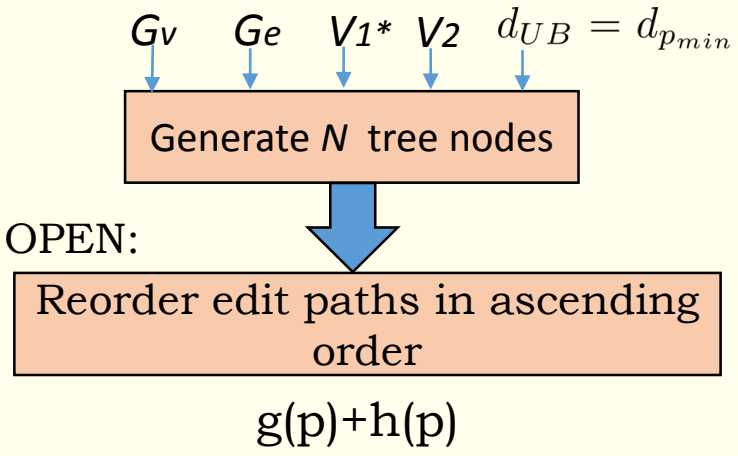
✓ Multi-threads

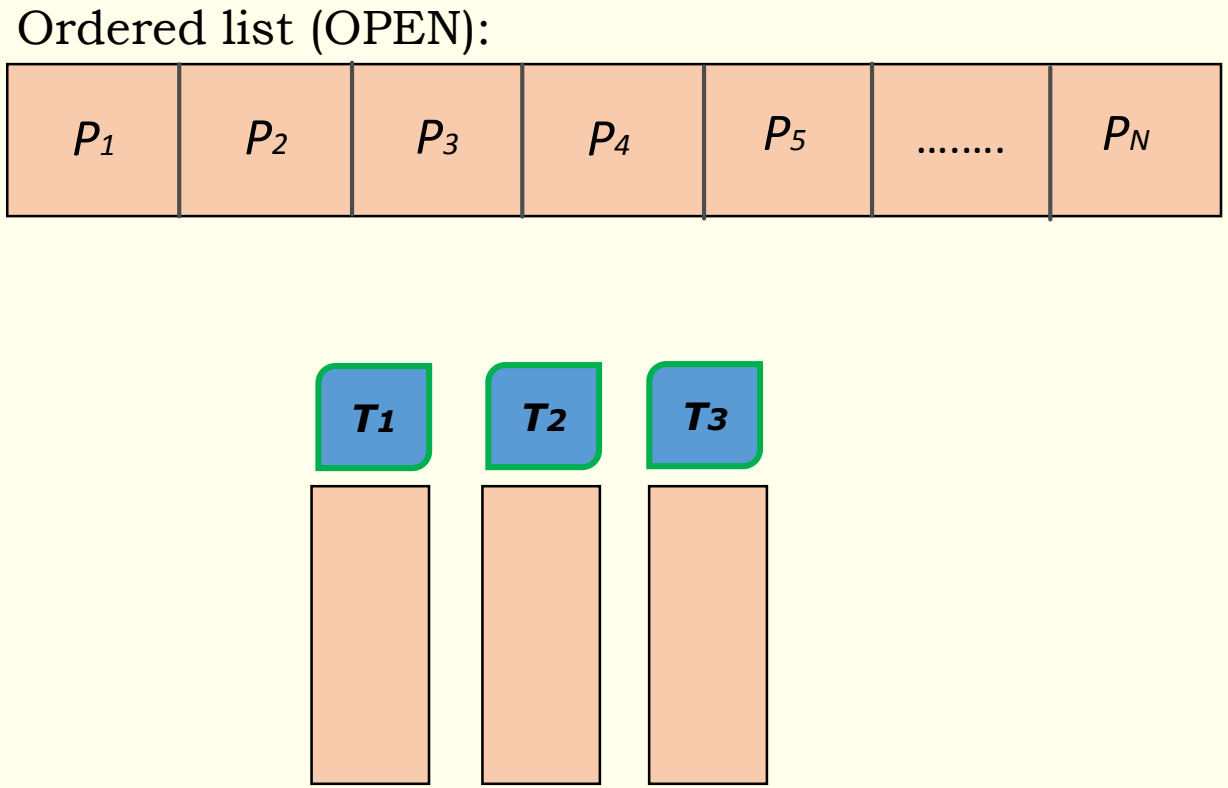
✓ Load Balancing

A Parallel Depth-First Graph Edit Distance (***PDFS***)



PDFS: Decomposition and Assignment

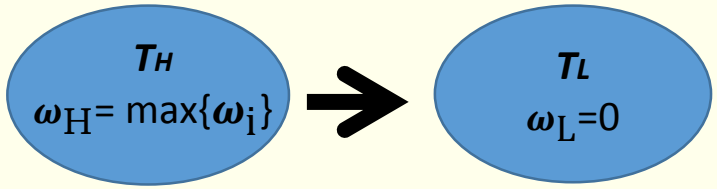




PDFS: Load Balancing

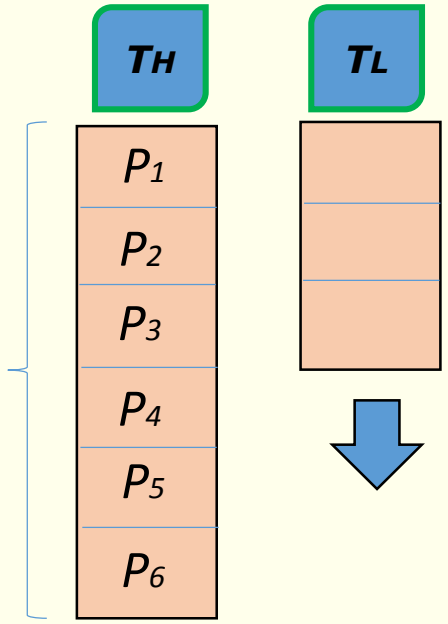
Load Balancing

Heavy thread → Light thread



$$w_i = \sum_{j=1}^{|OPEN_{T_i}|} |pendingV_1(p(j))| \quad \forall i \in T_i$$

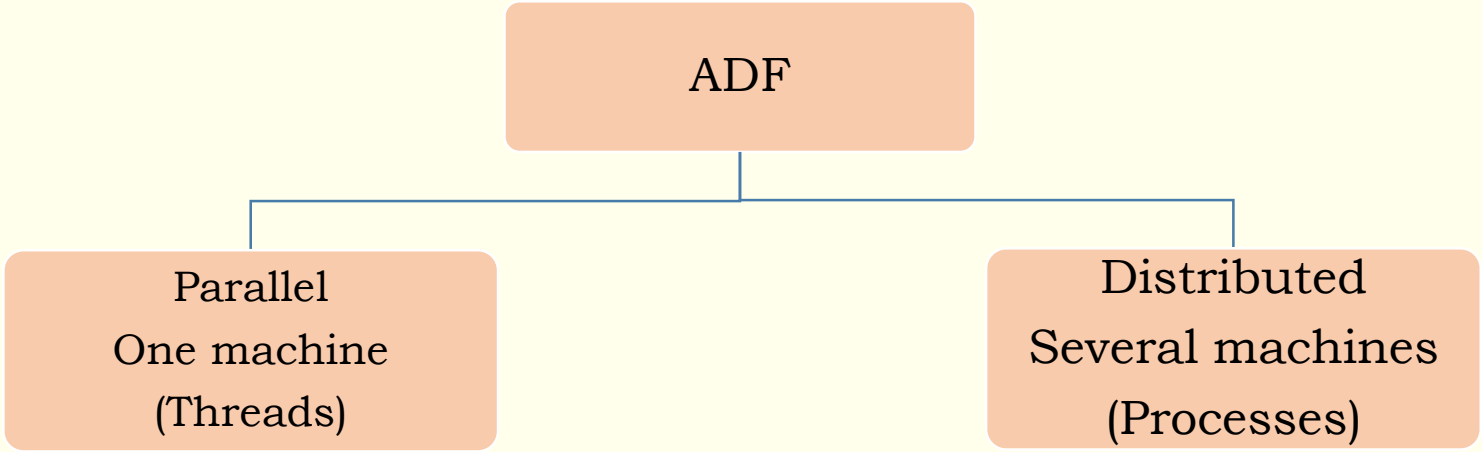
Reorder edit paths in ascending order of ***g+h***



Stop load balancing when $\omega_L > \frac{\omega_H}{2}$

Parallel and Distributed Branch-and-Bound Algorithms

- None of branch-and-bound algorithms was dedicated to solving error-tolerant graph matching.



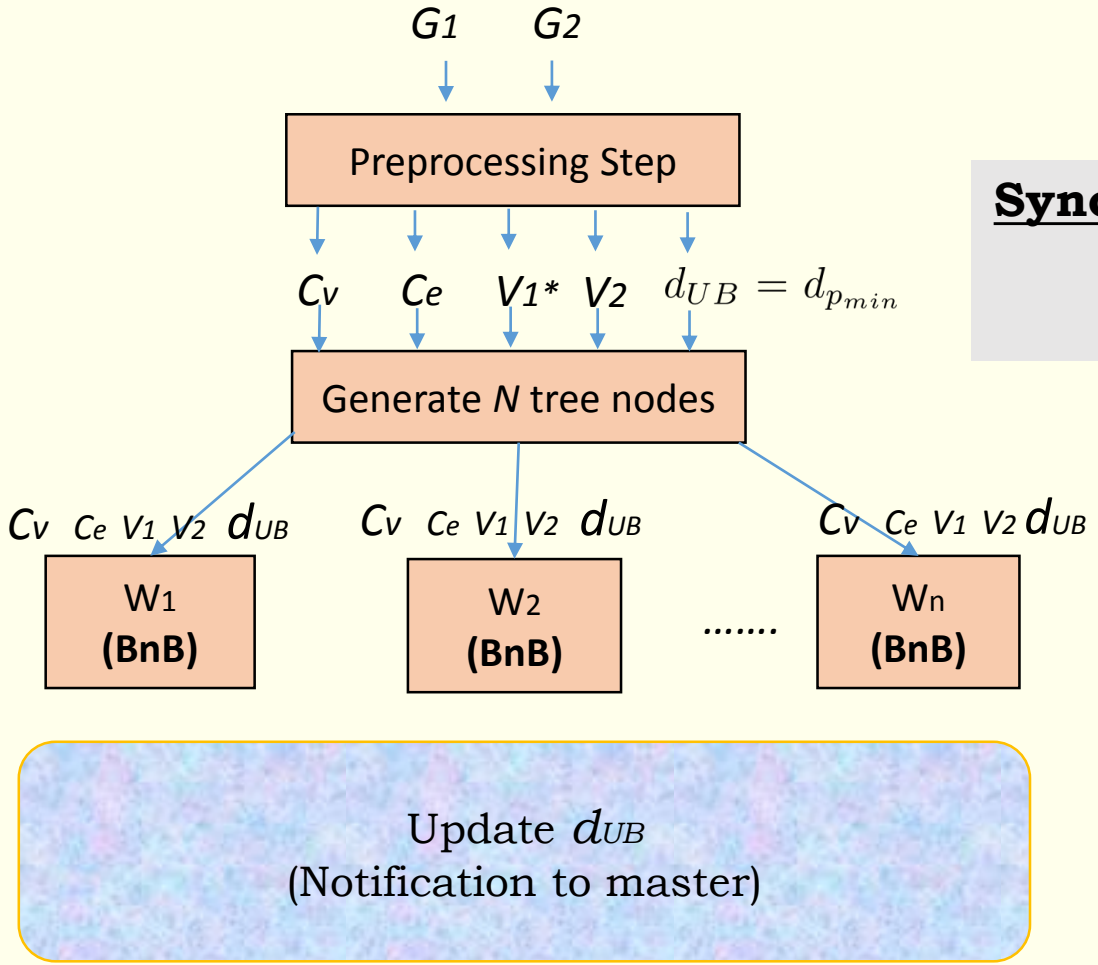
- Master-Slave architecture on top of Hadoop
- No load balancing
  - ✓ Fault Tolerance

Z. Abu-Aisheh, R. Raveaux, J-Y Ramel and P. Martineau: A Distributed Algorithm for Graph Edit Distance. GraphSM'2016.

A Distributed Graph Edit Distance Algorithm (***D-DF***)

Preprocessing, Decomposition and Assignment

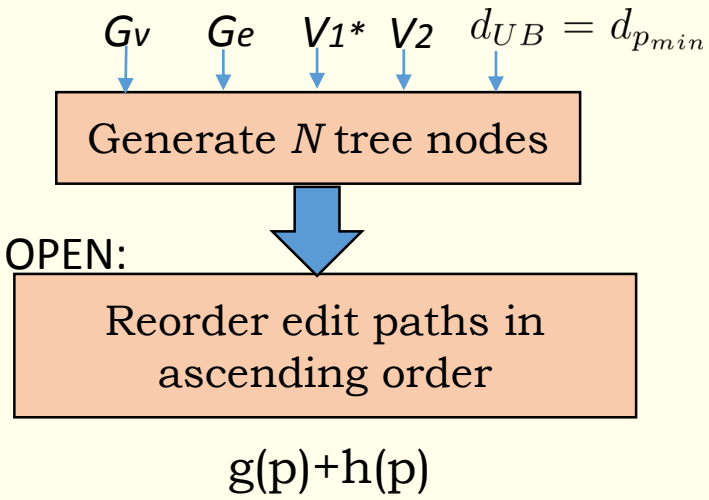
Branch and Bound



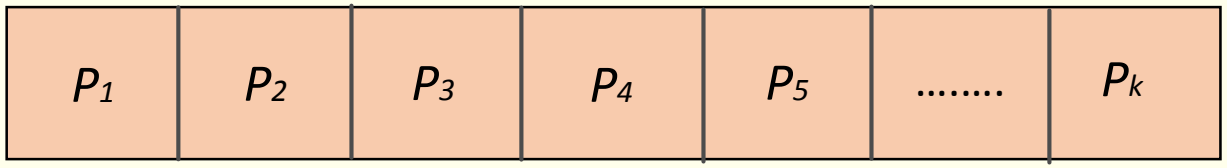
Synchronized variable

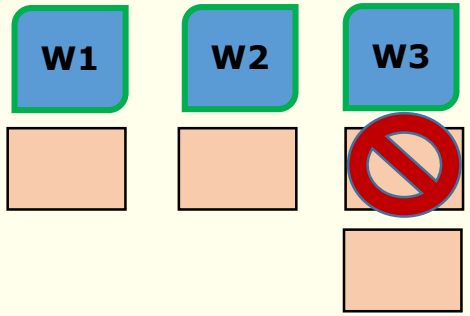
d_{UB}

D-DF: Decomposition and Assignment



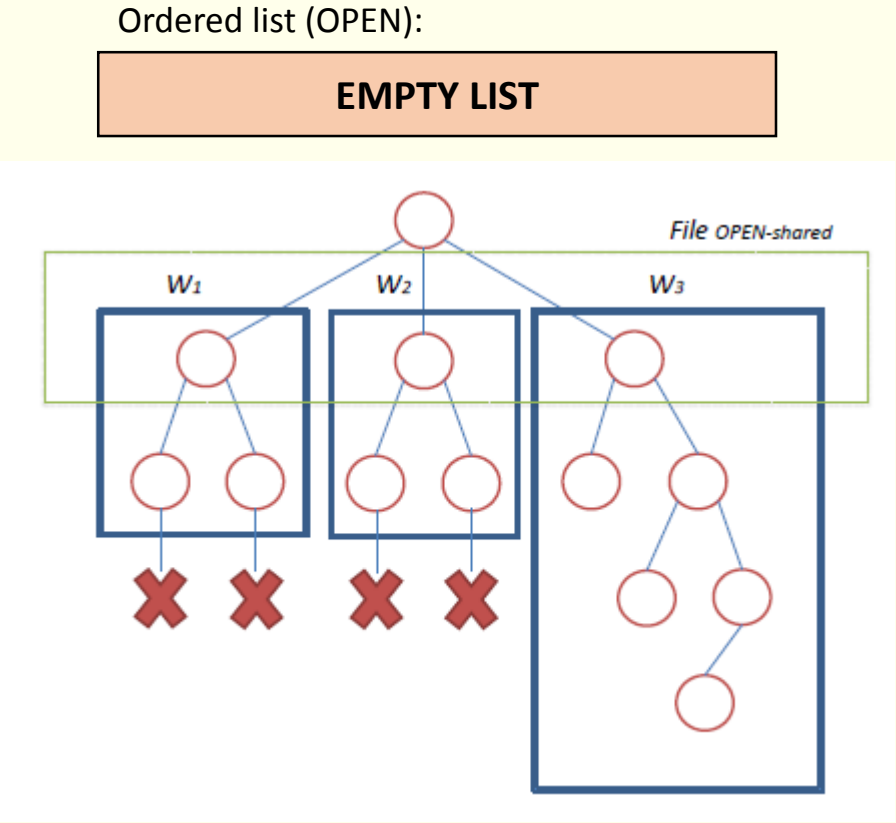
Ordered list (OPEN):





PDFS and ***D-DF*** Theoretical Observations

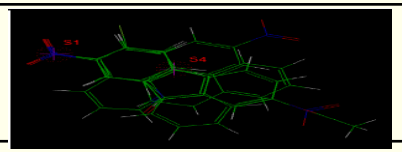
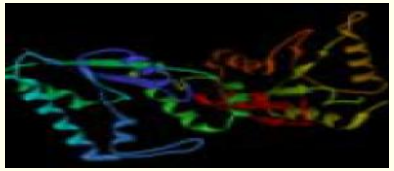
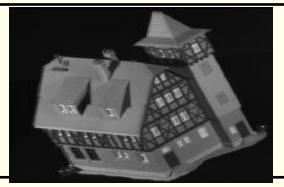
- ***PDFS*** and ***D-DF***:
  - ✓ **Inputs:**
    - ➢ Generate N nodes before dispatching the work
    - ➢ *h*: estimated cost
    - ➢ *PDFS*: Number of threads
    - ➢ *D-DF*: Number of processes
  - ✓ Exploring different parts of the search tree
- ***D-DF***:
  - ✓ Single job
  - ✓ Workers may be left without any task to do



Outline

- Introduction
- Strength and Weakness of Graph Matching Methods
- Optimization Techniques
- **Metrics and Datasets for Performance Evaluation**
  - **GDR4GED: Graph Repository**
  - **GDR4GED: Performance Evaluation Metrics**
- Experiments
- Conclusion

GDR4GED: Graph Repository - Databases

Database	Decomposition	Overview	Purpose
GREC	MIX, 5, 10, 15 and 20		Classification
MUTA	MIX ,10, 20, 30, … and 70		Classification
Protein	MIX, 20, 30 and 40		Classification
CMU	30		Matching

Repository link: https://iapr-tc15.greyc.fr/links.html#Benchmarking and data sets

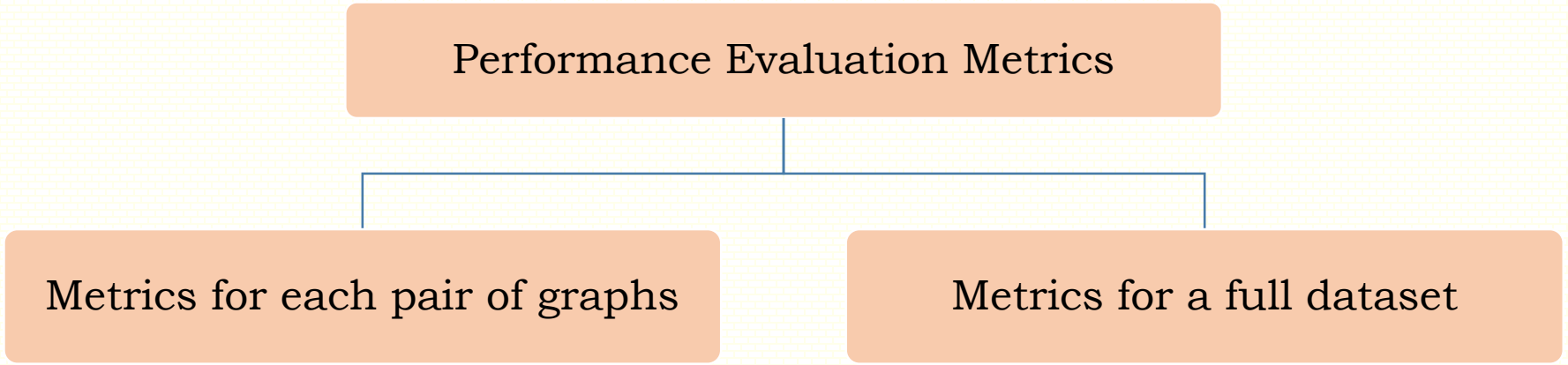
Z. Abu-Aisheh, R. Raveaux and J-Y Ramel: A Graph Database Repository and Performance Evaluation Metrics for Graph Edit Distance. GbRPR 2015 : 138-147.

GDR4GED: Graph Repository – Graph-Level Annotations

- **7 methods (Time constraint: 300 seconds per comparison):**
  - ✓ **Exact methods:**
    - ➢ *A** [Riesen 2007]
    - ➢ JHBLP [Justice 2006]
    - ➢ *DF* [Abu-Aisheh 2015]
    - ➢ *D-DF* [Abu-Aisheh 2016]
  - ✓ **Approximate methods:**
    - ➢ BeamSearch [Neuhaus 2006]
    - ➢ BP [Riesen 2009]
    - ➢ Hausdorff [Fischer 2013]
- **For each graph matching pair:**
  - ✓ The name of the method that found the best solution
    - ➢ Distance between each pair of graphs (Optimal/suboptimal)
    - ➢ Vertex-Vertex matching

GDR4GED: Performance Evaluation Metrics

- Under time and memory constraints

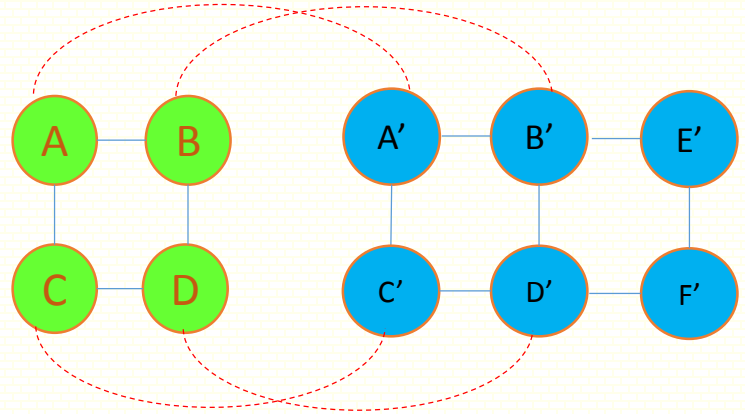


Metrics for each Pair of Graphs

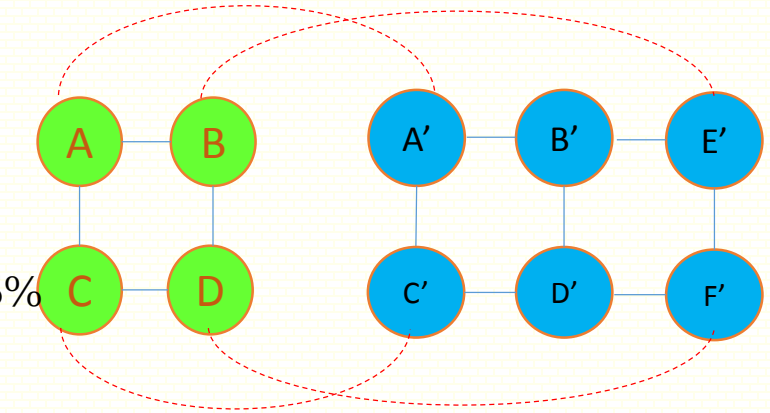
- Given the reference R:
  - ✓ Deviation
  - ✓ Matching Dissimilarity

- ✓ Best found solution

- Reference Distance (RD) = 5
- Reference Matching (RM):



- Distance (m) = 6
  - ✓ Deviation = 20%
- Matching Dissimilarity = 75%
- Best found solution = ?



Metrics for a Full Dataset

- Number of unfeasible solutions:
  - ✓ i.e. incomplete editpath
- Number of optimal solutions:
- Number of Time-Out and Out-Of-Memory cases
- Mean number of explored nodes
- Mean running time in milliseconds
- **Running time-Deviation plot:**
  - ✓ Projection on a two-dimensional space ($\mathbb{R}^2$)

Outline

- Introduction
- Strength and Weakness of Graph Matching Methods
- Optimization Techniques
- Metrics and Datasets for Performance Evaluation
- **Experiments**
  - **Included Methods**
  - **Tests under Soft and Hard Time Constraints**
  - **Quality as a Function of Time**
  - **Density Tests**
  - **Classification under Time and Memory Constraints**
- Conclusion

Included Methods

	Methods	Reference	Parameters
Exact	***A****	[Riesen 2007]	-
	JHBLP	[Justice 2006]	-
Approximate	***BP***	[Riesen 2009]	-
	FBP	[Serratosa 2014]	-
	H	[Fischer 2013]	-
	BS-x	[Riesen 2006]	x = 1, 10 and 100

Our Exact Approaches and their Parameters

- **Sequential Methods, *ADF* and *PDFS*:**
  - ✓ 24-core Intel i5 processor 2.10GHz
    16GB memory

- ***D-DF*:**
  - ✓ 5-node cluster of machines running Hadoop MapReduce version 1.0.4
    Each node contains a 4-core Intel i7 processor 3.07GHz, 8GB memory
  - ✓ Zookeeper: Message passing tool

Methods	Parameters
ADF	*h(p)*= heursitic*(BP)*
PDFS	*h(p)*= heursitic*(BP)* # threads = 64 *N* = 100
D-DF	*Lb(p)*= heursitic*(BP)* # machines = 5 # workers per machine = 4 *N* = 250

Objectives of the Experiments

1. **Tests under soft and hard time constraints.**
   - ✓ Soft Constraint: 300 seconds per comparison
   - ✓ Hard Constraint:

$$C_T = \max_{m,i,j}\{time_m(G_i, G_j)\}$$

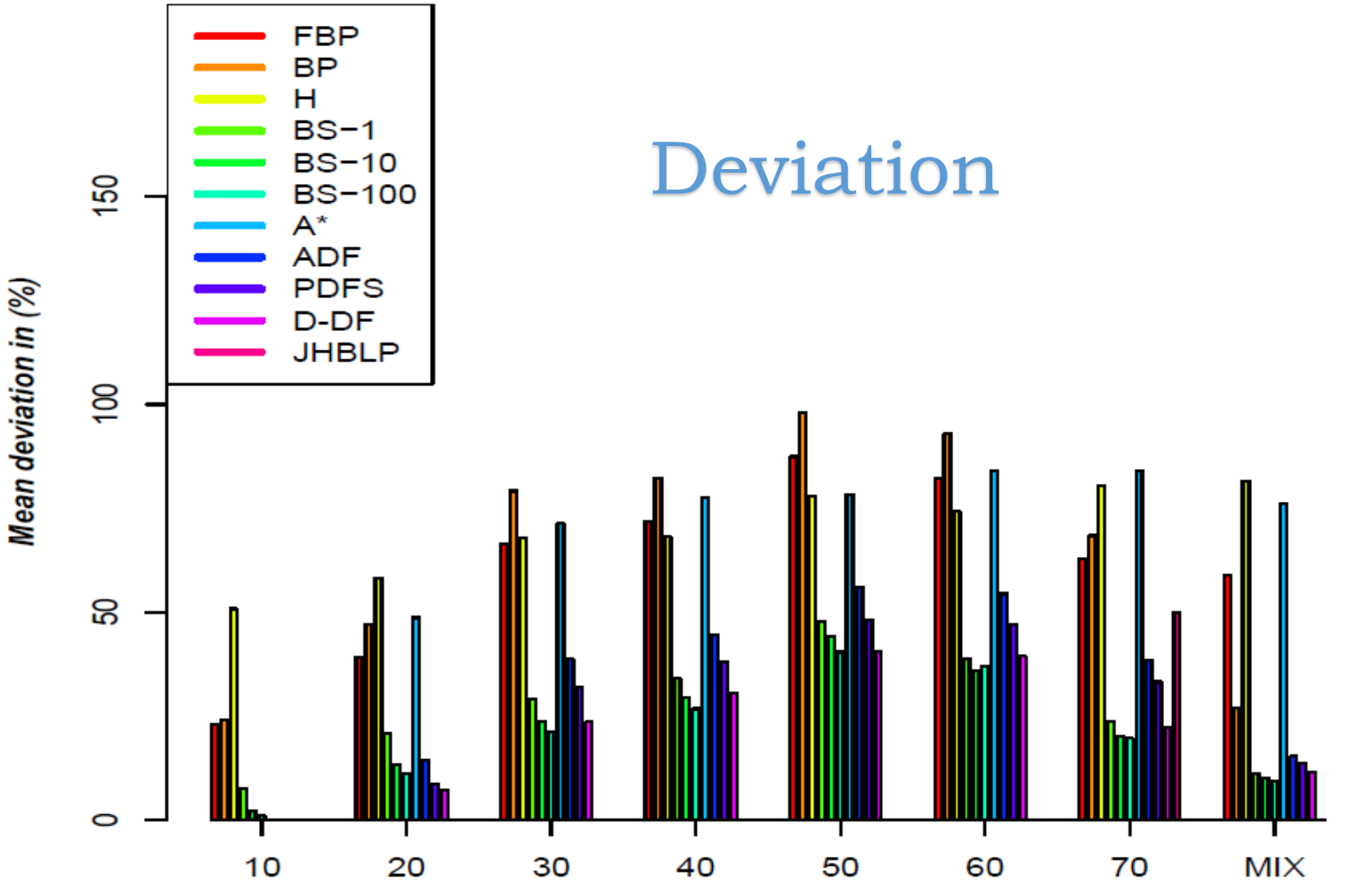
m is *BS*, *BP* or *FBP*

2. Trade-off between quality and time

3. Classification under time constraints

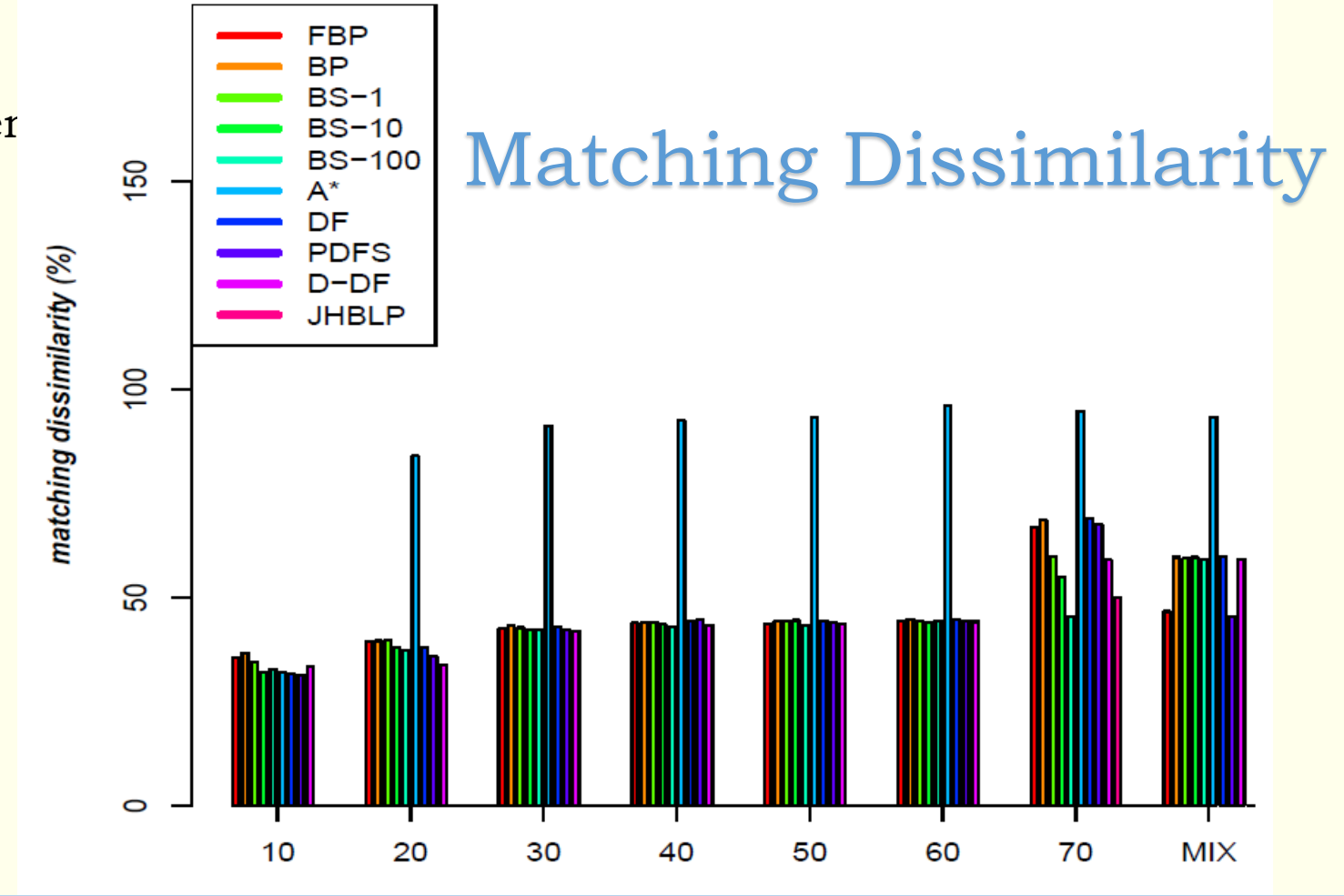
Soft Time Constraints: MUTA – 300 seconds

Methods	Avg Deviation (%)
FBP	*61*
BP	64
H	69.91
BS-1	26.6409
BS-10	**22.25503**
BS-100	**20.77122**
A*	65.04
ADF	32.37
PDFS	27.5
D-DF	**21.5**
JHBLP	**6.25**



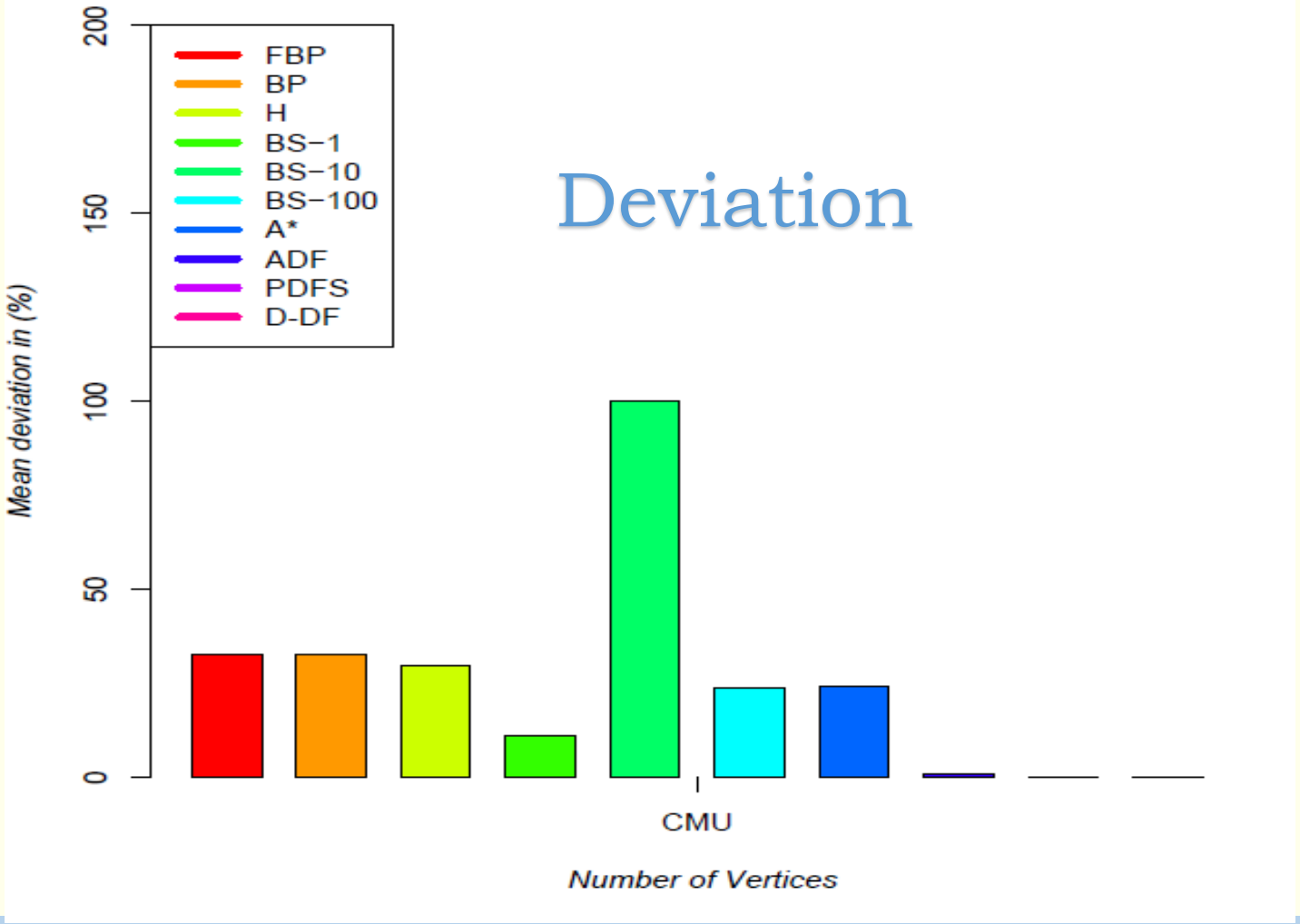
Soft Time Constraints: MUTA – 300 seconds

- Deviation is small
- Matching error is bigger



Soft Time Constraints: CMU – 300 seconds

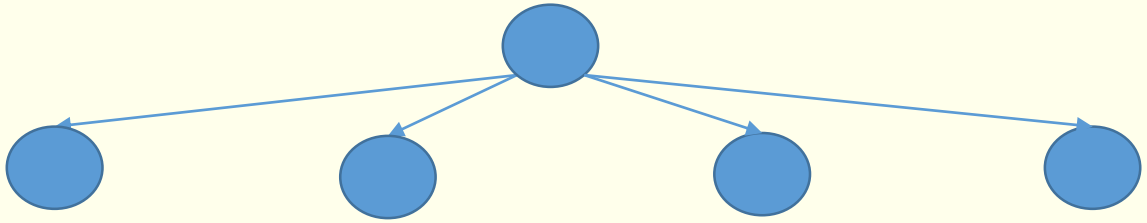
- **Out of memory**
  - ✓ *BS-100* and *A**
- *BS-10* is the worst
- *BS-1* is the best
  - ✓ 10.6%
- *D-DF* (0.78%)



Hard Time Constraints

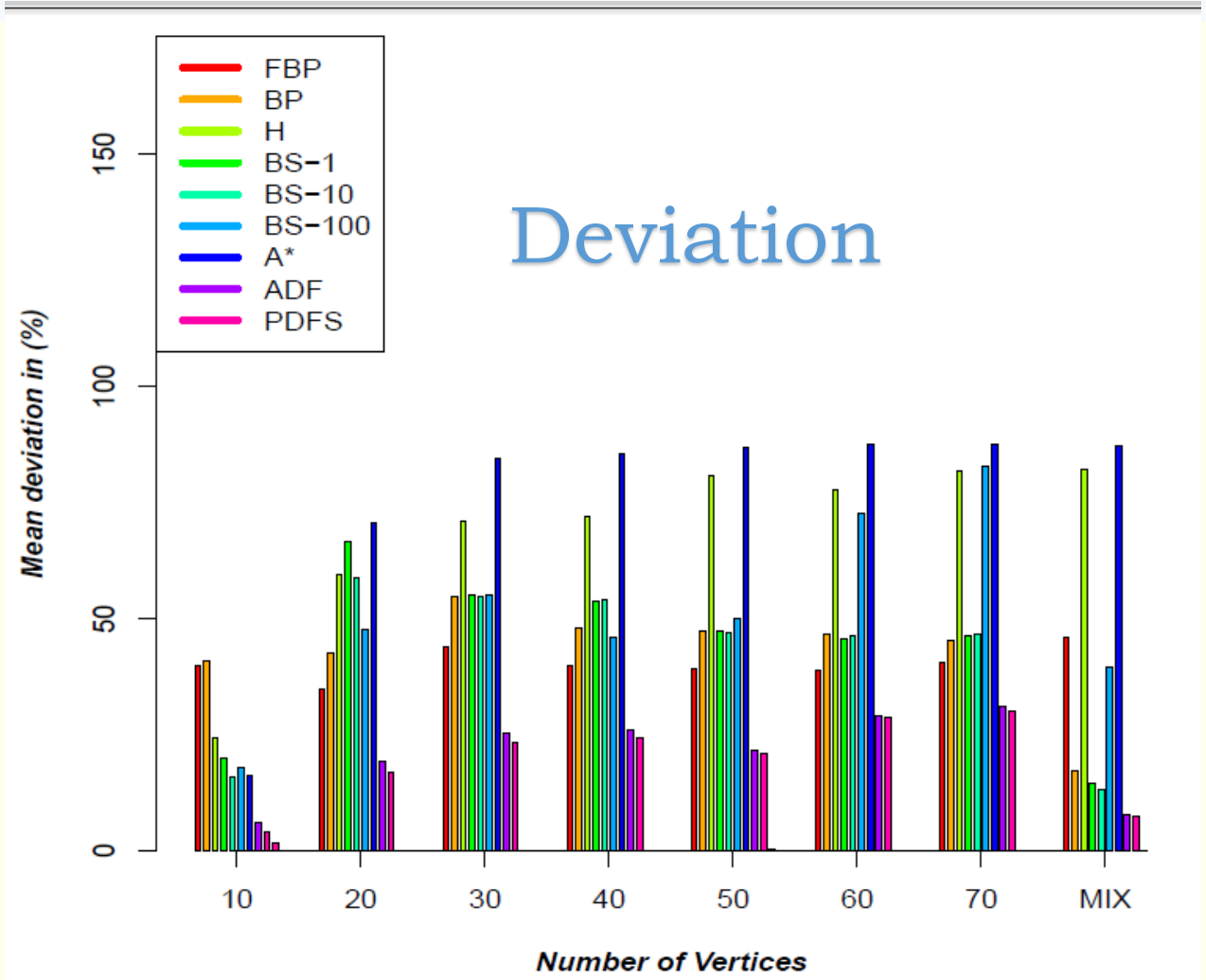
- *D-DF* cannot be tested under hard time constraints

- Since C_T is quite small:
  - ✓ The number of threads in *PDFS* is set to 3
  - ✓ Removing the lower bound from all of *BS*, *A**, *ADF* and *PDFS*
  - ✓ *N* is limited to the first level of the search tree



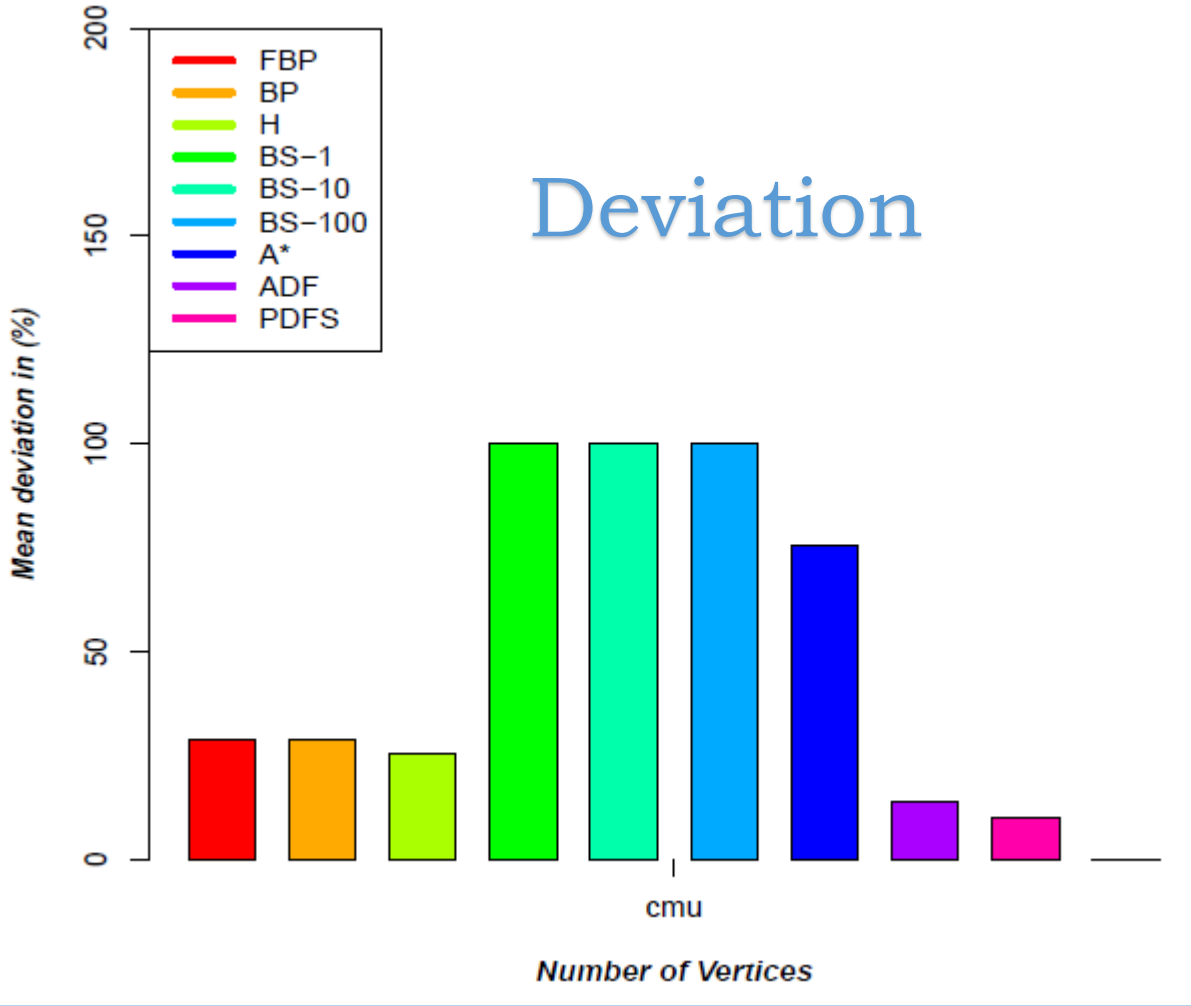
Hard Time Constraints: MUTA – 500 milliseconds

- ***JHBLP***
  - ✓ unfeasible solutions for Graphs > 30 vertices
- ***PDFS*:** 18%
- ***BS-100*** outputted unfeasible solutions on MUTA-50, 60, 70 and MIX
- ***BP***: 40.37%



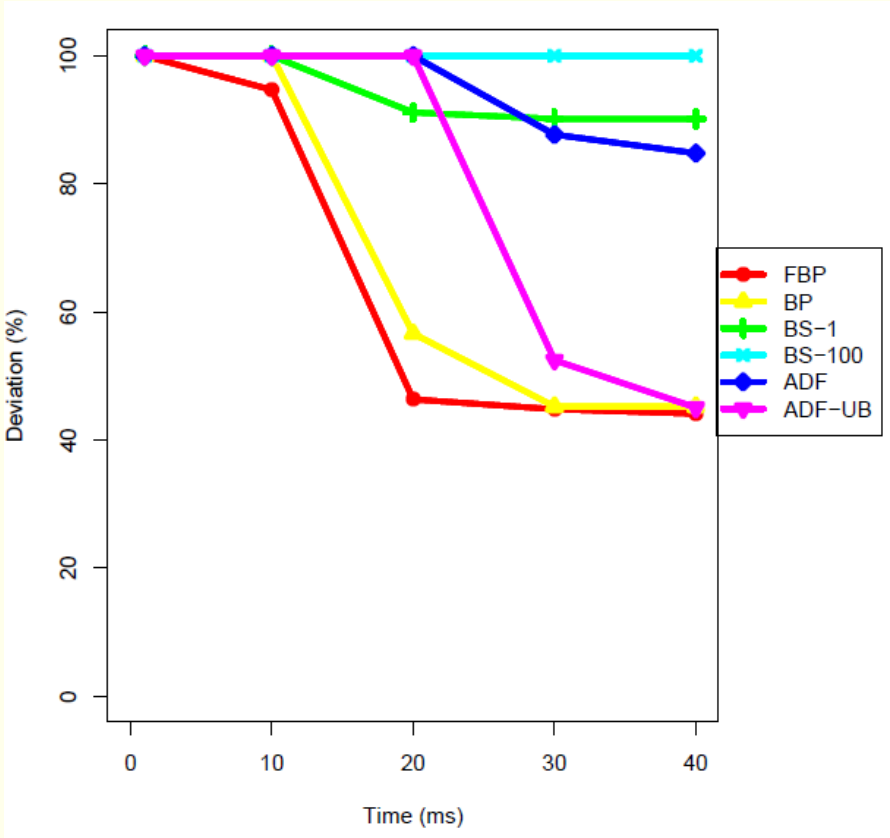
Hard Time Constraints: CMU – 500 milliseconds

- *BS-10* and *BS-100* outputted unfeasible solutions

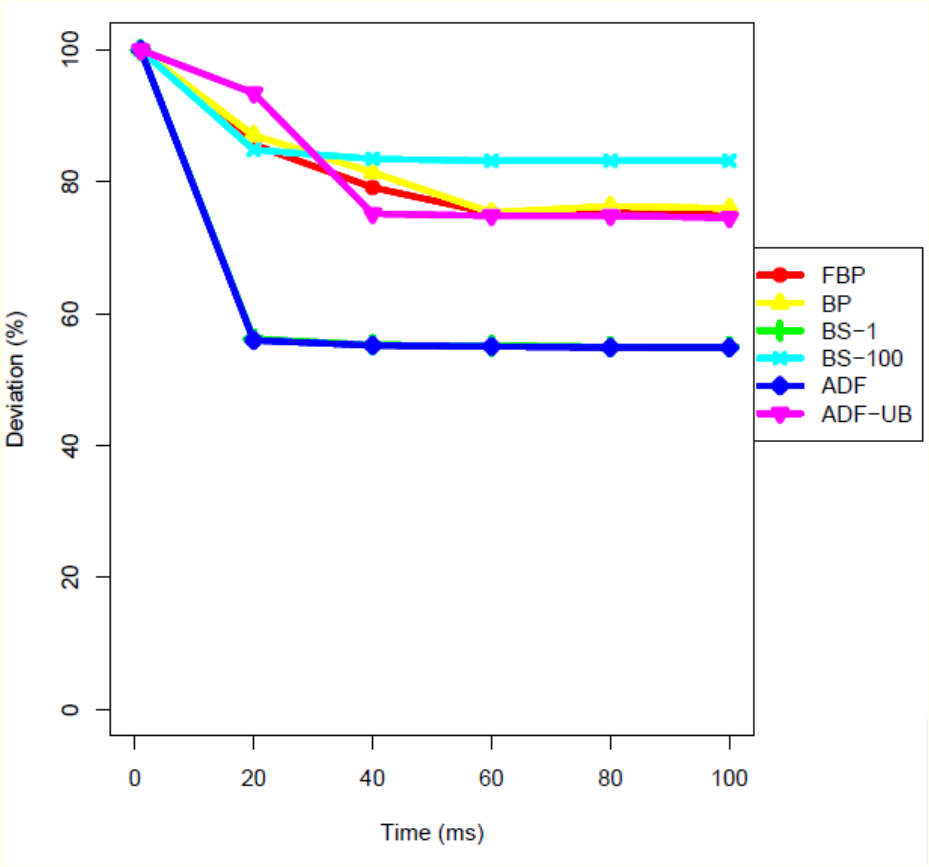


Deviation as a Function of Time – 40 milliseconds

Muta

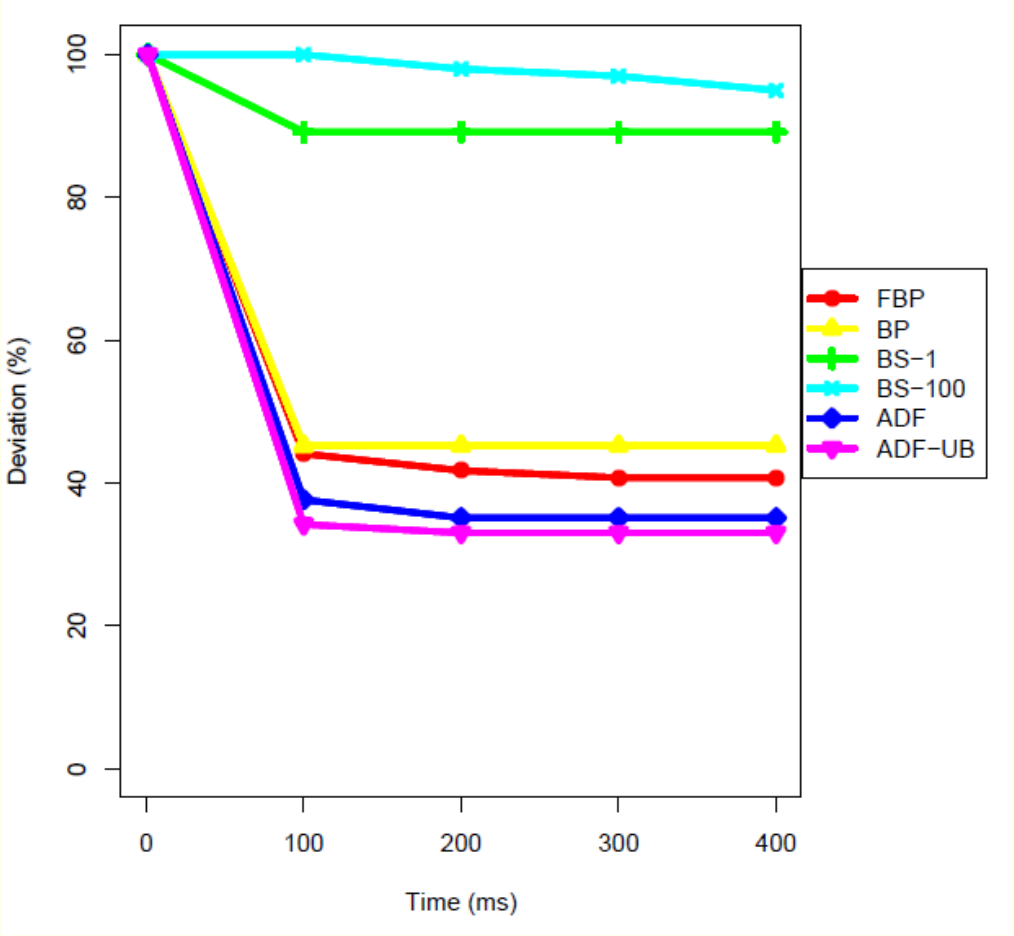


CMU

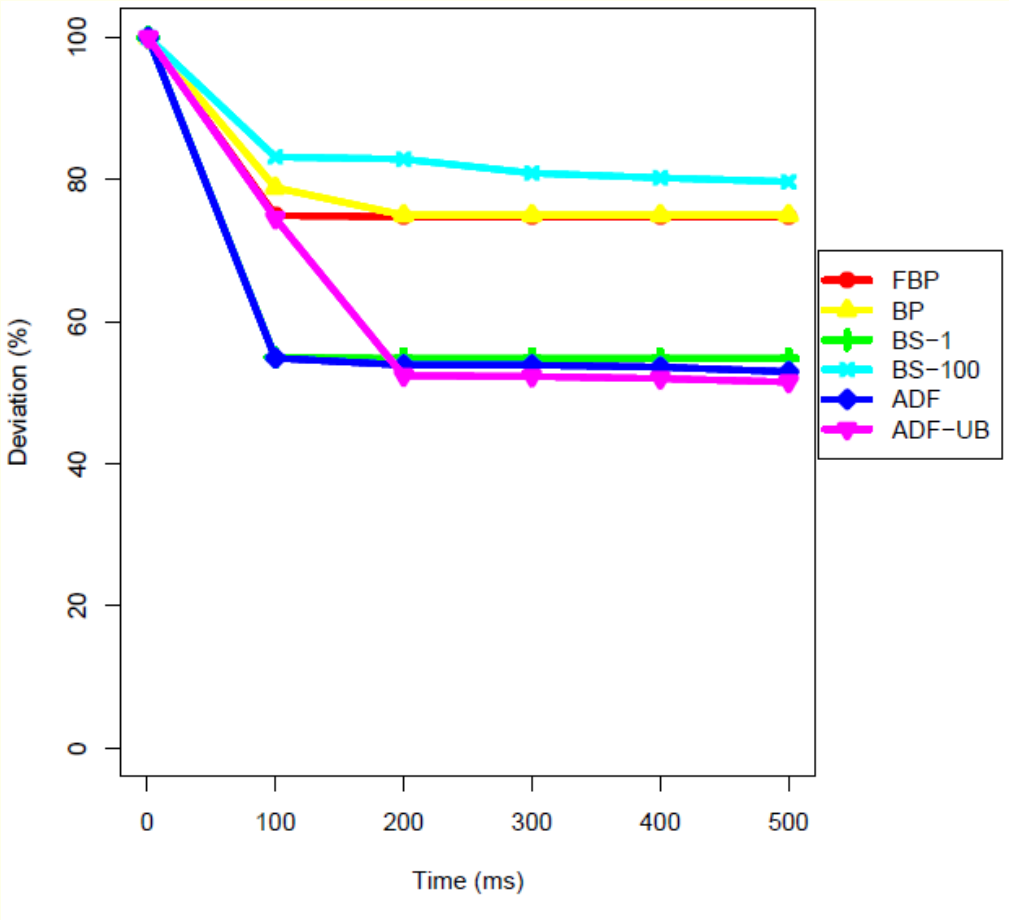


Deviation as a Function of Time – 500 milliseconds

Muta



CMU

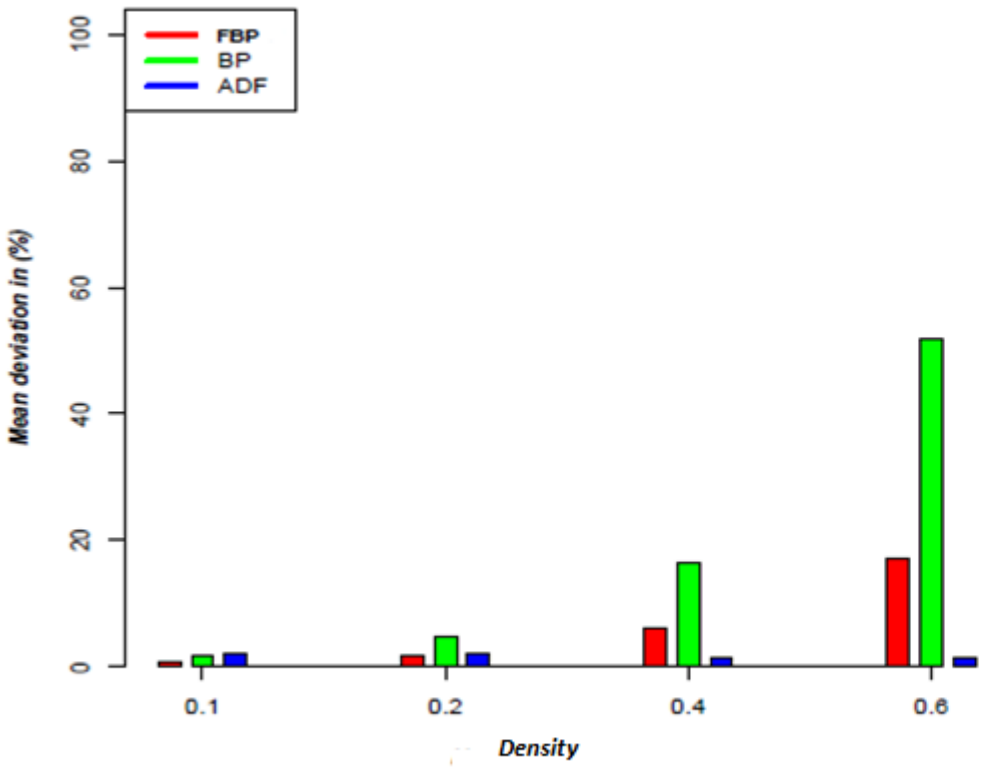


Density Scalability

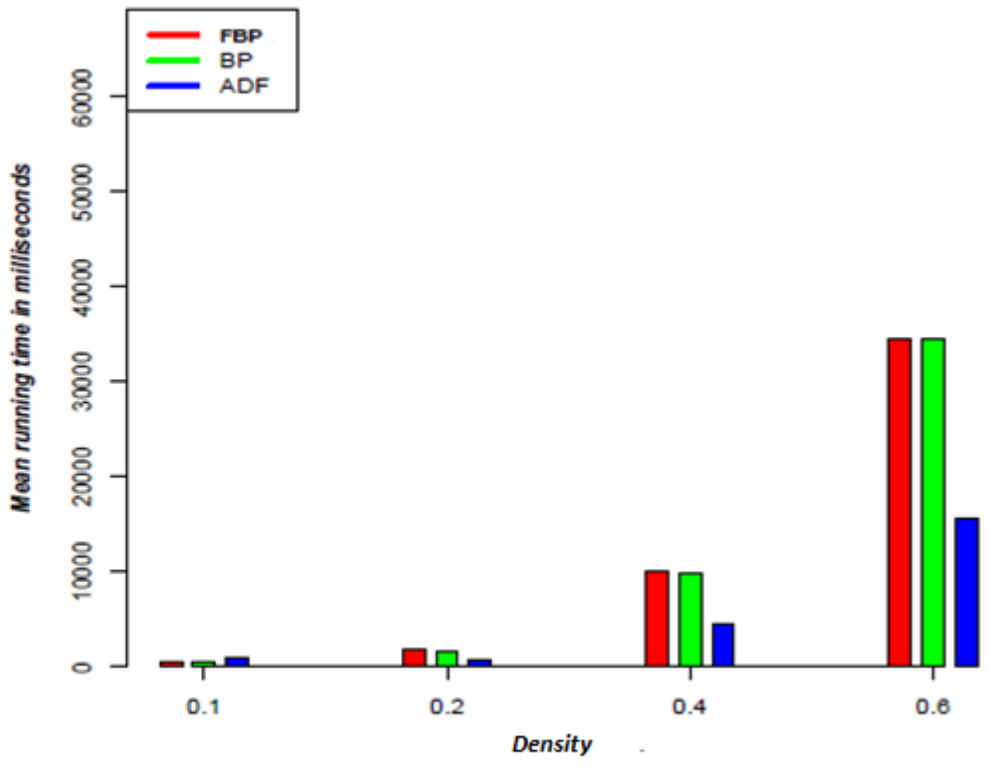
Why ADF was faster on CMU?

- FBP and BP are faster when having low density graphs
- ADF is faster when having high density graphs
- Studying the behavior of GED methods when increasing density
  - ✓ Setup Time
  - ✓ Four subsets (0.1, 0.2, 0.4 and 0.6)
  - ✓ 10 graphs of 100 vertices per subset, each subset's size is 100 vertices

ADF versus *FBP* and *BP*



(a) Deviation



(b) Running-Time

Classification under Time and Memory Constraints

- 1-*NN* Classifier
  - ✓ **GREC:** train set (286 graphs), test set (528 graphs) -- 400 milliseconds
  - ✓ **Protein:** train set (200 graphs), test set (200 graphs) -- 400 milliseconds
  - ✓ **Mutagenicity:** train set (1500 graphs), test set (2337 graphs) -- 500 milliseconds

Classification under Time and Memory Constraints

- 1-*NN* Classifier
  - ✓ **GREC:** train set (286 graphs), test set (528 graphs) -- 400 milliseconds
  - ✓ **Protein:** train set (200 graphs), test set (200 graphs) -- 400 milliseconds
  - ✓ **Mutagenicity:** train set (1500 graphs), test set (2337 graphs) -- 500 milliseconds

Methods	Classification Rate	Response Time (ms)
ADF-UB-LB	0.985	140525.48
PDFS-UB-LB	0.985	99850.79
A*-LB	0.530	222045.94
BS-1	0.985	69236.34
BP	0.985	62294.60
FBP	0.985	27922.65
H	0.962	63563.74

Classification under Time and Memory Constraints

- 1-*NN* Classifier
  - ✓ **GREC:** train set (286 graphs), test set (528 graphs) -- 400 milliseconds
  - ✓ **Protein:** train set (200 graphs), test set (200 graphs) -- 400 milliseconds
  - ✓ **Mutagenicity:** train set (1500 graphs), test set (2337 graphs) -- 500 milliseconds

Methods	Classification Rate	Response Time (ms)
ADF-UB	0.52	124361.61
PDFS-UB	0.52	80038.33
A*-LB	0.29	1065106.80
BS-100	0.26	141265.41
BP	0.52	59041.84
FBP	0.385	39425.69
H	0.43	71990.62

Classification under Time and Memory Constraints

- 1-*NN* Classifier
  - ✓ **GREC:** train set (286 graphs), test set (528 graphs) -- 400 milliseconds
  - ✓ **Protein:** train set (200 graphs), test set (200 graphs) -- 400 milliseconds
  - ✓ **Mutagenicity:** train set (1500 graphs), test set (2337 graphs) -- 500 milliseconds

Methods	Classification Rate	Response Time (ms)
ADF	0.70089	1139134.29
PDFS	0.70089	760861.518
A*	0.4574	856793.020
BS-1	0.55840	1015688.00
BP	0.70089	528546.64
FBP	0.70089	376135.51
H	0.58964	525610.25

Outline

- Introduction
- Strength and Weakness of Graph Matching Methods
- Optimization Techniques
- Metrics and Datasets for Performance Evaluation
- Experiments
- **Conclusion and Future Work**

Conclusion

- Compare graphs under time constraints
- Study deeply matching quality (graph-level)

- Focusing on Graph Edit Distance

- Adding a new methods family called «*Anytime*»
  - ✓ Proposing a Depth-First GED method
  - ✓ Converting it to Anytime (*ADF*)

- Extensions to *ADF*, to be able to match larger graphs with better quality:
  - ✓ Parallel Graph Edit Distance (*PDFS*):
    - ➢ Load balancing
  - ✓ Distributed Graph Edit Disrance (*D-DF*):
    - ➢ Master-Slave on top of Hadoop

Conclusion: (Repository + Evaluation Metrics)

- **Public repository (GDR4GED):**
  - ✓ Scalability: database decomposition
  - ✓ https://iapr-tc15.greyc.fr/links.html#Benchmarking and data sets
- **Additional low level annotation:**
  - ✓ Best distance and vertex-vertex matchings
  - ✓ 7 methods (4 exact and 3 approximate)
  - ✓ Four databases (GREC, Mutagenicity, Protein and CMU)
- **Metrics:**
  - ✓ Under soft and hard time constraints
  - ✓ Mertics for each pair of graphs
  - ✓ Metrics for a full dataset

Conclusion: (Experiments)

- **Soft Constraints:** *PDFS* and *D-DF* had the minimum deviation
  - ✓ Except on MUTA
- **Hard Constraints:**
  - ✓ *PDFS* is slow, however, it was always the most precise one
- **Deviation as a Function of Time:**
  - ✓ *FBP* and *BP* were faster on less dense graphs
  - ✓ *ADF* was faster on more dense graphs
- **Classification Tests:**
  - ✓ *PDFS* was the best exact GED method

Perspectives

- ***ADF*:**
  - ✓ Better lower bounds
  - ✓ Learn:
    - ➢ Lower bounds
    - ➢ Sort vertices of G_1
    - ➢ Stop the algorithm when the quality is sufficient
- ***PDFS:***
  - ✓ Extending *PDFS* to multi-machines (MPI)
- ***D-DF:***
  - ✓ Extending *D-DF* to multi-jobs algorithm

Perspectives

- **GDR4GED:**
  - ✓ Integrating better answers in the database (if found)
  - ✓ Expanding this repository by integrating other publicly (available databases), dense graphs, etc.

- **A general perspective:**
  - ✓ Knowing when to use one algorithm or another

Thanks for your attention …