Simple Efficient Algorithm for \textit{MPQ}-tree of an Interval Graph

Toshiki SAITOH  Masashi KIYOMI  Ryuhei UEHARA

Japan Advanced Institute of Science and Technology (JAIST)
School of Information Science
Interval Graphs

- Have interval representations
  - Each interval corresponds to a vertex on graph $G=(V, E)$ ($|V|=n$, $|E|=m$)
  - Two intervals intersect $\iff$ corresponding two vertices are adjacent

![Diagram of intervals and graph]

$0 1 2 3 4 5 6$

$I_1, I_2, I_3, I_4$
Applications of Interval Graphs

- bioinformatics
- scheduling problems

DNA sequence

\[ x_1: \text{ACGGTTTA} \]
\[ x_2: \text{ATCGGAACG} \]
\[ x_3: \text{AACGTTTAC} \]
\[ x_4: \text{TTCGACGTGGT} \]

interval graph

ATCGGGAACGGTTTACGTGGT

interval representation
Our Problem

- Input: An interval representation of an interval graph
- Output: An MPQ-tree
  - canonical and compact
  - Isomorphism, (random generation, enumeration)
**PQ-tree** (Booth and Lueker 1979)

- Auxiliary data structure for interval graph
  - Ordered tree
  - Internal nodes are labeled ‘P’ or ‘Q’
  - Leaf ↔ maximal clique
  - $O(n)$ space
- Interval graph recognition
  - $O(n+m)$ time
- Only partial information
  - No information for vertices
**MPQ-tree** (Korte and Möhring 1989)

- Data structure for interval graph
  - Modified $PQ$-tree
  - node $\leftrightarrow$ vertices
  - $O(n)$ space
- Interval graph recognition
  - $O(n+m)$ time
- Interval graph isomorphism
  - $O(n+m)$ time
Are the two interval graphs corresponding to these interval representations isomorphic?
Are the two interval graphs corresponding to these interval representations isomorphic?
Known Algorithms for constructing (M)PQ-trees

   - Input: a graph representation
   - Output: an MPQ-tree
   - $O(n+m)$ time
   - Many conditional branches

   - Input: an interval representation
   - Output: a PQ-tree
   - $O(n \log n)$ time
     - When inputs are sorted, $O(n)$ time
   - Too generalized
MPQ-tree from interval representation

Graph representation

Interval representation

MPQ-tree

PQ-tree
MPQ-tree from Interval Representation

Our approach

$O(n+m)$ time

$O(n log n)$ time

MPQ-tree

$O(n)$ space

PQ-tree

$O(n)$ space
Outline of Our Algorithm

An interval representation
\[ O(n) \text{time} \]

A compact interval representation
\[ O(n) \text{time} \]

\( P \)-nodes, \( Q \)-nodes and leaves
\[ O(n) \text{time} \]

An \( MPQ \)-tree
Characterization of nodes

Theorem

- Leaves
  - Intervals of length 0
- $Q$-nodes
  - Overlapped intervals
- $P$-nodes
  - Other intervals
Order of Sweep

- Sweep intervals from left to right
- Left endpoints precede right endpoints
  - When left endpoint, long interval precedes short intervals
  - When right endpoint, short interval precedes long intervals
Finding $Q$-nodes

for each endpoint $i$ do
  
  if $i$ is a left endpoint, PUSH(S,$i$)

  if $i$ is a right endpoint, compare the stack top with $i$
    
    if they don’t match, the intervals from the stack top to $i$ on stack belongs to a $Q$-node
Outline of Our Algorithm

An interval representation  \( O(n) \) time

A compact interval representation  \( O(n) \) time

\( P \)-nodes, \( Q \)-nodes and leaves  \( O(n) \) time

An \( MPQ \)-tree  \( O(n) \) time
Conclusion

- New algorithm for $MPQ$-trees from interval representations
  - simple
  - $O(n \log n)$
    - When inputs are sorted, $O(n)$ time (theoretically optimal)

Future Works

- Applications of $MPQ$-trees
  - Enumeration and random generation of interval graphs