Faster Algorithms for the Maximum Common Subtree Isomorphism Problem

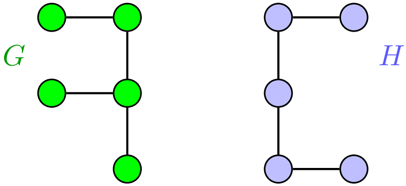
Andre Droschinsky[1] Nils Kriege[1] Petra Mutzel[1]

[1]Dept. of Computer Science, Technische Universität Dortmund, Germany

41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)
August 22 – 26, 2016

Maximum Common Subtree Isomorphism (MCSI) Problem

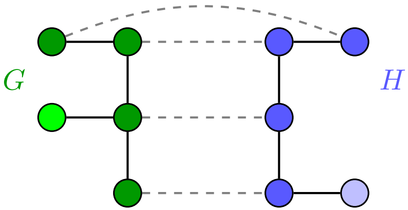
Input: Trees G and H



Maximum Common Subtree Isomorphism (MCSI) Problem

Input: Trees G and H

Output: An isomorphism between subtrees of G and H with the maximum possible number of vertices



Motivation

- Trees or graphs are often used as an abstract representation for e.g. molecules, XML, or social networks

Motivation

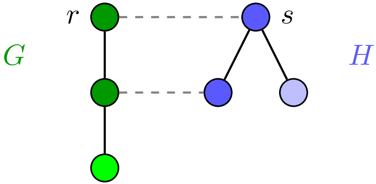
- Trees or graphs are often used as an abstract representation for e.g. molecules, XML, or social networks
- Natural measurement of similarity.

Motivation

- Trees or graphs are often used as an abstract representation for e.g. molecules, XML, or social networks
- Natural measurement of similarity.
- Application examples:
  - Chemistry, Biology [Ehrlich and Rarey, 2011]
  - Computer Vision [Englert and Kovács, 2015]
  - Binary Programs [Gao et al., 2008]

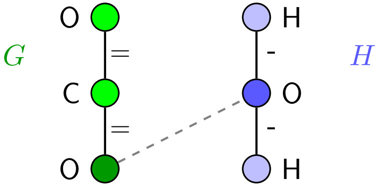
Problem variants

- Rooted MCSI
  - Map roots of input trees



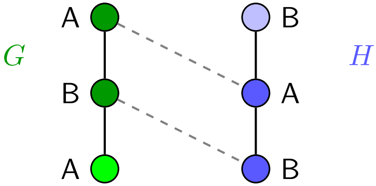
Problem variants

- Rooted MCSI
  - Map roots of input trees
- Labelled Trees
  - Labels on vertices/edges must match



Problem variants

- Rooted MCSI
  - Map roots of input trees
- Labelled Trees
  - Labels on vertices/edges must match
- Weight function $w : V_G \times V_H \to \mathbb{R}$ on pairs of vertices
  - Maximize weight
    - Example: $w(\text{A,A}) = w(\text{B,B}) = 2,\ w(\text{A,B}) = w(\text{B,A}) = 1$



Related problems

- Maximum Common Subgraph Isomorphism
    - NP-hard

Related problems

- Maximum Common Subgraph Isomorphism
  - NP-hard
- Subgraph Isomorphism
  - NP-hard

Related problems

- Maximum Common Subgraph Isomorphism
  - NP-hard
- Subgraph Isomorphism
  - NP-hard
- Graph Isomorphism
  - Unknown if NP-hard or in P

Related problems, cont.

- Tree Isomorphism: Solvable $\mathcal{O}(n)$

Related problems, cont.

- Tree Isomorphism: Solvable $\mathcal{O}(n)$
- Subtree Isomorphism, polynomial time solvable
  - $\mathcal{O}(n^{2.5})$, rooted trees [Matula, 1978]
  - $\mathcal{O}(n^{2.5})$, unrooted trees [Chung, 1987]
  - $\mathcal{O}(n^{2.5}/\log n)$, unrooted trees [Shamir and Tsur, 1999]

Related problems, cont.

- Tree Isomorphism: Solvable $\mathcal{O}(n)$
- Subtree Isomorphism, polynomial time solvable
  - $\mathcal{O}(n^{2.5})$, rooted trees [Matula, 1978]
  - $\mathcal{O}(n^{2.5})$, unrooted trees [Chung, 1987]
  - $\mathcal{O}(n^{2.5}/\log n)$, unrooted trees [Shamir and Tsur, 1999]
- MCSI, rooted
  - $\mathcal{O}(n^{2.5}\log n)$, integer weights in $\mathcal{O}(n)$
  - $\mathcal{O}(n^3)$, unrestricted weights [Valiente, 2002]

Related problems, cont.

- Tree Isomorphism: Solvable $\mathcal{O}(n)$
- Subtree Isomorphism, polynomial time solvable
  - $\mathcal{O}(n^{2.5})$, rooted trees [Matula, 1978]
  - $\mathcal{O}(n^{2.5})$, unrooted trees [Chung, 1987]
  - $\mathcal{O}(n^{2.5}/\log n)$, unrooted trees [Shamir and Tsur, 1999]
- MCSI, rooted
  - $\mathcal{O}(n^{2.5}\log n)$, integer weights in $\mathcal{O}(n)$
  - $\mathcal{O}(n^{3})$, unrestricted weights [Valiente, 2002]
- MCSI on unrooted trees: $\mathcal{O}(n^{5})$ [Matula, 1978]

Our contribution, main result

MCSI in time $\mathcal{O}(|G||H|\Delta(G,H)) \subseteq \mathcal{O}(n^3)$

- Unrooted trees G and H
- Weight function
- $\Delta(G,H) := \min\{\Delta(G),\Delta(H)\} + \log\max\{\Delta(G),\Delta(H)\}$

In the following

1) Dynamic programming for MCSI between rooted trees, $\mathcal{O}(n^3)$

In the following

1) Dynamic programming for MCSI between rooted trees, $\mathcal{O}(n^3)$
2) Adaption to unrooted trees, $\mathcal{O}(n^5)$

In the following

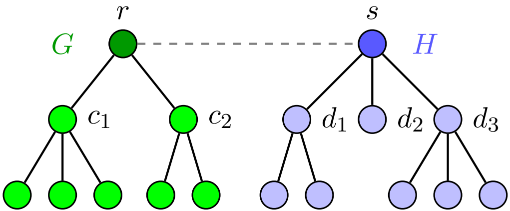
1) Dynamic programming for MCSI between rooted trees, $\mathcal{O}(n^3)$
2) Adaption to unrooted trees, $\mathcal{O}(n^5)$
3) Improve the upper bound to $\mathcal{O}(|G||H|\Delta(G,H))$
   - Reduce the number of subproblems
   - Efficiently solve similar subproblems

In the following

1) Dynamic programming for MCSI between rooted trees, $\mathcal{O}(n^3)$
2) Adaption to unrooted trees, $\mathcal{O}(n^5)$
3) Improve the upper bound to $\mathcal{O}(|G||H|\Delta(G,H))$
    - Reduce the number of subproblems
    - Efficiently solve similar subproblems
4) Lower bounds

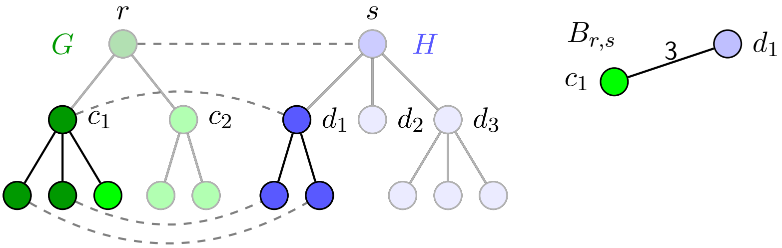
Dynamic Programming Approach between rooted trees

- Rooted trees G and H, $r \mapsto s$



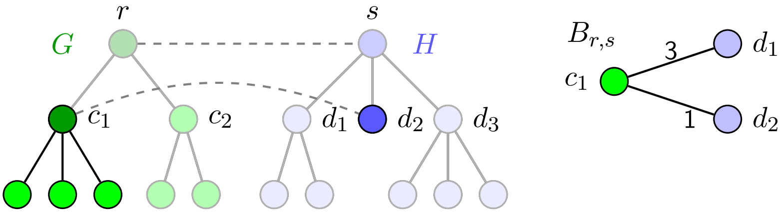
Dynamic Programming Approach between rooted trees

- Rooted trees G and H, $r \mapsto s$
- Recursively compute rooted MCSIs between children of r and s and their descendants



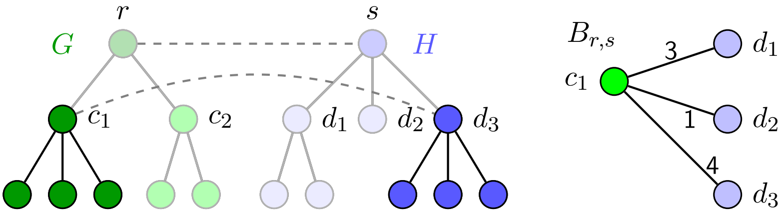
Dynamic Programming Approach between rooted trees

- Rooted trees G and H, $r \mapsto s$
- Recursively compute rooted MCSIs between children of r and s and their descendants



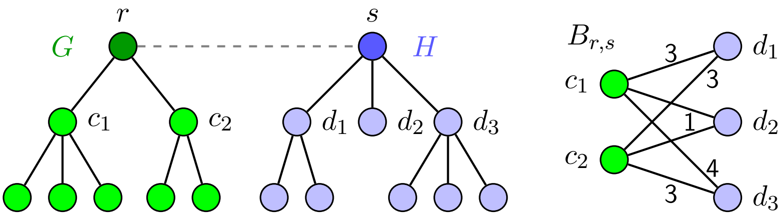
Dynamic Programming Approach between rooted trees

- Rooted trees G and H, $r \mapsto s$
- Recursively compute rooted MCSIs between children of r and s and their descendants



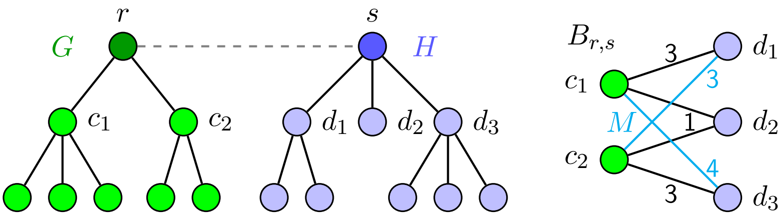
Dynamic Programming Approach between rooted trees

- Rooted trees G and H, $r \mapsto s$
- Recursively compute rooted MCSIs between children of r and s and their descendants



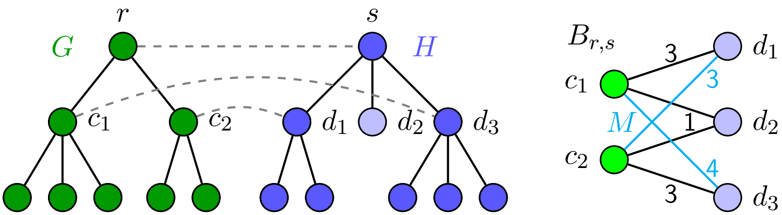
Dynamic Programming Approach between rooted trees

- Rooted trees G and H, $r \mapsto s$
- Recursively compute rooted MCSIs between children of r and s and their descendants
- Compute a Maximum Weight Matching M on $B_{r,s}$



Dynamic Programming Approach between rooted trees

- Rooted trees G and H, $r \mapsto s$
- Recursively compute rooted MCSIs between children of r and s and their descendants
- Compute a Maximum Weight Matching M on $B_{r,s}$
- Edges of M determine MCSI between G and H



Computation time for an MCSI between two rooted trees

- Compute $\mathcal{O}(n^2)$ Maximum Weight Matchings (MWMs)
  - ≤ 1 MWM for each pair $(u, v) \in V_G \times V_H$

Computation time for an MCSI between two rooted trees

- Compute $\mathcal{O}(n^2)$ Maximum Weight Matchings (MWMs)
  - ≤ 1 MWM for each pair $(u, v) \in V_G \times V_H$
  - Solvable in time $\mathcal{O}(kl(k + \log l))$, $k + l$ vertices, $k \leq l$

Computation time for an MCSI between two rooted trees

- Compute $\mathcal{O}(n^2)$ Maximum Weight Matchings (MWMs)
  - ≤ 1 MWM for each pair $(u, v) \in V_G \times V_H$
  - Solvable in time $\mathcal{O}(kl(k + \log l))$, $k + l$ vertices, $k \leq l$
- Total time $\mathcal{O}(n^3)$

Basic approach for MCSI between two unrooted trees

- Compute rooted MCSIs for all roots $(r, s) \in V_G \times V_H$

Basic approach for MCSI between two unrooted trees

- Compute rooted MCSIs for all roots $(r, s) \in V_G \times V_H$
- Select maximum solution

Basic approach for MCSI between two unrooted trees

- Compute rooted MCSIs for all roots $(r, s) \in V_G \times V_H$
- Select maximum solution
- Total time $\mathcal{O}(n^5)$

Basic approach for MCSI between two unrooted trees

- Compute rooted MCSIs for all roots $(r, s) \in V_G \times V_H$
- Select maximum solution
- Total time $\mathcal{O}(n^5)$

Improvement 1: Reduce the number of subproblems

- Determine arbitrary root $r \in V_G$
- Consider all $s \in V_H$ as root

Basic approach for MCSI between two unrooted trees

- Compute rooted MCSIs for all roots $(r, s) \in V_G \times V_H$
- Select maximum solution
- Total time $\mathcal{O}(n^5)$

Improvement 1: Reduce the number of subproblems

- Determine arbitrary root $r \in V_G$
- Consider all $s \in V_H$ as root
  - Total time $\mathcal{O}(n^4)$

Basic approach for MCSI between two unrooted trees

- Compute rooted MCSIs for all roots $(r, s) \in V_G \times V_H$
- Select maximum solution
- Total time $\mathcal{O}(n^5)$

Improvement 1: Reduce the number of subproblems

- Determine arbitrary root $r \in V_G$
- Consider all $s \in V_H$ as root
  - Total time $\mathcal{O}(n^4)$
- Correctness
  - If $\exists$ MCSI ϕ, where $r \in \text{dom}(\phi)$: Roots $(r, \phi(r))$ yield an MCSI between G, H ✓

Basic approach for MCSI between two unrooted trees

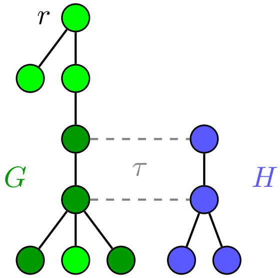
- Compute rooted MCSIs for all roots $(r, s) \in V_G \times V_H$
- Select maximum solution
- Total time $\mathcal{O}(n^5)$

Improvement 1: Reduce the number of subproblems

- Determine arbitrary root $r \in V_G$
- Consider all $s \in V_H$ as root
  - Total time $\mathcal{O}(n^4)$
- Correctness
  - If $\exists$ MCSI ϕ, where $r \in \text{dom}(\phi)$: Roots $(r, \phi(r))$ yield an MCSI between G, H ✓
  - If not, returned solution is no MCSI, but...

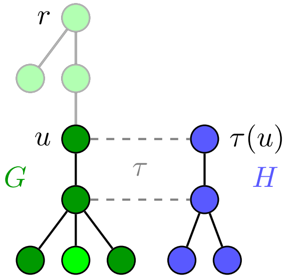
Reduce the number of subproblems, cont.

- Given: $\nexists$ MCSI ϕ, where $r \in \text{dom}(\phi)$; let τ be any MCSI



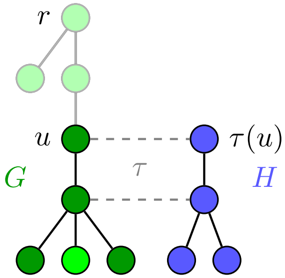
Reduce the number of subproblems, cont.

- Given: $\nexists$ MCSI ϕ, where $r \in \text{dom}(\phi)$; let τ be any MCSI
- For each $u \in V_G$ and its descendants compute rooted MCSI between this subtree and H



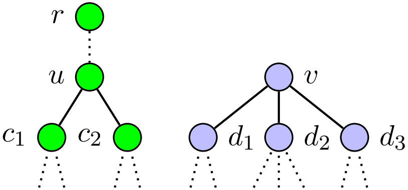
Reduce the number of subproblems, cont.

- Given: $\nexists$ MCSI ϕ, where $r \in \text{dom}(\phi)$; let τ be any MCSI
- For each $u \in V_G$ and its descendants compute rooted MCSI between this subtree and H
- Total time $\mathcal{O}(n^4)$



Improvement 2: Efficiently solve similar subproblems

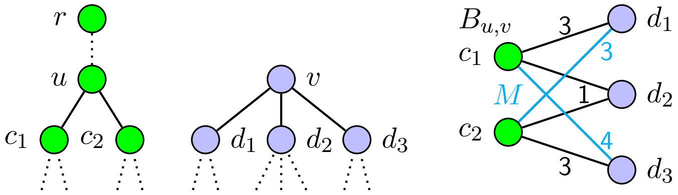
- Let G be rooted at r
- Let $k =$ degree of u, $l =$ degree of v



Improvement 2: Efficiently solve similar subproblems

- Let G be rooted at r
- Let k = degree of u, l = degree of v

1) Root $v \in V_H$: bipartite graph $B_{u,v}$ and MWM M

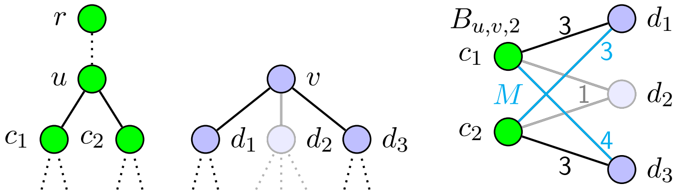


1) Time $\mathcal{O}(kl(\min\{k,l\} + \log\max\{k,l\}))$,

Improvement 2: Efficiently solve similar subproblems

- Let G be rooted at r
- Let $k =$ degree of u, $l =$ degree of v

1) Root $v \in V_H$: bipartite graph $B_{u,v}$ and MWM M
2) Root $d_2 \in V_H$: bipartite graph $B_{u,v,2}$ and MWM M

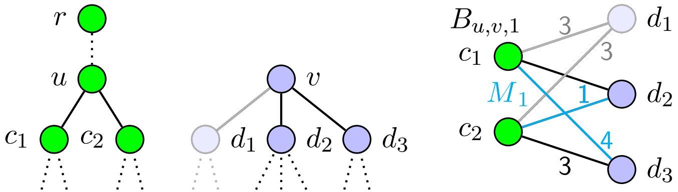


1) Time $\mathcal{O}(kl(\min\{k,l\} + \log\max\{k,l\}))$, 2) $\mathcal{O}(1)$,

Improvement 2: Efficiently solve similar subproblems

- Let G be rooted at r
- Let $k =$ degree of u, $l =$ degree of v

1) Root $v \in V_H$: bipartite graph $B_{u,v}$ and MWM M
2) Root $d_2 \in V_H$: bipartite graph $B_{u,v,2}$ and MWM M
3) Root $d_j \in V_H$, $j \in \{1,3\}$: derive MWMs M_j from M



1) Time $\mathcal{O}(kl(\min\{k,l\} + \log \max\{k,l\}))$, 2) $\mathcal{O}(1)$, 3) as 1) for all M_j

Theorem

An MCSI between two trees G and H can be computed in time $\mathcal{O}(|G||H|\Delta(G,H))$.

- $\Delta(G,H) := \min\{\Delta(G),\Delta(H)\} + \log\max\{\Delta(G),\Delta(H)\}$

Theorem

An MCSI between two trees G and H can be computed in time $\mathcal{O}(|G||H|\Delta(G,H))$.

- $\Delta(G,H) := \min\{\Delta(G),\Delta(H)\} + \log\max\{\Delta(G),\Delta(H)\}$

Proposition

For trees of bounded degree we obtain running time $\Theta(|G||H|)$, which is optimal.

Lower bound for unrestricted degree

- Running time for trees with n nodes and unrestricted degree: $\mathcal{O}(n^3)$

Lower bound for unrestricted degree

- Running time for trees with n nodes and unrestricted degree: $\mathcal{O}(n^3)$
- The best known time bound for the assignment problem in a graph with n nodes and $\Theta(n^2)$ edges of unrestricted weight since more than 30 years: $\mathcal{O}(n^3)$

Lower bound for unrestricted degree

- Running time for trees with n nodes and unrestricted degree: $\mathcal{O}(n^3)$
- The best known time bound for the assignment problem in a graph with n nodes and $\Theta(n^2)$ edges of unrestricted weight since more than 30 years: $\mathcal{O}(n^3)$
- There is a linear time reduction from the assignment problem to MCSI.

Lower bound for unrestricted degree

- Running time for trees with n nodes and unrestricted degree: $\mathcal{O}(n^3)$
- The best known time bound for the assignment problem in a graph with n nodes and $\Theta(n^2)$ edges of unrestricted weight since more than 30 years: $\mathcal{O}(n^3)$
- There is a linear time reduction from the assignment problem to MCSI.
- Time $o(n^3)$ for MCSI $\Rightarrow$ Time $o(n^3)$ for the assignment problem.

Conclusion

We developed/showed

- An algorithm for MCSI, time $\mathcal{O}(|G||H|\Delta(G, H))$
- Achieved this improved time bound by:
  - Rooting one tree, but also consider subtrees of this tree
  - Efficiently solving similar subproblems
- Results about optimality

Next up

- Improve time bound for MCSI on unweighted trees
- Adapt our algorithm to other graph classes

Conclusion

We developed/showed

- An algorithm for MCSI, time $\mathcal{O}(|G||H|\Delta(G,H))$
- Achieved this improved time bound by:
  - Rooting one tree, but also consider subtrees of this tree
  - Efficiently solving similar subproblems
- Results about optimality

Next up

- Improve time bound for MCSI on unweighted trees
- Adapt our algorithm to other graph classes

A. Abboud, A. Backurs, T. D. Hansen, V. V. Williams, and O. Zamir. Subtree isomorphism revisited. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1256–1271. SIAM, 2016. ISBN 978-1-611974-33-1.

A. Droschinsky, B. Heinemann, N. Kriege, and P. Mutzel. Enumeration of maximum common subtree isomorphisms with polynomial-delay. In H.-K. Ahn and C.-S. Shin, editors, *Algorithms and Computation (ISAAC)*, LNCS, pages 81–93. Springer, 2014. ISBN 978-3-319-13074-3. doi:10.1007/978-3-319-13075-0_7.

H.-C. Ehrlich and M. Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(1):68–79, 2011. ISSN 1759-0884. doi:10.1002/wcms.5. URL `http://dx.doi.org/10.1002/wcms.5`.

D. W. Matula. Subtree isomorphism in $O(n^{5/2})$. In P. H. B. Alspach and D. Miller, editors, *Algorithmic Aspects of Combinatorics*, volume 2 of *Annals of Discrete Mathematics*, pages 91–106. Elsevier, 1978. doi:10.1016/S0167-5060(08)70324-8.

G. Valiente. *Algorithms on Trees and Graphs*. Springer-Verlag, Berlin, 2002. ISBN 3-540-43550-6.