Computing and Counting the Longest Paths on Circular-Arc Graphs in Polynomial Time

George B. Mertzios
(University of Haifa, Israel)

Ivona Bezáková
(Rochester Institute of Technology, U.S.A.)

LAGOS 2011, March 29, 2011
The Longest Path Problem

Input: an undirected graph $G$
Output: a path with the largest possible length

Example:
The Longest Path Problem

Input: an undirected graph $G$
Output: a path with the largest possible length

Example:
The Longest Path Problem

Input: an undirected graph $G$
Output: a path with the largest possible length

Example:
The Longest Path Problem

Input: an undirected graph $G$

Output: a path with the largest possible length

Example:
The Longest Path Problem

Input: an undirected graph $G$
Output: a path with the largest possible length

Example:

Hamiltonian path: a path going through all vertices

Note: Longest Path generalizes the Hamiltonian path problem, hence Longest Path is NP-hard
The Longest Path Problem

Hamiltonian and Longest Path are NP-hard on:
- general graphs
- planar graphs
- bipartite graphs
- split graphs
- ... [see, e.g., Mertzios PhD Thesis ’09 for the references]

Inapproximability results [Karger-Motwani-Ramkumar ’97]:
- For any $\epsilon \in (0,1)$, finding a path of length $n-n^\epsilon$ in a graph with a Hamiltonian path is NP-hard
- If $P \neq NP$, no constant-factor approx. for Longest Path
The Longest Path Problem – Related Works

Path problems in polynomial time:

- Ham. Path on
  - proper interval graphs [Bertossi '83]
  - interval graphs [Arikati-Rangan '90, Keil '85]
- circular-arc graphs [Damaschke '93, Hung-Chang-Laio '09, Shih-Chern-Hsu '92]
- cocomparability graphs [Hung-Chang '06, Hung-Chang-Laio '09]
The Longest Path Problem – Related Works

Path problems in polynomial time:

- Longest path – very recent:
  - trees [Bulterman-van der Sommen-Zwaan-Verhoeff-van Gasteren-Feijen ’02]
  - weighted trees and block graphs [Uehara-Unoh ’04]
  - bipartite permutation graphs [Uehara-Valiente ’07]
  - ptolemaic graphs [Takahara-Teramoto-Uehara ’08]

- interval graphs [Ioannidou-Mertzios-Nikolopoulos ’09]
- cocomparability graphs [Ioannidou-Nikolopoulos ’10 – O(n^8), Mertzios-Corneil ’10 – O(n^4)]

Examples of applications of Longest Path:
e.g., computational biology [Colinge-Bennett ’07, Cuntz-Borst ’07]
Interval vs. Circular-Arc Graphs

Interval graphs: intersection graphs of intervals

An interval graph representation:

The corresponding graph:
Interval vs. Circular-Arc Graphs

Circular-arc graphs: intersection graphs of arcs on a circle

A circular-arc graph representation:

The corresponding graph:
Interval vs. Circular-Arc Graphs

For some problems, interval graphs are very different from circular-arc graphs, e.g.:

Minimum chromatic number (coloring):
- polynomial-time for interval graphs [Garey-Johnson-Miller-Papadimitriou '80]
- NP-complete for circular-arc graphs [e.g., Golumbic '04]

Motivation for circular-arc graphs:
- scheduling periodic tasks
Our contributions

- Reduction from circular-arc graphs to interval graphs
- Simplification of the algorithm for interval graphs by Ioannidou, Mertzios, and Nikolopoulos ’09
- Counting and sampling of (“normal”) longest paths
Reduction: circular-arc $\rightarrow$ interval graphs

Some terminology:
- denoting arcs be their right end-points:
- right-end ordering of arcs:
Some terminology:
- denoting arcs be their right end-points:
- right-end ordering of arcs:

- edge $v_1v_2$ is:

right-going  left-going  both
Reduction: circular-arc $\rightarrow$ interval graphs

Some terminology:

- **path-arc representation:**
  representing edges by the red arcs,
  e.g. path $u_6u_2u_3u_1u_4u_5$. 
Some terminology:

- **path-arc representation:** representing edges by the red arcs,
  e.g. path $u_6u_2u_3u_1u_4u_5$:
Reduction: circular-arc $\rightarrow$ interval graphs

Some terminology:

- **path-arc representation:** representing edges by the red arcs, e.g. path $u_6 u_2 u_3 u_1 u_4 u_5$:
Reduction: circular-arc $\rightarrow$ interval graphs

Some terminology:

- **path-arc representation:** representing edges by the red arcs,
e.g. path $u_6 u_2 u_3 u_1 u_4 u_5$:
Reduction: circular-arc → interval graphs

Some terminology:
- **path-arc representation:** representing edges by the red arcs, e.g. path $u_6u_2u_3u_1u_4u_5$:
Reduction: circular-arc → interval graphs

Some terminology:

- path-arc representation:
  representing edges by the red arcs,
  e.g. path $u_6u_2u_3u_1u_4u_5$:
Reduction: circular-arc $\rightarrow$ interval graphs

Some terminology:

- **path-arc representation:** representing edges by the red arcs, e.g. path $u_6u_2u_3u_1u_4u_5$:

**Idea:** for every path $P$, there exists a path $P'$ on the same set of vertices with path-arc representation that does not cover the entire circle
Some terminology:

- **path-arc representation:** representing edges by the red arcs, e.g. path $u_6u_2u_3u_1u_4u_5$:

- given a path-arc representation:
  - **right-cut** and **left-cut** at a vertex $u$:
    number of right- and left-going red arcs containing $u$, e.g. for $u_5$: right-cut = left-cut = 1
  - **cut** at a vertex $u = right-cut + left-cut$
Reduction: circular-arc $\rightarrow$ interval graphs

Some terminology:
- cutting the circle before/after a vertex, e.g. for $u_5$:

original representation

representation for cut before $u_5$ $\rightarrow$ get graph $G_{u_5}$

representation for cut after $u_5$
Reduction: circular-arc $\rightarrow$ interval graphs

Lemma 1: Let $v_1v_2\ldots v_k$ be a path with a path-arc representation such that cut($v_1$) = 0. Then at least one of the graphs obtained from cutting before $v_1$ or after $v_1$ contains the path $v_1v_2\ldots v_k$. 
Reduction: circular-arc \rightarrow interval graphs

**Lemma 2:** Let \( v_1v_2...v_k \) be a path such that \( \sum_{i=1}^{k} \text{cut}(v_i) \) is the smallest possible across all paths on \( \{v_1, v_2, ..., v_k\} \) (and their representations). Then, \( \text{right-cut}(v_1) = 0 \).

**Proof:** By contradiction, \( \text{right-cut}(v_1) > 0 \).
Lemma 2: Let \( v_1v_2...v_k \) be a path such that \( \sum_{i=1}^{k} \text{cut}(v_i) \) is the smallest possible across all paths on \( \{v_1,v_2,...,v_k\} \) (and their representations). Then, right-cut\((v_1) = 0\).

Proof: By contradiction, right-cut\((v_1) > 0\).

Let \( v_iv_{i+1} \) be a right-going edge with red arc containing \( v_1 \).
Lemma 2: Let $v_1v_2...v_k$ be a path such that $\sum_{i=1}^{k} \text{cut}(v_i)$ is the smallest possible across all paths on $\{v_1,v_2,...,v_k\}$ (and their representations). Then, $\text{right-cut}(v_1) = 0$.

Proof: By contradiction, $\text{right-cut}(v_1) > 0$.

Let $v_iv_{i+1}$ be a right-going edge with red arc containing $v_1$.

Consider path $v_iv_{i-1}v_{i-2}...v_1v_{i+1}v_{i+2}...v_k$. 
Lemma 2: Let $v_1v_2...v_k$ be a path such that $\sum_{i=1}^{k} \text{cut}(v_i)$ is the smallest possible across all paths on $\{v_1,v_2,...,v_k\}$ (and their representations). Then, right-$\text{cut}(v_1) = 0$.

Proof: By contradiction, right-$\text{cut}(v_1)>0$. Let $v_iv_{i+1}$ be a right-going edge with red arc containing $v_1$. Consider path $v_iv_{i-1}v_{i-2}...v_1v_{i+1}v_{i+2}...v_k$. 
Reduction: circular-arc \rightarrow interval graphs

Lemma 2: Let \( v_1 v_2 \ldots v_k \) be a path such that \( \sum_{i=1}^{k} \text{cut}(v_i) \) is the smallest possible across all paths on \( \{v_1, v_2, \ldots, v_k\} \) (and their representations). Then, right-cut(\( v_1 \)) = 0.

Proof: By contradiction, right-cut(\( v_1 \)) > 0.

Let \( v_iv_{i+1} \) be a right-going edge with red arc containing \( v_1 \).

Consider path \( v_i v_{i-1} v_{i-2} \ldots v_1 v_{i+1} v_{i+2} \ldots v_k \).

Edge \( v_i v_{i+1} \) disappeared.
Reduction: circular-arc $\rightarrow$ interval graphs

**Lemma 2:** Let $v_1v_2...v_k$ be a path such that $\sum_{i=1}^{k} \text{cut}(v_i)$ is the smallest possible across all paths on $\{v_1,v_2,...,v_k\}$ (and their representations). Then, right-cut$(v_1) = 0$.

**Proof:** By contradiction, right-cut$(v_1) > 0$.

Let $v_iv_{i+1}$ be a right-going edge with red arc containing $v_1$.

Consider path $v_i v_{i-1} v_{i-2} ... v_1 v_{i+1} v_{i+2} ... v_k$.

Edge $v_i v_{i+1}$ disappeared.

What about edge $v_1 v_{i+1}$?
Lemma 2: Let \( v_1 v_2 \ldots v_k \) be a path such that \( \sum_{i=1}^{k} \text{cut}(v_i) \) is the smallest possible across all paths on \( \{v_1, v_2, \ldots, v_k\} \) (and their representations). Then, \( \text{right-cut}(v_1) = 0 \).

Proof: By contradiction, \( \text{right-cut}(v_1) > 0 \).

Claim: edge \( v_1 v_{i+1} \) exists.

[Follows from existence of right-going edge \( v_i v_{i+1} \).]
Lemma 2: Let $v_1v_2...v_k$ be a path such that $\sum_{i=1...k} \text{cut}(v_i)$ is the smallest possible across all paths on \(\{v_1,v_2,...,v_k\}\) (and their representations). Then, $\text{right-cut}(v_1) = 0$.

Proof: By contradiction, $\text{right-cut}(v_1) > 0$.

The new path decreases $\sum_{i=1...k} \text{cut}(v_i)$, a contradiction. \(\square\)
Lemma 2: Let $v_1v_2...v_k$ be a path such that $\sum_{i=1}^{k} \text{cut}(v_i)$ is the smallest possible across all paths on $\{v_1,v_2,...,v_k\}$ (and their representations). Then, right-cut($v_1$) = 0.

Corollary 2: Under the same conditions, left-cut($v_k$)=0.
Reduction: circular-arc → interval graphs

Lemma 2: Let $v_1v_2...v_k$ be a path such that $\sum_{i=1}^{k} \text{cut}(v_i)$ is the smallest possible across all paths on $\{v_1,v_2,...,v_k\}$ (and their representations). Then, $\text{right-cut}(v_1) = 0$.

Corollary 2: Under the same conditions, $\text{left-cut}(v_k)=0$.

Lemma 3: Under the same conditions, $\text{cut}(v_1) = 0$ or $\text{cut}(v_k) = 0$. 
Lemma 2: Let $v_1v_2...v_k$ be a path such that $\sum_{i=1}^{k} \text{cut}(v_i)$ is the smallest possible across all paths on $\{v_1,v_2,...,v_k\}$ (and their representations). Then, right-cut$(v_1) = 0$.

Corollary 2: Under the same conditions, left-cut$(v_k)=0$.

Lemma 3: Under the same conditions, cut$(v_1)=0$ or cut$(v_k)=0$.

Theorem 1: For any path $P$ there exists a path $P'$ on the same vertex set and a vertex $v$ such that the path $P'$ is a path in the interval graph obtained by cutting the circle before $v$. 
**Lemma 3:** Let $v_1v_2...v_k$ be a path such that $\sum_{i=1}^{k} \text{cut}(v_i)$ is the smallest possible across all paths on $\{v_1,v_2,...,v_k\}$ (and their representations). Then, $\text{cut}(v_1)=0$ or $\text{cut}(v_k)=0$.

**Proof idea:**

If $v_1$'s arc goes right and $v_k$'s arc comes from right, we get:

The full proof contains significant case analysis.
The idea of the simplified algorithm

Some terminology:
- **normal paths** in interval graphs:

  \[ v_1v_2...v_k \] is normal if

  - \( v_1 \) is the left-most vertex
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i, v_{i+1}, v_{i+2}, ..., v_k\} \)
The idea of the simplified algorithm

Some terminology:
- **normal paths** in interval graphs:

  \( v_1v_2\ldots v_k \) is normal if
  - \( v_1 \) is the left-most vertex
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i,v_{i+1},v_{i+2},\ldots,v_k\} \)
The idea of the simplified algorithm

Some terminology:

- **normal paths** in interval graphs:

  \[ v_1v_2...v_k \text{ is normal if} \]
  
  - \( v_1 \) is the left-most vertex
  
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i,v_{i+1},v_{i+2},...,v_k\} \)
The idea of the simplified algorithm

Some terminology:

- **normal paths** in interval graphs:

  \[ v_1v_2\ldots v_k \text{ is normal if} \]

  - \( v_1 \) is the left-most vertex
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i,v_{i+1},v_{i+2},\ldots,v_k\} \)
The idea of the simplified algorithm

Some terminology:

- **normal paths** in interval graphs:

  \[ v_1v_2\ldots v_k \] is normal if

  - \( v_1 \) is the left-most vertex
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i,v_{i+1},v_{i+2},\ldots,v_k\} \)
The idea of the simplified algorithm

Some terminology:

- **normal paths** in interval graphs:

  \( v_1v_2\ldots v_k \) is normal if
  
  - \( v_1 \) is the left-most vertex
  
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i,v_{i+1},v_{i+2},\ldots,v_k\} \)
The idea of the simplified algorithm

Some terminology:

- **normal paths** in interval graphs:

  \( v_1v_2...v_k \) is normal if

  - \( v_1 \) is the left-most vertex
  
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i,v_{i+1},v_{i+2},...,v_k\} \)
The idea of the simplified algorithm

Some terminology:
- **normal paths** in interval graphs:
  \[v_1v_2...v_k\] is normal if
  - \(v_1\) is the left-most vertex
  - for every \(i\), \(v_i\) is the left-most neighbor of \(v_{i-1}\) out of all neighbors of \(v_{i-1}\) among \(\{v_i,v_{i+1},v_{i+2},...,v_k\}\)
The idea of the simplified algorithm

Some terminology:

- **normal paths** in interval graphs:

  \( v_1 v_2 \ldots v_k \) is normal if

  - \( v_1 \) is the left-most vertex

  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i, v_{i+1}, v_{i+2}, \ldots, v_k\} \)
The idea of the simplified algorithm

Some terminology:

- **normal paths** in interval graphs:

  \[ v_1v_2...v_k \text{ is normal if} \]
  - \( v_1 \) is the left-most vertex
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i,v_{i+1},v_{i+2},...,v_k\} \)
The idea of the simplified algorithm

Some terminology:
- **normal paths** in interval graphs:

  \[ v_1 v_2 \ldots v_k \] is normal if
  
  - \( v_1 \) is the left-most vertex
  
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i, v_{i+1}, v_{i+2}, \ldots, v_k\} \)
The idea of the simplified algorithm

Some terminology:
- **normal paths** in interval graphs
  
  [Ioannidou-Mertzios-Nikolopoulos '09]:

  \[ v_1 v_2 \ldots v_k \text{ is normal if} \]
  
  - \( v_1 \) is the left-most vertex
  - for every \( i \), \( v_i \) is the left-most neighbor of \( v_{i-1} \) out of all neighbors of \( v_{i-1} \) among \( \{v_i, v_{i+1}, v_{i+2}, \ldots, v_k\} \)

Note:
similar notions introduced in [Damaschke '93, Keil '85]
The idea of the simplified algorithm

Some terminology:

- **normal paths** in circular-arc graphs:
  
  \( v_1v_2...v_k \) is normal if it is normal in the interval graph obtained by cutting the circle before a vertex \( u \)

**Theorem 2:** For every path \( P \) in a circular-arc graph there exists a normal path \( P' \) on the same vertex set as \( P \).
The idea of the simplified algorithm

Algo for interval graphs [Ioannidou-Mertzios-Nikolopoulos ’09]:
- runs in time $O(n^4)$
- for technical reasons uses special dummy nodes that do not (much) modify the length of the longest path

Algo for circular-arc graphs:
- runs in time $O(n^4)$ [avoids the extra cost of the reduction]
- does not need the dummy nodes; as a byproduct of this simplification, it can be used for counting the normal paths [exactly for interval graphs, n-approximation for circular-arc graphs]
The idea of the simplified algorithm

Algo for circular-arc graphs - idea:
- dynamic programming
- need the following:
  - $G_{i}(j)$ := induced subgraph of $G_{u_{i}}$ with vertices $\{u_{i}, u_{i+1}, \ldots, u_{j}\}$ where $i, j \in \{1, 2, \ldots, n\}$
  - $G(i, j)$ := induced subgraph of $G$ with vertices $\{u_{i}, u_{i+1}, \ldots, u_{j}\}\{u_{k} \mid u_{k} \text{ contains the right endpoint of } u_{i-1}\}$ where $i, j \in \{1, 2, \ldots, n\}$ and $j \neq i-1 \mod n$
  - $\ell_{i}(u_{k}, j)$ := the length of a longest normal path of $G_{i}(j)$ with $u_{k}$ as its last vertex
  - $\ell(u_{k}, i, j)$ := the length of a longest normal path of $G(i, j)$ with $u_{k}$ as its last vertex
The idea of the simplified algorithm

**Algo for circular-arc graphs - idea:**

- why $O(n^4)$:
  - two loops to go through all $i, j$ pairs
  - one loop for $k$
  - one loop to consider an intermediate “joining” vertex that connects two shorter paths

- can keep track of the number of all normal paths corresponding to $\ell_i(u_{k,j})$ and $\ell(u_{k,i,j})$ → a counting algorithm
Counting and sampling

Counting (and sampling) of normal paths:
- in $O(n^4)$:
  - exact for interval graphs
  - $n$-approximation for circular-arc graphs
- for some graphs the number of normal paths can be exponentially large

Counting/sampling of paths considered in other works:
- #P-complete for general graphs [Dyer-Frieze-Jerrum '94]
- approx for special graph classes:
  dense graphs [Dyer-Frieze-Jerrum '94], nearly regular [Frieze '00]
- self-avoiding walks in lattice graphs [Randall-Sinclair '00]
Open problems

Highlights from this talk:

Open problems:
- count all longest paths in interval/circular-arc graphs
- improve the approximation ratio for counting normal paths in circular-arc graphs
Open problems

Highlights from this talk:

- count all longest paths in interval/circular-arc graphs
- improve the approximation ratio for counting normal paths in circular-arc graphs

Open problems:
- count all longest paths in interval/circular-arc graphs
- improve the approximation ratio for counting normal paths in circular-arc graphs

Thanks for your attention 😊