Computing Quartet Distance Is Equivalent to Counting 4-Cycles

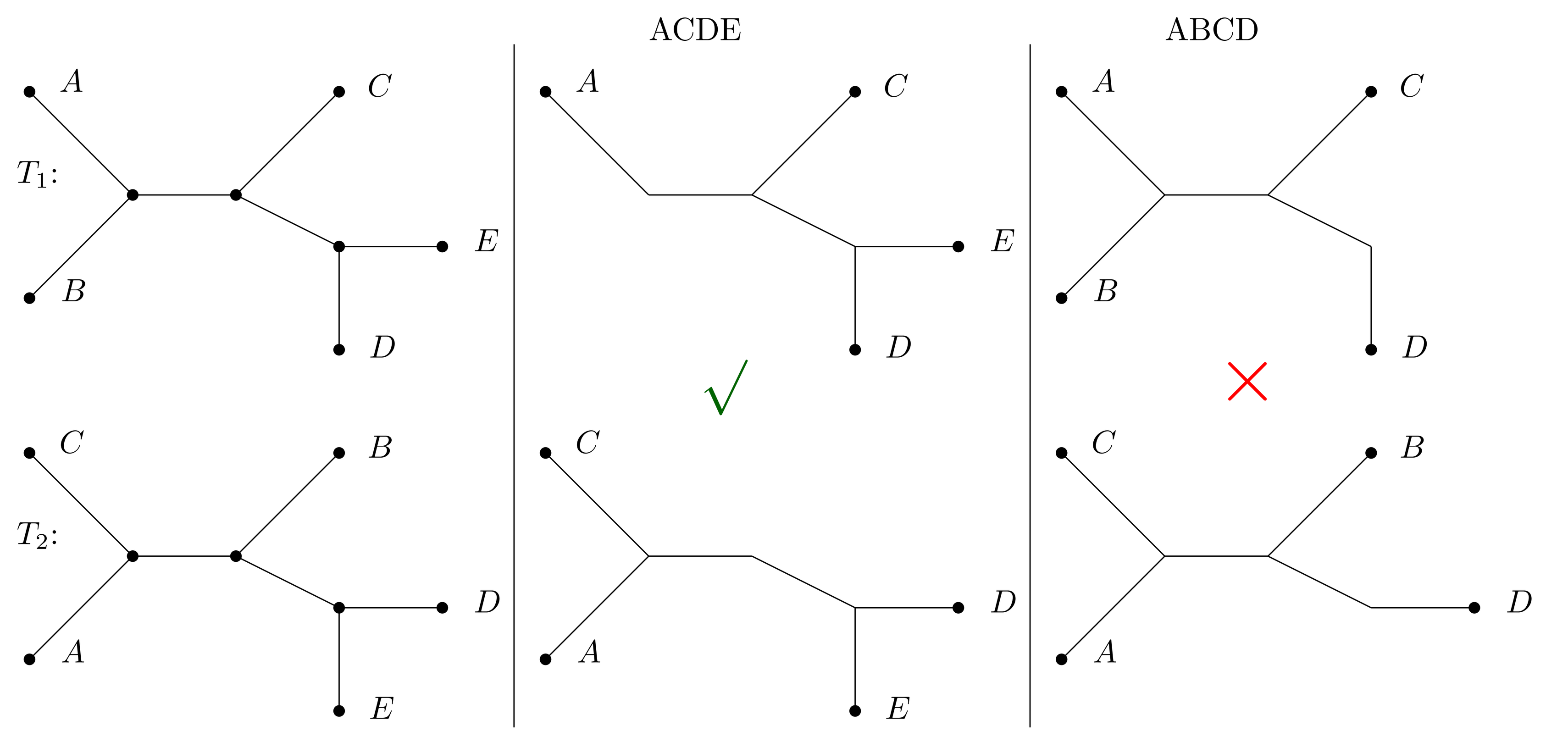
Bartłomiej Dudek* and Paweł Gawrychowski

University of Wrocław, Poland

* Supported by the National Science Centre, Poland, under grant number 2017/27/N/ST6/02719.

Quartet distance

Input: two trees with the same set of leaves
Output: number of quartets of leaves inducing different topologies

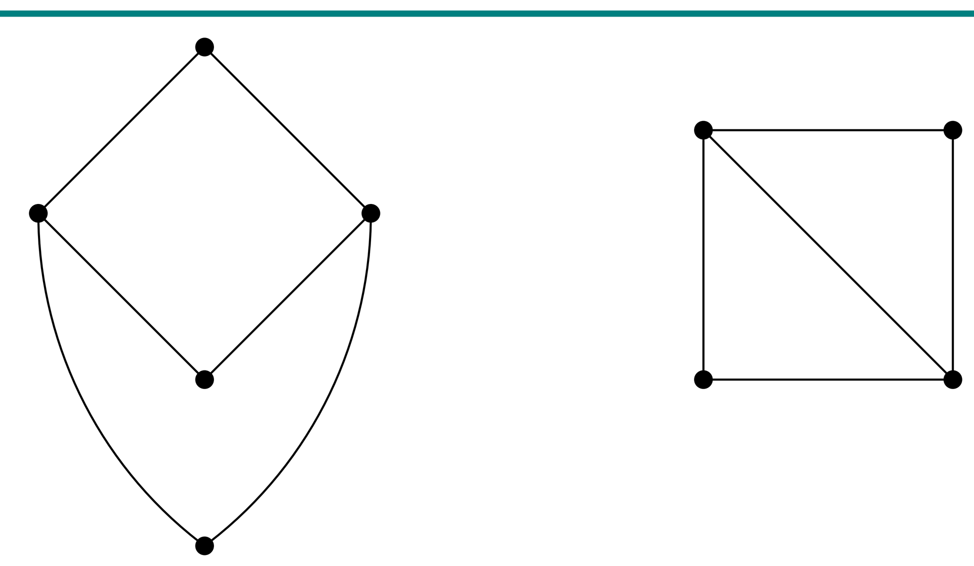


History of quartet distance

Year & Venue	Authors	Runtime	Arbitrary degree
	Folklore	$\mathcal{O}(n^4)$	✓
SystBiol 1993	Steel and Penny	$\mathcal{O}(n^3)$	×
SODA 2000	Bryant et al.	$\mathcal{O}(n^2)$	×
ISAAC 2001	Brodal et al.	$\mathcal{O}(n\log^2 n)$	×
ISAAC 2004	Brodal et al.	$\mathcal{O}(n\log n)$	×
APBC 2007	Stissing et al.	$\mathcal{O}(d^9 n\log n)$	✓
AlMoB 2011	Nielsen et al.	$\mathcal{O}(n^{2.688})$	✓
SODA 2013	Brodal et al.	$\mathcal{O}(dn\log n)$	✓
	Our contribution		
STOC 2019		$\mathcal{O}(n^{1.48})$	✓
		$\mathcal{O}(d^{0.77}n)$	✓
		no $\mathcal{O}(n^{4/3-\varepsilon})$ (probably)	✓

Counting 4-cycles

Input: simple, undirected graph
Output: number of simple cycles of length 4



3 1

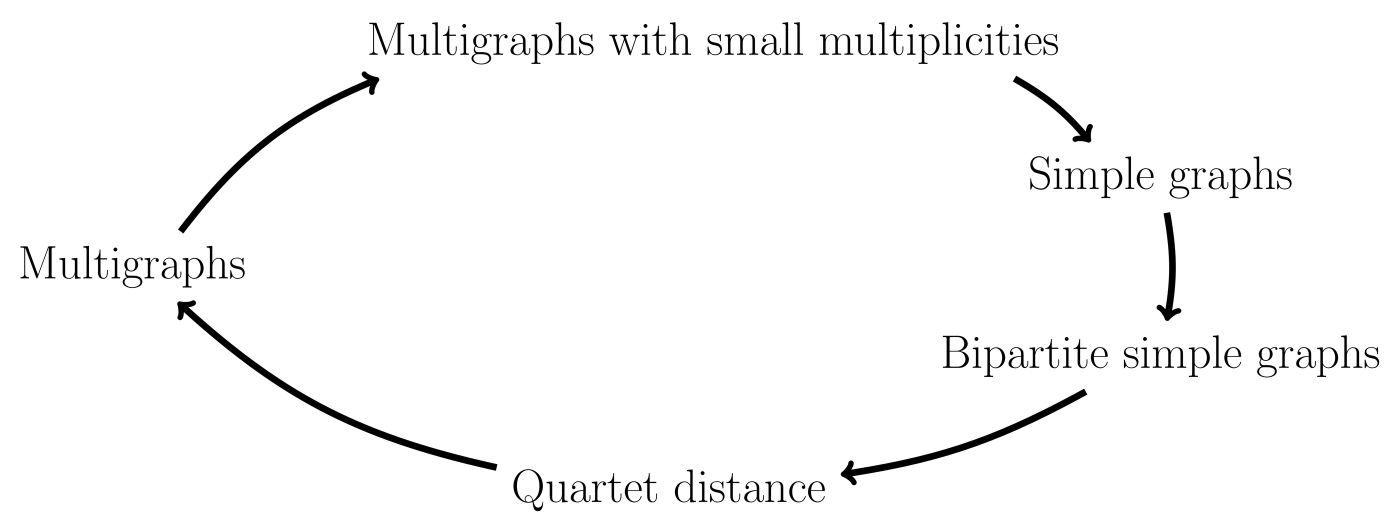
History of $2k$-cycles

Year & Venue	Authors	Runtime	Variant
	Folklore	$\mathcal{O}(n^3)$	
JCombTheory 1974	Bondy and Simonovits	$m \geq 100kn^{1+1/k}$	$\implies$ exists $2k$-cycle
Algorithmica 1997	Alon et al.	$\mathcal{O}(n^\omega)$	count 4-cycles
		$\mathcal{O}(m^{4/3})$	find a 4-cycle
JDiscrMath 1997	Yuster and Zwick	$\mathcal{O}(n^2)$	find a $2k$-cycle
SODA 2015	Vassilevska Williams et al.	$\mathcal{O}(m^{1.48})$	count 4-cycles
STOC 2017	Dahlgaard et al.	$\mathcal{O}(m^{2k/(k+1)})$	find a $2k$-cycle

Conjecture [Dahlgaard et al., STOC'17]

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(m^{4/3-\varepsilon})$ time.

Our contribution



(*) All reductions are up to polylogarithmic factors.

Summary

1. An $\mathcal{O}(n^\delta)$-time algorithm for quartet distance gives $\mathcal{O}(m^\delta)$-time algorithm for counting 4-cycles in simple graphs.
   $\implies$ probably no $\mathcal{O}(n^{4/3-\varepsilon})$-time algorithm for QD
2. Counting 4-cycles in simple graphs in $\mathcal{O}(m^\delta)$ time gives $\tilde{\mathcal{O}}(n^\delta)$-time algorithm for quartet distance.
   $\implies$ an $\mathcal{O}(n^{1.48})$-time algorithm for QD

Two types of 4-edge shapes

Shapes counted in linear time: ⋲, ⋜, ⋞ and ⋚:

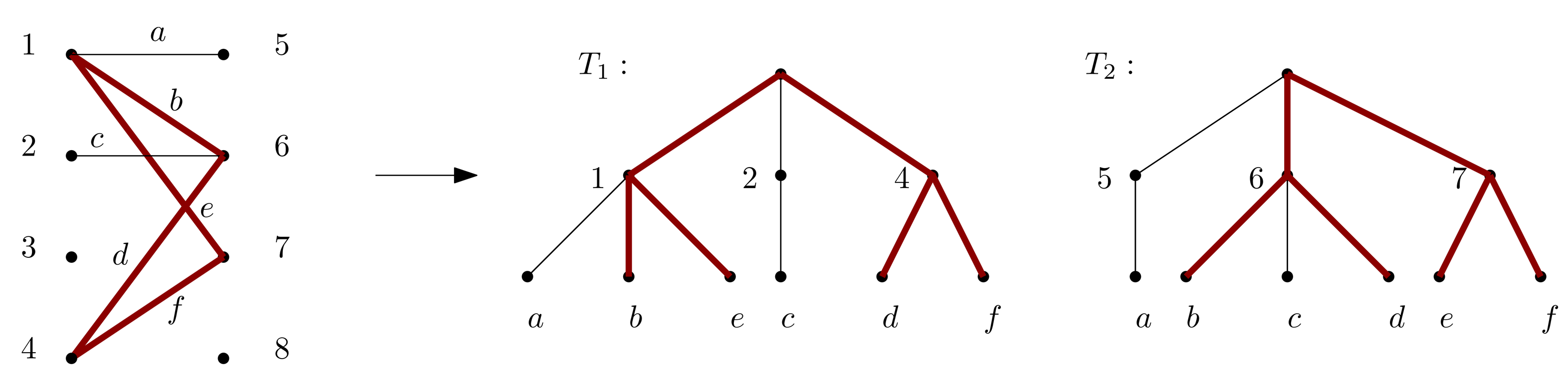
$$\#⋲ = \sum_{v\in V_1}\binom{\deg(v)}{4}$$

$$\#⋜ = \sum_{(u,v)\in E}\binom{d(u)-1}{2}(d(v)-1)$$

Shapes equivalent to 4-cycles: ⋚, ⋚, ⋚, ⋚:

$$\#⋚ = \ldots - 2\cdot\#⧖$$
$$\#\equiv = \ldots + 1\cdot\#⧖$$

$\#⧖$ (bipartite graphs) $\to$ quartet distance

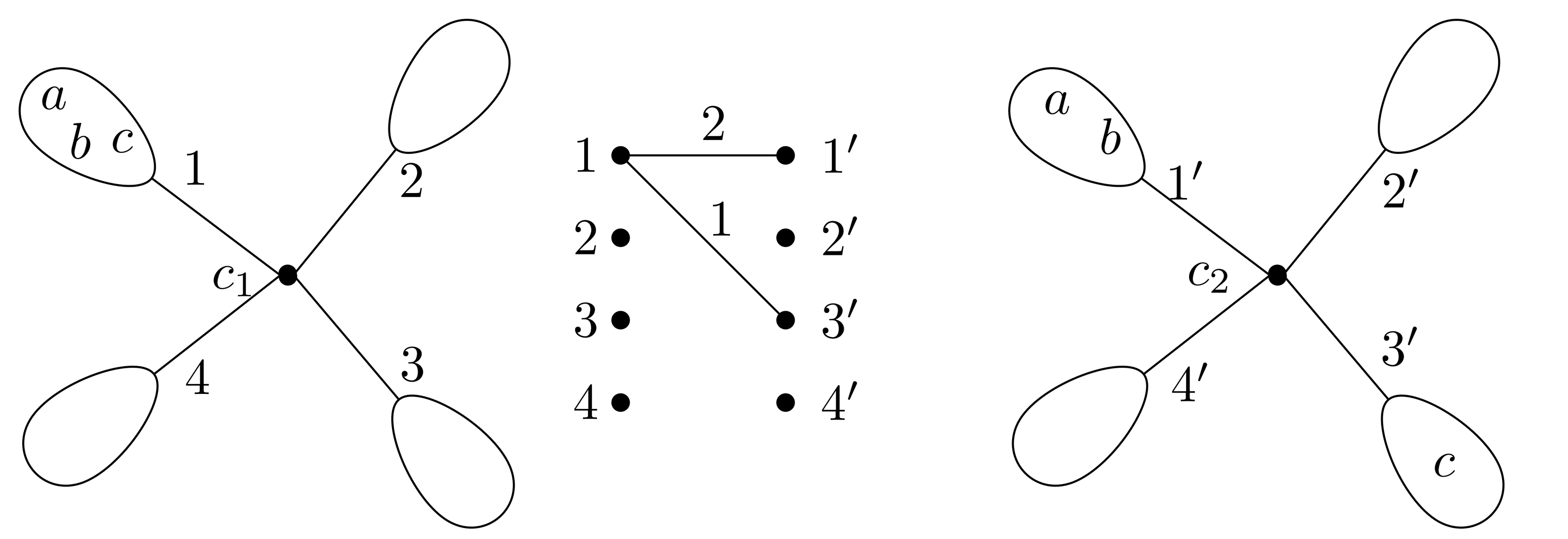


$$\binom{\#\text{ of leaves}}{4} - \mathtt{QD}(T_1,T_2) = (\#⋲) + (\#⋺) + (\#⋞) + (\#⋟) + (\#\equiv) + (\#⋚)$$

$$\mathtt{QD}(T_1,T_2) = \ldots + (\#⧖)$$

Quartet distance $\to$ $(\#\equiv)$(multigraphs)

Shared ≻≺ : $\mathcal{O}(n\log n)$ algorithm by Brodal et al. [SODA'13].
Shared ╳ : consider all pairs (c_1, c_2) of central nodes and count 4-matchings $\equiv$.



Counting all shared ╳ $\implies$ many instances of $(\#\equiv)$ in multigraphs of total size $\tilde{\mathcal{O}}(n)$.