A Study on Fixed-Parameter Algorithms and Property Testing

Student: Joseph Chuang-Chieh Lin

Advisors: Professor Maw-Shang Chang
Professor Peter Rossmanith

Department of Computer Science and Information Engineering
National Chung Cheng University
Taiwan

26 July 2011

Outline

1. Introduction
   - Fixed-parameter algorithms
   - Property testing
   - Motivations
2. Our contributions

Introduction:
Fixed-parameter algorithms

Fixed-parameter algorithms

A parameterized problem

A language $L \subseteq \Sigma^* \times \mathbb{Z}^+$ that consists of input pairs (I, k).

- The first component: problem instance (Σ: finite alphabet).
- The second component: parameter.

- Fixed-parameter algorithms:
  - Solving parameterized problems in $O(f(k) \cdot \text{poly}(n))$ time.
  - $f(k)$: an arbitrary function solely depending on k.
- Characteristics of fixed-parameter algorithms:
  - They can be used to solve **NP**-hard problems exactly.
  - Efficient when k is small.
- Problems admit such algorithms: **FPT** (fixed-parameter tractable).

Fixed-parameter algorithms (contd.)

This research field has been very active since 1990s.

- Articles in this field appear in important conferences.
  - FOCS, STOC, SODA, ICALP, STACS, MFCS, SWAT, ISAAC, ESA, WG, IPEC, etc.
  - IPEC: International Symposium on Parameterized and Exact Computation.
- At least three monographs & books in this field.
  - R. G. Downey & M. R. Fellows: *Parameterized Complexity*. Springer-Verlag, New York, 1999.
  - J. Flum & M. Grohe: *Parameterized Complexity Theory*. Springer-Verlag, 2006.
  - R. Niedermeier: *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
- Well-known experts in this field:
  - H. L. Bodlaender, J. Chen, E. Demaine, M. R. Fellows, R. Niedermeier, P. Rossmanith, etc.

Fixed-parameter algorithms (contd.)

- The main tasks in this field:
  - Find the "parameters" which make **NP**-hard problems difficult.
  - Improve known **FPT** results.
    - e.g., $O(2^k n^2) \rightarrow O(1.62^k n^2) \rightarrow \ldots \rightarrow O(1.2738^k + kn)$.
- Common types of parameters:
  - Target size (e.g., the size of a vertex cover)
  - Structure of the input (e.g., the treewidth of a graph)

A parameter associated with the target: vertex cover

- Given a graph $G = (V, E)$,

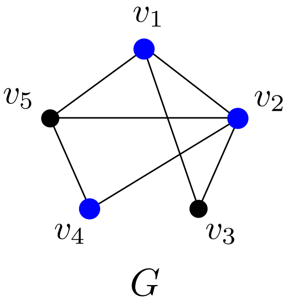
  $C \subseteq V$ is a vertex cover:

  each edge in E has at least one of its endpoints in C.

- The Vertex Cover problem:

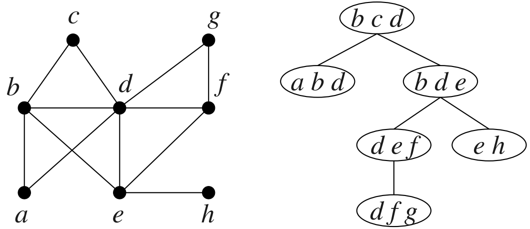
  Determine if a graph has a vertex cover of size $\leq k$:

  ▷ **FPT**: $O(1.2738^k + kn)$ [Chen *et al.* 2010].



A parameter associated with the input structure: treewidth

- Tree-decomposition of a graph:



- The *width* of the tree-decomposition: $|\text{maximum bag}| - 1$.
- The treewidth of G: the minimum width over all tree-decompositions of G.

A parameter associated with the input structure: treewidth

- Roughly speaking, treewidth measures how close a graph is to being a tree.
- Many **NP**-hard graph problems can be solved in polynomial time or even linear time when the treewidth of the input graph is bounded.
  - the Maximum Independent Set problem
  - the Minimum Dominating Set problem
  - the Hamiltonian Cycle problem

  ⋮

Fixed-parameter *intractable*?

- $I \subseteq V$ is an independent set:
    - None of pairs of the vertices in I are adjacent.

⋆ A well-known fact:

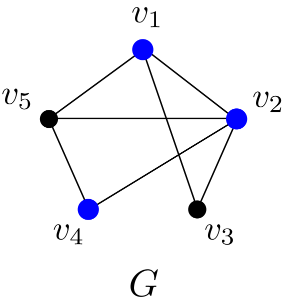
- A graph G has a vertex cover of size k
  $\Leftrightarrow$ G has an independent set of size $k' = n - k$.

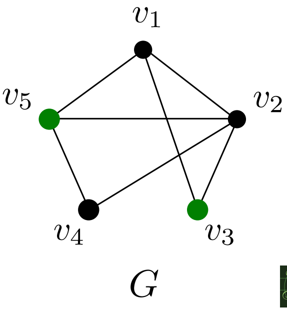
The Independent Set problem, the Clique problem, the Dominating Set problem, . . ., etc.

Fixed-parameter *intractable*?

- $I \subseteq V$ is an independent set:
  - None of pairs of the vertices in I are adjacent.

⋆ A well-known fact:
  - A graph G has a vertex cover of size k
    $\Leftrightarrow$ G has an independent set of size $k' = n - k$.





Fixed-parameter *intractable*?

- $I \subseteq V$ is an independent set:
  - None of pairs of the vertices in I are adjacent.

⋆ A well-known fact:
  - A graph G has a vertex cover of size k
    $\Leftrightarrow$ G has an independent set of size $k' = n - k$.

- $\exists$ an $O(f(k') \cdot \mathrm{poly}(n))$ algorithm $\mathcal{A}$ for the Independent Set problem
  $\Rightarrow$ $\mathcal{A}$ can be used to solve the Vertex Cover problem efficiently even when $k = n - k'$ is large.

- Fixed-parameter intractable problems:
  The Independent Set problem, the Clique problem, the Dominating Set problem, . . ., etc.

Fixed-parameter *intractable*?

- $I \subseteq V$ is an independent set:
  - None of pairs of the vertices in I are adjacent.

⋆ A well-known fact:
  - A graph G has a vertex cover of size k
    $\Leftrightarrow$ G has an independent set of size $k' = n - k$.

- $\exists$ an $O(f(k') \cdot \text{poly}(n))$ algorithm $\mathcal{A}$ for the Independent Set problem $\Rightarrow$ $\mathcal{A}$ can be used to solve the Vertex Cover problem efficiently even when $k = n - k'$ is large.

- Fixed-parameter intractable problems:

  The Independent Set problem, the Clique problem,
  the Dominating Set problem, . . ., etc.

Introduction:
Property testing

Property testing

- General notion: Rubinfeld & Sudan (*SIAM J. Comput.* 1996).
  - Graph property testing: Goldreich, Goldwasser & Ron (*J. ACM* 1998).

Task of property testing

Input: An input I and a specified property $\mathcal{P}$

Task: Fulfill the following requirements in $o(|I|)$ time.

- If I satisfies $\mathcal{P}$ $\Rightarrow$ answer "yes" with prob. $\geq \frac{2}{3}$; ($= 1$: *1-sided error*)
- If I is **far** from satisfying $\mathcal{P}$ $\Rightarrow$ answer "no" with prob. $\geq \frac{2}{3}$.

- A notion of "approximating" yes/no problems in sublinear time.
  - $f(n) = o(g(n))$ if $\lim_{n\to\infty} f(n)/g(n) = 0$.
- **Property testers**: algorithms accomplishing the above task.

Property testing (contd.)

This research field has been very active since 1996.

- Articles in this field appear in important conferences.
  - FOCS, STOC, SODA, ICALP, STACS, ESA, APPROX+RANDOM, etc.
- Surveys for this field:
  - ⋆ [Goldreich 1998], [Fischer 2001], [Ron 2001], [Alon & Shapira 2005].
- Well-known experts in this field:
  - N. Alon, A. Czumaj, L. Fortnow, O. Goldreich, D. Ron, R. Rubinfeld, A. Shapira, L. Trevisan, etc.

Property testing (ϵ-far)

- In property testing, we use ϵ-far to say that the input is far from a certain property.
- ϵ: the least fraction of the input that needs to be modified.
- For example, $L = (0, 2, 1, 4)$ over $\{0, 1, 2, 3, 4\}$ is $1/4$-far from being monotonically nondecreasing.
  - $(0, 2, 1, 4) \Rightarrow (0, 2, 3, 4)$.

Property testing (ϵ-far)

- In property testing, we use ϵ-far to say that the input is far from a certain property.
- ϵ: the least fraction of the input that needs to be modified.
- For example, $L = (0, 2, 1, 4)$ over $\{0, 1, 2, 3, 4\}$ is $1/4$-far from being monotonically nondecreasing.
  - $(0, 2, 1, 4) \Rightarrow (0, 2, 3, 4)$.

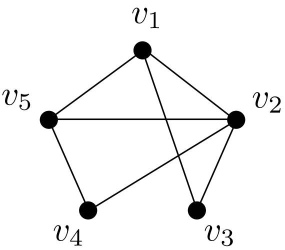
Property testing (ϵ-far)

- In property testing, we use ϵ-far to say that the input is far from a certain property.
- ϵ: the least fraction of the input that needs to be modified.
- For example, $L = (0, 2, 1, 4)$ over $\{0, 1, 2, 3, 4\}$ is $1/4$-far from being monotonically nondecreasing.
  - $(0, 2, 1, 4) \quad \Rightarrow \quad (0, 2, 3, 4)$.

Property testing (ϵ-far)

G is ?-far from having a vertex cover of size ≤ 2.

	v_1	v_2	v_3	v_4	v_5
v_1	–	1	1	0	1
v_2	–	–	1	1	1
v_3	–	–	–	0	0
v_4	–	–	–	–	1
v_5	–	–	–	–	–

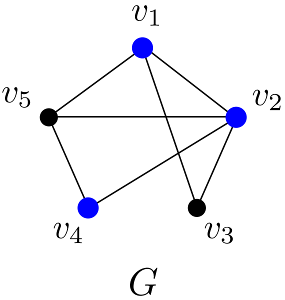


G

Property testing (ϵ-far)

G is ?-far from having a vertex cover of size ≤ 2.

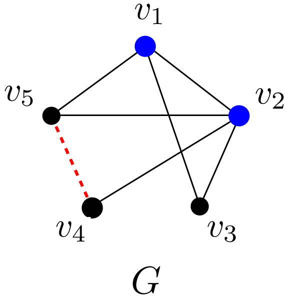
	v_1	v_2	v_3	v_4	v_5
v_1	–	1	1	0	1
v_2	–	–	1	1	1
v_3	–	–	–	0	0
v_4	–	–	–	–	1
v_5	–	–	–	–	–



Property testing (ϵ-far)

G is (1/10)-far from having a vertex cover of size ≤ 2.

	v_1	v_2	v_3	v_4	v_5
v_1	–	1	1	0	1
v_2	–	–	1	1	1
v_3	–	–	–	0	0
v_4	–	–	–	–	0
v_5	–	–	–	–	–



Property testing (contd.)

- A property is
  - testable: it admits a property tester of time complexity **independent of the input size**.
  - **easily**-testable: it admits a **1-sided-error** property tester of time complexity $O(\textbf{poly}(1/\epsilon))$.
- A property tester is non-adaptive: it makes queries to the input **without** knowing the results of previous ones.
- Time complexity $= \Omega(\#\text{queries})$.

The sparse / dense model for testing graph properties

⋆ **The sparse model**

- $f_G : V(G) \times [d] \mapsto V(G) \cup \varnothing$ (adjacency list)
  - $[d] := \{1, 2, \ldots, d\}$
  - $f_G(v, i) = u$: (u, v) is the ith edge incident to v.
  - $f_G(v, i) = \varnothing$: there is no such edge.
- ϵ-far:
  $\geq \epsilon dn$ edge insertions or removals are required.

⋆ **The dense model**

- $M_G : V(G) \times V(G) \mapsto \{0, 1\}$ (adjacency matrix)
  - $M_G(v, u) = 1$: u and v are adjacent.
- ϵ-far:
  $\geq \epsilon n^2$ edge insertions or removals are required.

Testable & easily testable results

Some graph properties that are testable, and even easily testable.

⋆ In the sparse model

- connectivity & k-connectivity (easily-testable)
- being cycle-free
- being Eulerian (easily-testable)
- minor-closed properties
- ...

⋆ In the dense model

- being H-free (easily-testable iff H is bipartite)
- being induced H-free (not easily-testable if $H \notin \{P_2, \bar{P}_2, P_3, \bar{P}_3, P_4, \bar{P}_4, C_4, \bar{C}_4\}$)
- k-colorability (easily-testable)
- Having a clique of size $\geq \rho n$ for $\rho \in (0, 1)$
- Hereditary graph properties
- ...

Non-testable results

Some graph properties that are NOT (or remain unknown) to be testable.

⋆ **In the sparse model**

- bipartiteness
- k-colorability for general $k \geq 3$
- Having a dominating set of size $\leq \rho n$ for $\rho \in (0, 1)$
- Having a vertex cover of size $\leq \rho n$ for $\rho \in (0, 1)$
- ...

⋆ **In the dense model**

- first-order graph properties with a quantifier alternation of type '$\forall\exists$'
- ...

Motivations

- Some properties cannot be tested efficiently, even non-testable.
- **FPT** problems can be efficiently solved when the parameters are small.
- We wish to know:
  - the parameters that make property testing difficult;
  - whether some properties can be tested more efficiently when some parameters are small.

Motivations: parameterized property testing

- Some properties cannot be tested efficiently, even non-testable.
- **FPT** problems can be efficiently solved when the parameters are small.
- We wish to know:
  - the parameters that make property testing difficult;
  - whether some properties can be tested more efficiently when some parameters are small.

⋆ A new concept: Parameterized property testing

  - Introducing parameters to the standard property testing.

Parameterized property testing

Parameterized property testers

Input: An input instance I, $\epsilon \in (0,1)$, and a parameter $k \in \mathbb{Z}^+$.

Property: $\mathcal{P}$.

Task: Testing if I has the property $\mathcal{P}$ in $O(f(k, 1/\epsilon) \cdot o|I|)$ time.

- $f(k, 1/\epsilon)$: a function solely depending on k and ϵ.

- $O(f(k, 1/\epsilon))$ time $\Rightarrow$ parameterized testable.

Parameterized property testing (previous work)

Positive results:

- k-colorability in the dense model:
  - $O(k^2 \ln^2 k/\epsilon^4)$ [Alon & Krivelevich 2002].
- being H-free (without having H as a subgraph; $|H| = h$) in the dense model:
  - $O(h^2(1/2\epsilon)^{h^2/4})$ [Alon 2002].
- k-connectivity in the sparse model: (weakly uniform)
  - $O(d(ck/\epsilon d)^k \log(k/\epsilon d))$ [Yoshida & Ito 2008].
    - For $n \leq \max\{120k^3, 400k^3/\epsilon d\} \Rightarrow$ Using an $O(\text{poly}(n))$ $= O(\text{poly}(k/\epsilon d))$ algorithm.

Parameterized property testing (previous work)

Positive results:

- k-colorability in the dense model:
  - $O(k^2 \ln^2 k/\epsilon^4)$ [Alon & Krivelevich 2002].
- being H-free (without having H as a subgraph; $|H| = h$) in the dense model:
  - $O(h^2(1/2\epsilon)^{h^2/4})$ [Alon 2002].
- k-connectivity in the sparse model: **(weakly uniform)**
  - $O(d(ck/\epsilon d)^k \log(k/\epsilon d))$ [Yoshida & Ito 2008].
    - For $n \leq \max\{120k^3, 400k^3/\epsilon d\} \Rightarrow$ Using an $O(\text{poly}(n))$ $= O(\text{poly}(k/\epsilon d))$ algorithm.

Parameterized property testing (previous work)

A negative result:

- k-colorability for $k \geq 3$ in the *sparse* model:
  - $\Omega(n)$ [Bogdanov, Obata & Trevisan 2002].

Parameterized property testing (trivial to test for small k's)

Properties are trivial to test when k is small (in the sparse model):

- Having a simple k-path / k-cycle ($\in$ **FPT**).
  - ⋆ $O(k)$ edge insertions for any graph. (e.g., $k < \epsilon dn$)
  - Answer "yes" for any input graph.
- Having a dominating set of size $\leq k$ ($\notin$ **FPT**).
  - ⋆ k vertices have $\leq k \cdot d$ neighbors.
  - Answer "no" for any input graph.
- Having a clique / an independent set of size $\leq k$ ($\notin$ **FPT**).
  - ⋆ $O(k^2)$ edge insertions for any graph.
  - Answer "yes" for any input graph.

When k is large, just run the exact algorithms. $\Rightarrow$ **weakly uniform**

Parameterized property testing (trivial to test for small k's)

Properties are trivial to test when k is small (in the sparse model):

- Having a simple k-path / k-cycle ($\in$ **FPT**).
  - ⋆ $O(k)$ edge insertions for any graph. (e.g., $k < \epsilon dn$)
  - Answer "yes" for any input graph.
- Having a dominating set of size $\leq k$ ($\notin$ **FPT**).
  - ⋆ k vertices have $\leq k \cdot d$ neighbors.
  - Answer "no" for any input graph.
- Having a clique / an independent set of size $\leq k$ ($\notin$ **FPT**).
  - ⋆ $O(k^2)$ edge insertions for any graph.
  - Answer "yes" for any input graph.

When k is large, just run the exact algorithms. $\Rightarrow$ **weakly uniform**

Parameterized property testing (trivial to test for small k's)

Properties are trivial to test when k is small (in the sparse model):

- Having a simple k-path / k-cycle ($\in$ **FPT**).
  - ⋆ $O(k)$ edge insertions for any graph. (e.g., $k < \epsilon dn$)
  - Answer "yes" for any input graph.
- Having a dominating set of size $\leq k$ ($\notin$ **FPT**).
  - ⋆ k vertices have $\leq k \cdot d$ neighbors.
  - Answer "no" for any input graph.
- Having a clique / an independent set of size $\leq k$ ($\notin$ **FPT**).
  - ⋆ $O(k^2)$ edge insertions for any graph.
  - Answer "yes" for any input graph.

When k is large, just run the exact algorithms. $\Rightarrow$ **weakly uniform**

Tackle a problem in computational biology through three aspects

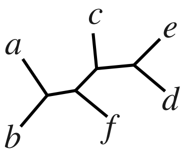
- In this dissertation, we study a problem related to consistency of quartet topologies.
    - It's a problem originated from computational biology,
        - **evolutionary tree reconstruction**.

⋆ We tackle the problems through the following algorithmic aspects:

- fixed-parameter algorithms
- property testing
- parameterized property testing

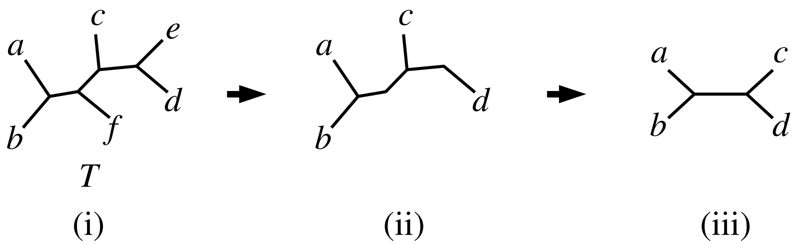
Evolutionary trees

- S: a set of taxa; $|S| = n$.
- An evolutionary tree T on S:
  - An *unrooted*, *leaf-labeled* tree;
  - The leaves are bijectively labeled by the taxa in S;
  - Each internal node has degree *three*.

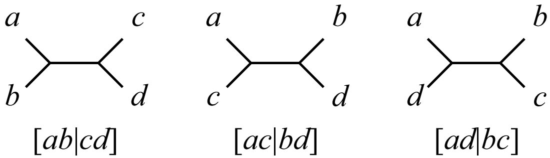


T

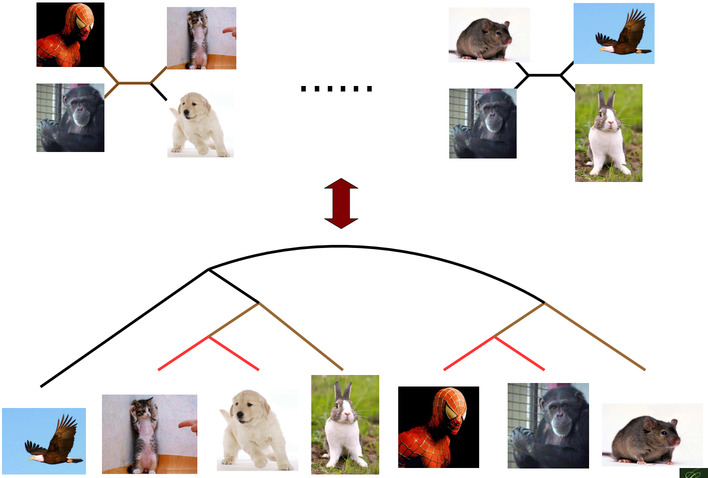
Evolutionary trees & quartets (contd.)



Evolutionary trees & quartets (contd.)

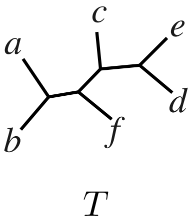


Evolutionary trees & quartets (contd.)



A set Q of quartet topologies over $\{a, b, c, d, e, f\}$:

$$[ac \mid bd], \quad [ab \mid ce], \quad [ab \mid cf],$$
$$[ab \mid de], \quad [ab \mid df], \quad [ab \mid ef],$$
$$[ac \mid de], \quad [af \mid cd], \quad [af \mid ce],$$
$$[af \mid de], \quad [bd \mid ce], \quad [bf \mid cd],$$
$$[bf \mid ce], \quad [bf \mid de], \quad [cf \mid de].$$

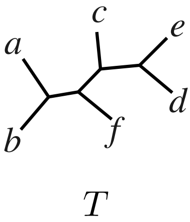


Quartet errors w.r.t. T.

- Determine if Q has 0 quartet error: **NP**-complete [Steel 1992].

A set Q of quartet topologies over $\{a, b, c, d, e, f\}$:

$[ac \mid bd]$,	$[ab \mid ce]$,	$[ab \mid cf]$,
$[ab \mid de]$,	$[ab \mid df]$,	$[ab \mid ef]$,
$[ac \mid de]$,	$[af \mid cd]$,	$[af \mid ce]$,
$[af \mid de]$,	$[bd \mid ce]$,	$[bf \mid cd]$,
$[bf \mid ce]$,	$[bf \mid de]$,	$[cf \mid de]$.

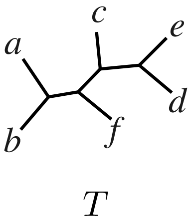


Quartet errors w.r.t. T.

- Determine if Q has 0 quartet error: **NP**-complete [Steel 1992].

A set Q of quartet topologies over $\{a, b, c, d, e, f\}$:

$[ac \mid bd]$,	$[ab \mid ce]$,	$[ab \mid cf]$,
$[ab \mid de]$,	$[ab \mid df]$,	$[ab \mid ef]$,
$[ac \mid de]$,	$[af \mid cd]$,	$[af \mid ce]$,
$[af \mid de]$,	$[bd \mid ce]$,	$[bf \mid cd]$,
$[bf \mid ce]$,	$[bf \mid de]$,	$[cf \mid de]$.

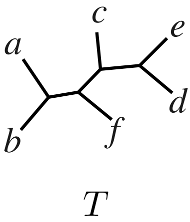


Quartet errors w.r.t. T.

- Determine if Q has 0 quartet error: **NP**-complete [Steel 1992].

A complete set Q of quartet topologies over $\{a, b, c, d, e, f\}$:

$[ac \mid bd]$, $[ab \mid ce]$, $[ab \mid cf]$,
$[ab \mid de]$, $[ab \mid df]$, $[ab \mid ef]$,
$[ac \mid de]$, $[af \mid cd]$, $[af \mid ce]$,
$[af \mid de]$, $[bd \mid ce]$, $[bf \mid cd]$,
$[bf \mid ce]$, $[bf \mid de]$, $[cf \mid de]$.

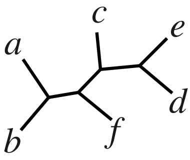


Quartet errors w.r.t. T.

- Determine if Q has 0 quartet error: $O(n^4)$ [Erdős *et al.* 1999].
- Compute # quartet errors of Q: **NP**-hard [Berry *et al.* 1999];
  - Approximation ratio: $O(n^2)$ [Jiang *et al.* 2000].

A complete set Q of quartet topologies over $\{a, b, c, d, e, f\}$:

$[ac \mid bd]$, $[ab \mid ce]$, $[ab \mid cf]$,
$[ab \mid de]$, $[ab \mid df]$, $[ab \mid ef]$,
$[ac \mid de]$, $[af \mid cd]$, $[af \mid ce]$,
$[af \mid de]$, $[bd \mid ce]$, $[bf \mid cd]$,
$[bf \mid ce]$, $[bf \mid de]$, $[cf \mid de]$.

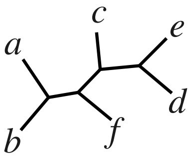


Quartet errors w.r.t. T.

- Determine if Q has 0 quartet error: $O(n^4)$ [Erdős *et al.* 1999].
- Compute # quartet errors of Q: **NP**-hard [Berry *et al.* 1999];
  - Approximation ratio: $O(n^2)$ [Jiang *et al.* 2000].

A complete set Q of quartet topologies over $\{a, b, c, d, e, f\}$:

$[ac \mid bd]$,	$[ab \mid ce]$,	$[ab \mid cf]$,
$[ab \mid de]$,	$[ab \mid df]$,	$[ab \mid ef]$,
$[ac \mid de]$,	$[af \mid cd]$,	$[af \mid ce]$,
$[af \mid de]$,	$[bd \mid ce]$,	$[bf \mid cd]$,
$[bf \mid ce]$,	$[bf \mid de]$,	$[cf \mid de]$.



T

Quartet errors w.r.t. T.

- Determine if Q has 0 quartet error: $O(n^4)$ [Erdős *et al.* 1999].
- Compute # quartet errors of Q: **NP**-hard [Berry *et al.* 1999];
  - Approximation ratio: $O(n^2)$ [Jiang *et al.* 2000].

Contribution: Fixed-parameter algorithms for MQI

- MQI: minimum quartet inconsistency

The parameterized MQI problem

Input: A complete set Q of quartet topologies, $k \in \mathbb{Z}^+$.

Task: Determine if Q has $\leq k$ quartet errors.

- Previous results:
  - $O(4^k n + n^4)$ [Gramm & Niedermeier 2003].
- Our results:
  - $O(3.0446^k n + n^4)$, $O(2.0162^k n^3 + n^5)$, $O^*((1+\varepsilon)^k)$.
    - $O^*((1+\varepsilon)^k)$: $\varepsilon \downarrow$, the polynomial factor $\uparrow$.
    - Chang, Lin, & Rossmanith: *Theory Comput. Syst.*, 2010.

Contribution: Fixed-parameter algorithms for MQI

- MQI: minimum quartet inconsistency

The parameterized MQI problem

Input: A complete set Q of quartet topologies, $k \in \mathbb{Z}^+$.

Task: Determine if Q has $\leq k$ quartet errors.

- Previous results:
  - ⋆ $O(4^k n + n^4)$ [Gramm & Niedermeier 2003].
- Our results:
  - $O(3.0446^k n + n^4)$, $O(2.0162^k n^3 + n^5)$, $O^*((1+\varepsilon)^k)$.
    - $O^*((1+\varepsilon)^k)$: $\varepsilon \downarrow$, the polynomial factor $\uparrow$.
    - ⋆ Chang, Lin, & Rossmanith: *Theory Comput. Syst.*, 2010.

Contribution: Fixed-parameter algorithms for MQI

- MQI: minimum quartet inconsistency

The parameterized MQI problem

Input: A complete set Q of quartet topologies, $k \in \mathbb{Z}^+$.

Task: Determine if Q has $\leq k$ quartet errors.

- Previous results:
  - ⋆ $O(4^k n + n^4)$ [Gramm & Niedermeier 2003].
- Our results:
  - $O(3.0446^k n + n^4)$, $O(2.0162^k n^3 + n^5)$, $O^*((1+\varepsilon)^k)$.
    - $O^*((1+\varepsilon)^k)$: $\varepsilon \downarrow$, the polynomial factor $\uparrow$.
    - ⋆ Chang, Lin, & Rossmanith: *Theory Comput. Syst.*, 2010.

Contribution: Testing tree-consistency of quartet topologies

The property: tree-consistent

- Q is tree-consistent: 0 quartet errors.

Testing tree-consistency of a complete Q

Input: A **complete** set Q of quartet topologies, $0 < \epsilon < 1$.

Task: Testing if Q is tree-consistent.

- Determine if a complete Q is tree-consistent: $O(n^4)$ [Erdős *et al.* 1999].

Our result:

- An $O(n^3/\epsilon)$ property tester (1-sided error & non-adaptive).
  - ⋆ Chang, Lin, & Rossmanith: *Theory Comput. Syst.*. Accepted. Online first.

Contribution: Testing tree-consistency of quartet topologies

The property: tree-consistent

- Q is tree-consistent: 0 quartet errors.

Testing tree-consistency of a complete Q

Input: A **complete** set Q of quartet topologies, $0 < \epsilon < 1$.

Task: Testing if Q is tree-consistent.

- Determine if a complete Q is tree-consistent: $O(n^4)$ [Erdős *et al.* 1999].

  Our result:

  - An $O(n^3/\epsilon)$ property tester (1-sided error & non-adaptive).
    - ⋆ Chang, Lin, & Rossmanith: *Theory Comput. Syst.*. Accepted. Online first.

Contribution: Parameterized property testing for tree-consistency

Missing quartets: quartets whose topologies are *missing*.

- T_{miss}: a set of k missing quartets.

Testing tree-consistency with k missing quartets

Input: A set Q of quartet topologies, a set T_{miss} of k missing quartets, $0 < \epsilon < 1$.

Task: Testing if Q is tree-consistent.

- Determine if Q is tree-consistent: **NP**-complete [Steel 1992].

Our results:

- Indicate: deterministically solvable in $O(3^k n^4)$ time.
- An $O(k3^k n^3/\epsilon)$ property tester (1-sided error, non-adaptive & uniform)
  - ⋆ Lin: *Proc. Workshop on Combin. Math. Comput. Theory*, 2011.

Contribution: Parameterized property testing for tree-consistency

Missing quartets: quartets whose topologies are *missing*.

- T_{miss}: a set of k missing quartets.

Testing tree-consistency with k missing quartets

Input: A set Q of quartet topologies, a set T_{miss} of k missing quartets, $0 < \epsilon < 1$.

Task: Testing if Q is tree-consistent.

- Determine if Q is tree-consistent: **NP**-complete [Steel 1992].

  Our results:

  - Indicate: deterministically solvable in $O(3^k n^4)$ time.
  - An $O(k3^k n^3/\epsilon)$ property tester (1-sided error, non-adaptive & uniform)
    - ⋆ Lin: *Proc. Workshop on Combin. Math. Comput. Theory*, 2011.

Summary of our contributions in quartet consistency

Property (Problem)	PC	PT	PPT
MQI	$O(3.0446^k n + n^4)$ $O(2.0162^k n^3 + n^5)$ $O^*((1+\varepsilon)^k)$	–	–
$\mathcal{P}_{tree}$	–	$O(n^3/\epsilon)$	$O(k3^k n^3/\epsilon)$

PC: parameterized complexity; PT: property testing; PPT: parameterized property testing.
MQI: minimum quartet inconsistency; $\mathcal{P}_{tree}$: tree-consistency.

Two more results on
parameterized property testing

Parameterized property testing for two graph properties

For the following two problems in **NP**-hard $\cap$ **FPT**:

- the Vertex Cover problem and
- the Treewidth problem,

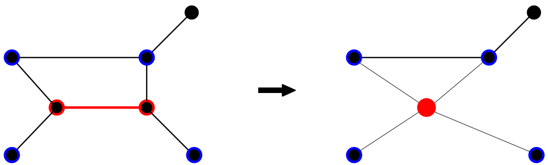
we propose parameterized property testers for their corresponding properties.

Parameterized property testers (Vertex Cover)

- The property: Having a vertex cover of size $\leq k$.
  - Denoted it by $\mathcal{P}_{VC\leq k}$.
- Previous results:
  - **FPT**: $O(1.2738^k + kn)$ [Chen *et al.* 2010]
  - **Property testing**:
    - $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\Big\}O(\text{poly}(1/\epsilon))\text{ 2's}$ (dense model) [Alon & Shapira 2008]
    - $\Omega(\sqrt{n})$ for $\mathcal{P}_{VC\leq \rho n}$ (sparse model) [Goldreich & Ron 2002]
- Our result: (in the sparse model)
  - ⋆ $O(1.2738^k + k^2/\epsilon + kd/\epsilon)$ (1-sided error, adaptive & weakly uniform).
  - – Joint work with M.-S. Chang, L.-J. Hung, A. Langer & P. Rossmanith.

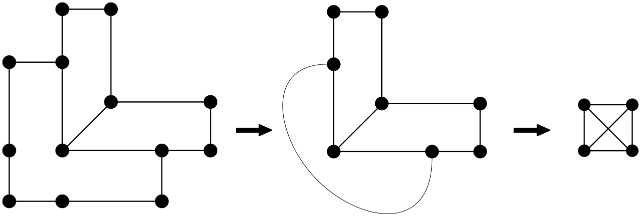
Parameterized property testers (Treewidth)

- Edge contractions.



Parameterized property testers (Treewidth)

- H is a minor of G:
  - H can be obtained from G by edge removals, vertex removals and edge *contractions*.



(a) (b) (c)

Parameterized property testers (Treewidth)

- Minor-closed properties: closed under taking minors.
- G satisfies a minor-closed property $\mathcal{P}$
  $\Leftrightarrow \exists$ a finite $\mathcal{F}$ s.t. H is not a minor of G for each $H \in \mathcal{F}$.
  [Robertson & Seymour 1983–2004]
- "Having treewidth $\leq k$" is minor-closed [Kloks 1994].
  - Yet the obstruction set $\mathcal{F}$ is not explicitly known for $k > 3$.

Parameterized property testers (Treewidth)

- The property: Having treewidth $\leq k$.
  - Denoted it by $\mathcal{P}_{tw \leq k}$.
- Previous results:
  - **FPT**: $2^{\Theta(k^3)} \cdot k^{O(1)} \cdot n$ [Bodlaender 1996]
  - **Property testing**: $2^{\text{poly}(1/\epsilon)}$ (sparse model) [Hassidim *et al.* 2009]
    - for minor-closed properties
- Our results: (in the sparse model)
  - $2^{d^{O(kd^3/\epsilon^2)}}$
  - $d^{(k/\epsilon)^{O(k^2)}} + 2^{\text{poly}(k,d,1/\epsilon)}$.
    - ⋆ Both are **uniform** w.r.t. k, d, ϵ; (2-sided error, adaptive).
    - ⋆ **Without** using the obstruction set of forbidden minors.
    - – Joint work with M.-S. Chang, L.-J. Hung, A. Langer & P. Rossmanith.

A summary of our contributions

Summary of our contributions

Property (Problem)	PC	PT	PPT
MQI	$O(3.0446^k n + n^4)$ $O(2.0162^k n^3 + n^5)$ $O^*((1+\varepsilon)^k)$	–	–
$\mathcal{P}_{tree}$	–	$O(n^3/\epsilon)$	$O(k3^k n^3/\epsilon)$
$\mathcal{P}_{VC \leq k}$	$O(1.2738^k + kn)$	$\left.2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\right\\}O(\text{poly}(1/\epsilon))\text{ 2's}$ †	$O(1.2738^k + k^2/\epsilon + kd/\epsilon)$ ‡
$\mathcal{P}_{VC \leq \rho \cdot n}$	–	$\Omega(\sqrt{n})$ ‡	–
$\mathcal{P}_{tw \leq k}$	$2^{\Theta(k^3)} \cdot k^{O(1)} \cdot n$	$O(2^{\text{poly}(1/\epsilon)})$ ‡	$2^{d^{O(kd^3/\epsilon^2)}}$ ‡ $d^{(k/\epsilon)^{O(k^2)}} + 2^{\text{poly}(k,d,1/\epsilon)}$ ‡

PC: parameterized complexity; PT: property testing; PPT: parameterized property testing.
MQI: minimum quartet inconsistency; $\mathcal{P}_{tree}$: tree-consistency; $\mathcal{P}_{VC \leq k}$: having a vertex cover of size $\leq k$; $\mathcal{P}_{tw}$: having treewidth $\leq k$; ‡: sparse model; †: dense model.

Summary of our contributions (Testers)

Property	Sublinear	Testable (easily)	Non-adaptive	1/2-sided error	uniform
$\mathcal{P}_{tree}$	Yes	? (?)	Yes	1	Yes
$\mathcal{P}_{VC\leq k}$	Yes	Yes (no)	No	1	weakly
$\mathcal{P}_{tw\leq k}$	Yes	Yes (?)	No	2	Yes

⋆ Parameterized easily testable: 1-sided error, uniform & $O(\text{poly}(k, d, 1/\epsilon))$ time.

Thanks for your attention.