Enumerating Unlabeled Cactus Graphs

Maryam Bahrani

Under the Direction of Dr. Jérémie Lumbroso
Princeton University Department of Computer Science

Dec 13, 2016
Trees
Trees

binary tree
Trees

binary tree

left-leaning red-black tree*
Trees

binary tree

trie

left-leaning red-black tree

* Images reprinted with permission from Robert Sedgewick and Kevin Wayne (Algorithms lecture sides)
Trees

binary tree

trie*

binomial heap

left-leaning red-black tree*
Symbolic Method on Trees

A binary tree is
— either a leaf
— or an internal node, and a left subtree, and a right subtree
A binary tree is
— either a leaf $\square$
— or an internal node,
    and a left subtree,
    and a right subtree

$\mathcal{T} = \square \cup (\mathcal{T} \bullet \mathcal{T})$

symbolic specification
Symbolic Method on Trees

A binary tree is
— either a leaf □
— or an internal node, and a left subtree, and a right subtree

\[ \mathcal{T} = \square \cup (\mathcal{T} \cdot \mathcal{T}) \]

symbolic specification

\[ T(z) = 1 + T(z) \times z \times T(z) \]
generating function
Symbolic Method on Trees

A binary tree is
- either a leaf □
- or an internal node, and a left subtree, and a right subtree

\[ T = □ \cup (T \cdot T) \]

\[ T(z) = 1 + T(z) \times z \times T(z) \]

1, 1, 2, 5, 14, 42, 132, 429, 1430, ...
Symbolic Method on Trees

A binary tree is
— either a leaf □
— or an internal node, and a left subtree, and a right subtree

\[ \mathcal{T} = \bigcirc \cup (\mathcal{T} \cdot \mathcal{T}) \]

\[ T(z) = 1 + T(z) \times z \times T(z) \]

1, 1, 2, 5, 14, 42, 132, 429, 1430, \ldots

Number of binary trees with 5 internal nodes
Symbolic Method on Trees

A binary tree is
— either a leaf
— or an internal node,
and a left subtree,
and a right subtree

\[ T = \mathbb{D} \cup (T \times T) \]

\[ T(z) = 1 + T(z) \times z \times T(z) \]

1, 1, 2, 5, 14, 42, 132, 429, 1430, …
Decomposing General Graphs
Generalizing Trees

Directed Acyclic Graphs (DAGs)

k-trees

block graphs (clique trees)

cactus graphs (cacti)
Generalizing Trees

Directed Acyclic Graphs (DAGs)

k-trees

Focus of this talk

cactus graphs (cacti)

block graphs (clique trees)
Cactus Graphs

A graph is a **cactus** iff every edge is part of *at most* one cycle.
A graph is a cactus iff every edge is part of at most one cycle.
A graph is a **cactus** iff every edge is part of *at most* one cycle.
A graph is a cactus iff every edge is part of at most one cycle.
Prior Work

Did you mean: cactus graphs

A linear-time algorithm for solving the center problem on weighted cactus graphs
For a nontrivial graph G (V, E), the distance d(u, v) between vertices u and v is the length of a shortest path p(u, v) in G if such a path exists. The eccentricity e(u) of a vertex u in a graph is the distance from u to a vertex furthest from u. That is, e(u) = max{d(u, v) : v ∈ V}. The
Cited by 28 Related articles All 6 versions Web of Science: 10 Cite Save More

The onnoxious center problem on weighted cactus graphs
S Zmazek, J Žerovnik - Discrete Applied Mathematics, 2004 - Elsevier
The onnoxious center problem in a graph G asks for a location on an edge of the graph such that the minimum weighted distance from this point to a vertex of the graph is as large as possible. An algorithm is given which finds the onnoxious center on a weighted cactus graph
Cited by 32 Related articles All 6 versions Web of Science: 14 Cite Save More

Cactus graphs for genome comparisons
B Paten, M DiEckhards, D Earl, Jl Jotham... - Journal of... 2011 - online.liebertpub.com
Abstract We introduce a data structure, analysis, and visualization scheme called a cactus graph for comparing sets of related genomes. In common with multi-break point graphs and A-Bruijn graphs, cactus graphs can represent duplications and general genomic
Cited by 40 Related articles All 4 versions Web of Science: 19 Cite Save More

Computing the weighted Wiener and Szeged number on weighted cactus graphs in linear time
S Zmazek, J Žerovnik - Croatian Chemical Acta, 2003 - hrcak.srce.hr
Szabatok Cactus is a graph in which every edge lies on at most one cycle. Linear algorithms for computing the weighted Wiener and Szeged numbers on weighted cactus graphs are given. Graphs with weighted vertices and edges correspond to molecular graphs with
Cited by 29 Related articles All 3 versions Web of Science: 14 Cite Save More

The ratio of the irredundance number and the domination number for block-cactus graphs
V Zverovich - Journal of Graph Theory, 1998 - eprints.uwe.ac.uk
Prior Work

Cactus graphs for genome comparisons

A linear-time algorithm for solving the center problem on weighted cactus graphs

A linear-time algorithm for solving the center problem on weighted cactus graphs

A linear-time algorithm for solving the center problem on weighted cactus graphs

Computing the weighted Wiener and Szeged number on weighted cactus graphs in linear time
B. Zmazek, J. Žerovnik - Croatica Chemica Acta, 2003 - croatica-chemica.hr

The ratio of the irredundance number and the domination number for block-cactus graphs
V. Zverovich - Journal of Graph Theory, 1998 - ejc.wiley.com

CENTDIAN COMPUTATION IN CACTUS GRAPHS
Boaz Ben-Moshe
Efficient Algorithms for the Weighted 2-Center Problem in a Cactus Graph
Nasreen Khan¹, Madhumangal Pal¹, Anita Pal²
¹Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore, India
²Department of Mathematics, National Institute of Technology, Durgapur, India
Email: (mmpalv, afaaruulidmphan, anita.buie)@gmail.com

L(0,1)-LABELLING OF CACTUS GRAPHS

A linear-time algorithm for solving the center problem on weighted cactus graphs

Mustapha Chellali
BOUNDS ON THE 2-DOMINATION NUMBER IN CACTUS GRAPHS
Murat Arcak, Senior Member IEEE

Diagonal Stability on Cactus Graphs and Application to Network Stability Analysis

Nasreen Khan¹, Anita Pal² and Madhumangal Pal†
Cactus Graphs for Genome Comparisons
Benedict Paten¹, Mark Diekhans¹, Dent Earl¹, John St. John¹, Jian Ma³

A CHARACTERIZATION OF WELL-COVERED BLOCK-CACTUS GRAPHS

Minko Markov, Mihail Manev, Kazuhiro Masuda, Tatsuya Takahashi, Junji Tanaka

RECENT DEVELOPMENTS IN TREE-PRUNING METHODS AND POLYNOMIALS FOR CACTUS GRAPHS AND TREES
K. Balasubramanian*
Department of Chemistry, Arizona State University, Tempe, AZ 85287-1604, USA
Prior Work

On the Number of Husimi Trees
Harary and Uhlenbeck (1952):

— derived functional equations for **non-plane, mixed, unlabeled** cacti.

— promised to provide “a more systematic treatment of the general case of pure k-cacti” in a subsequent paper (it appears they never published such a paper)

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

If $c_1^*, \ldots, c_4^*$ denote again the number of dissimilar quadrilaterals of these symmetry types occurring in a given tree, then one has

$$
\begin{align*}
L^* &= 4c_4^* + 2c_2^* + c_1^* + 3c_4^* + c_1^* \\
C^* &= c_1^* + c_2^* + c_3^* + c_4^* \\
C^A &= c_2^* + c_3^* + c_4^* \\
C^B &= c_1^* + c_4^*. \\
\end{align*}
$$

Thus one gets from (6), since $a$ is of course again zero:

$$1 = p^* - 3c_1^* - 2c_2^* - c_3^* - c_4^*. \quad (6c)$$
Prior Work

On the Number of Husimi Trees
Harary and Uhlenbeck (1952):

— derived functional equations for non-plane, mixed, unlabeled cacti.

— promised to provide “a more systematic treatment of the general case of pure k-cacti” in a subsequent paper (it appears they never published such a paper)

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

If \( c_1^*, \ldots, c_6^* \) denote again the number of dissimilar quadrilaterals of these symmetry types occurring in a given tree, then one has

\[
\begin{align*}
I^* &= 4c_1^* + 2c_2^* + c_3^* + 3c_4^* + c_5^* + c_6^* \\
C_e^* &= c_1^* + c_2^* + c_3^* + c_4^* + c_5^* + c_6^* \\
C_e^p &= c_4^* + c_5^* + c_6^* \\
C_{eu} &= c_1^* + c_6^*.
\end{align*}
\]

Thus one gets from (6), since \( \rho^* \) is of course again zero:

\[
1 = \rho^* - 3c_1^* - 2c_2^* - c_3^* - c_4^* - c_5^* - c_6^*.
\]
Prior Work

On the Number of Husimi Trees
Harary and Uhlenbeck (1952):

— derived functional equations for non-plane, mixed, unlabeled cacti.

— promised to provide “a more systematic treatment of the general case of pure k-cacti” in a subsequent paper (it appears they never published such a paper)

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

\[ P^* = 4c_1^* + 2c_2^* + c_3^* + 3c_4^* + c_5^* \]

\[ c_1^* = c_1^* + c_2^* + c_3^* + c_4^* + c_5^* \]

\[ c_2^* = c_1^* + c_2^* \]

\[ c_3^* = c_1^* + c_2^* \]

\[ c_4^* = c_1^* + c_2^* \]

\[ c_5^* = c_1^* + c_2^* \]

Thus one gets from (6), since \( a \) is of course again zero:

\[ 1 = p^* - 3c_1^* - 2c_2^* - c_3^* - c_4^* - c_5^* \] (6c)

Enumeration of m-ary cacti
Miklós Bóna et al. (1999):

— enumerated pure, plane, unlabeled cacti.

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

\[ P^* = 4c_1^* + 2c_2^* + c_3^* + 3c_4^* + c_5^* \]

\[ c_1^* = c_1^* + c_2^* + c_3^* + c_4^* + c_5^* \]

\[ c_2^* = c_1^* + c_2^* \]

\[ c_3^* = c_1^* + c_2^* \]

\[ c_4^* = c_1^* + c_2^* \]

\[ c_5^* = c_1^* + c_2^* \]

Thus one gets from (6), since \( a \) is of course again zero:

\[ 1 = p^* - 3c_1^* - 2c_2^* - c_3^* - c_4^* - c_5^* \] (6c)
Prior Work

On the Number of Husimi Trees
Harary and Uhlenbeck (1952):

— derived functional equations for non-plane, mixed, unlabeled cacti.

— promised to provide “a more systematic treatment of the general case of pure k-cacti” in a subsequent paper (it appears they never published such a paper)

Enumeration of m-ary cacti
Miklós Bóna et al. (1999):

— enumerated pure, plane, unlabeled cacti.

only plane cacti complicated methods

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

\[
\begin{align*}
F^s &= 4c_1^s + 2c_2^s + c_3^s + 3c_4^s + c_5^s, \\
F^e &= c_1^e + c_2^e + c_3^e + c_4^e + c_5^e, \\
F^u &= c_1^u + c_2^u + c_3^u + c_4^u + c_5^u.
\end{align*}
\]

Thus one gets from (6), since \( a \) is of course again zero:

\[
1 = p^s - 3c_2^s - 2c_4^s - c_5^s - c_4^s.
\]
New Result

Exact enumeration of unlabeled, non-plane, pure $n$-cacti.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, ...</td>
</tr>
<tr>
<td>4</td>
<td>0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, ...</td>
</tr>
<tr>
<td>5</td>
<td>0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, ...</td>
</tr>
<tr>
<td>6</td>
<td>0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 4, 0, 0, 0, 0, 0, 13, 0, 0, 0, 67, ...</td>
</tr>
</tbody>
</table>
New Result

Exact enumeration of unlabeled, non-plane, pure $n$-cacti.

<table>
<thead>
<tr>
<th>$n$</th>
<th>0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 4$</td>
<td>0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, ...</td>
</tr>
<tr>
<td>$n = 5$</td>
<td>0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, ...</td>
</tr>
<tr>
<td>$n = 6$</td>
<td>0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 4, 0, 0, 0, 0, 0, 13, 0, 0, 0, 0, 67, ...</td>
</tr>
</tbody>
</table>

Number of pure 3-cacti with 9 vertices
New Result

Exact enumeration of unlabeled, non-plane, pure $n$-cacti.

| $n = 3$ | 0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, … |
| $n = 4$ | 0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, … |
| $n = 5$ | 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, … |
| $n = 6$ | 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 13, 0, 0, 0, 0, 67, … |

The first non-zero term is always 1 (corresponding to polygon)
# New Result

Exact enumeration of **unlabeled, non-plane, pure** $n$-cacti.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, ...</td>
</tr>
<tr>
<td>4</td>
<td>0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, ...</td>
</tr>
<tr>
<td>5</td>
<td>0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, ...</td>
</tr>
<tr>
<td>6</td>
<td>0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 13, 0, 0, 0, 0, 67, ...</td>
</tr>
</tbody>
</table>

Our approach is **simpler and more general** than Bóna et al.:

- can easily be extended to derive their result (**plane** cacti)
  
  e.g. plane 5-cacti: 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 17, 0, 0, 0, 102, ...  

- easily extendable to **mixed** cacti
Tree Decomposition of Graphs

\[ G = \mathcal{Z} \times (\mathcal{P} + \mathcal{S}_C) \]

\[ \mathcal{P} = \text{SEQ}_{=4} (\mathcal{Z} + \mathcal{S}_X) \]

\[ \mathcal{S}_X = \mathcal{Z} \times \text{SEQ}_{\geq 1} (\mathcal{P}) \]

\[ \mathcal{S}_C = \text{CYC}_{\geq 2} (\mathcal{P}) \]

split decomposition

symbolic specification

computer algebra system (CAS)

0, 0, 1, 0, 1, 0, 2, 0,
4, 0, 8, 0, 19, 0, 48,
0, 126, 0, 355, 0,
1037, …
Tree Decomposition of Graphs

\[ G = \mathcal{Z} \times (\mathcal{P} + S_C) \]
\[ \mathcal{P} = \text{SEQ}_{=4}(\mathcal{Z} + S_X) \]
\[ S_X = \mathcal{Z} \times \text{SEQ}_{\geq 1}(\mathcal{P}) \]
\[ S_C = \text{CYC}_{\geq 2}(\mathcal{P}) \]
The Split Decomposition

Gives a *graph-labeled tree* representation of a graph via a series of *split* operations.

— Can read adjacencies from *alternated paths*.
The Split Decomposition

Gives a graph-labeled tree representation of a graph via a series of split operations
— Can read adjacencies from alternated paths.

Decomposition base cases:

degenerate nodes:

prime nodes:
  e.g.

clique $\mathcal{K}$

star $\mathcal{S}$

cycle $\mathcal{P}$
Subtleties

Where do we start decomposing from?
— unlabeled structures have \textit{symmetries}
— different set of symmetries for different starting points ("roots")
Subtleties

Where do we start decomposing from?
— unlabeled structures have symmetries
— different set of symmetries for different starting points (“roots”)

Dissymmetry theorem (from species theory):
— allows us to correct for symmetries of trees

Cycle-pointing (based on Pólya theory):
— allows us to correct for symmetries of graphs
Verifying the enumeration:
— proof of the characterization

**Theorem 10** (split-decomposition tree characterization of 3-cacti). A graph $G$ with the reduced split-decomposition tree $(T, \mathcal{F})$ is a triangular cactus graph if and only if

- (a) $T$ is a clique-star tree;
- (b) the centers of all star-nodes are attached to leaves;
- (c) the extremities of star-nodes are only attached to clique-nodes;
- (d) every clique-node has degree 3.

**Proof.** By Lemma 12, we know that 3-cacti are exactly as the class of block graphs with induced $K_{\geq 4}$. 
Verification

Verifying the enumeration:
— proof of the characterization

---

**Theorem 10** (split-decomposition tree characterization of 3-cacti). A graph $G$ with the reduced split-decomposition tree $(T, \mathcal{F})$ is a triangular cactus graph if and only if
(a) $T$ is a clique-star tree;
(b) the centers of all star-nodes are attached to leaves;
(c) the extremities of star-nodes are only attached to clique-nodes;
(d) every clique-node has degree 3.

**Proof.** By Lemma 12, we know that 3-cacti are exactly as the class of block graphs with induced $K_{\geq 4}$.
Verification

Verifying the enumeration:
— proof of the characterization

**Theorem 10** (split-decomposition tree characterization of 3-cacti). A graph $G$ with the reduced split-decomposition tree $(T, \mathcal{F})$ is a triangular cactus graph if and only if
(a) $T$ is a clique-star tree;
(b) the centers of all star-nodes are attached to leaves;
(c) the extremities of star-nodes are only attached to clique-nodes;
(d) every clique-node has degree 3.

Proof. By Lemma 12, we know that 3-cacti have clique-width exactly as the class of block graphs with $K_{3,3}$-induced $K_{3,3}$.

— manual generation of small instances

— brute force generation of small instances

```python
class FourCactusGenerator(VertexIncrementalGenerator):
    def __init__(self, size):
        initial = _nx.complete_graph(1)
        self._operations = [WL_C4]
        super(FourCactusGenerator, self).__init__(size = size, initial = initial)

class FiveCactusGenerator(VertexIncrementalGenerator):
    def __init__(self, size):
        initial = _nx.complete_graph(1)
        self._operations = [WL_C5]
        super(FiveCactusGenerator, self).__init__(size = size, initial = initial)

class SixCactusGenerator(VertexIncrementalGenerator):
    def __init__(self, size):
        initial = _nx.complete_graph(1)
        self._operations = [WL_C6]
        super(SixCactusGenerator, self).__init__(size = size, initial = initial)
```
Conclusion

Summary
— Derived an exact enumeration for cactus graphs (previously unknown)
— Further extended the split decomposition introduced by Chauve et al. (2014), and extended by Bahrani and Lumbroso (2016)
— For the first time studied a graph class with a split decomposition tree that contains prime nodes

Next Steps
— Parameter analysis
— Random sampling
— Consider other kinds of prime nodes (e.g. bipartite nodes are prime nodes for parity graphs and were studied asymptotically by Jessica Shi ’18)

Note
— This work is related to independent work project by Sam Pritt ’17, who developed software for extracting enumerations from symbolic equations
Thank you!