# Optimal Parallel Algorithms for Cut Vertices, Bridges, and Hamiltonian Path in Bounded Interval Tolerance Graphs

Gur Saran Adhar
Department of Computer Science
Univ. of North Carolina at Wilmington,
NC 28403, USA
adharg@uncwil.edu

## Abstract

*We present parallel algorithms to find cut vertices, bridges, and Hamiltonian Path in bounded interval tolerance graphs. For a graph with $n$ vertices, the algorithms require $O(\log n)$ time and use $O(n)$ processors to run on Concurrent Read Exclusive Write Parallel RAM (CREW PRAM) model of computation. Our approach transforms the original graph problem to a problem in computational geometry. The total work done by the parallel algorithms is comparable to the work done by the best known sequential algorithms for the more restricted class of graphs, namely, interval graphs and permutation graphs. In this sense our algorithms have optimal complexity.*

## 1 Introduction

Bounded interval tolerance graphs offer a more general model to represent wire-nets of a channel on a VLSI chip, than a interval graph model [11] does. Interval tolerance graphs also provide a model to represent conflicts among events occurring within a block of time, when a tolerance of acceptable overlap is associated with each event. For this reason interest to examine problems in bounded interval tolerance graphs has grown [1], [2], [4].

Interval tolerance graphs which were first introduced in [2] generalize both interval graphs and permutation graphs. Other characterizations of interval tolerance graphs place them within the class of trapezoid graphs [4], the class of alternatingly orientable graphs [3], the class of astroid-triple free graphs [2], and in the class of co-comparability graphs [2].

A vertex $v$ of a graph $G$ is a *cut vertex* if the removal of $v$ results in a graph having more components than the graph $G$, and a *bridge* is an edge whose removal results in a graph having more components than the graph $G$. A connected graph with no cut vertices is called *biconnected* and the maximal biconnected subgraphs of $G$ are called *blocks* of graph $G$. The term *biconnected components* is also used in literature for the blocks.

Optimal Parallel algorithms to find cut vertices, bridges, and blocks in general graphs appear in [5] and for the particular class of interval graphs optimal parallel algorithms have been discovered [6]. Sequential algorithms for finding Hamiltonian circuits in proper interval graphs and for permutation graphs appear in [7], and [8].

In this paper we present parallel algorithms for the larger class of bounded interval tolerance graphs, which includes both interval graphs and permutation graphs as subclasses. Our algorithm uses the square embedding representation for the bounded interval tolerance graphs which was first defined in [3].

The paper is organized as follows. Definitions appear in Section 2 and some characteristics are presented in Section 3 which provide the basis of the algorithms. The parallel algorithms and analysis appear in Section 4.

## 2 Definitions

**Definition 2.1- Interval Tolerance Graph**

An undirected graph $G = (V, E)$ is called a *interval tolerance graph* if there exists a collection $\mathcal{I} = \{I_x \mid x \in V\}$ of closed intervals on a line and a set $\mathcal{T} = \{t_x \mid x \in V\}$ of positive numbers satisfying $xy \in E \Leftrightarrow \mid I_x \cap I_y \mid \geq \min\{t_x, t_y\}$ where $\mid I \mid$ denotes the length of interval $I$. The pair $< \mathcal{I}, \mathcal{T} >$ is called *tolerance representation* of $G$. A tolerance representation $< \mathcal{I}, \mathcal{T} >$ is a *bounded* tolerance representation if $t_x \leq \mid I_x \mid$ for every $x \in V$ and a tolerance graph is called a *bounded inteval tolerance* graph if it admits a

bounded tolerance representation.
Bounded interval tolerance representation for an example set of interval-tolerance pairs is shown in Figure 1 and the tolerance graph for the same example set appears in Figure 2.

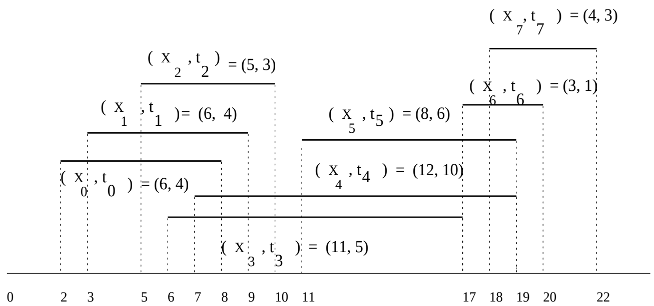


**Figure 1. Tolerance representation for an example set of interval-tolerance pair**

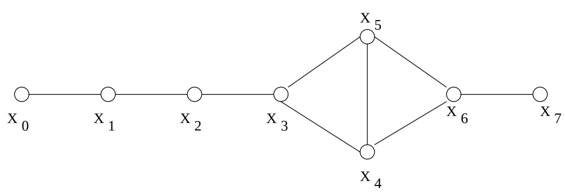


**Figure 2.** *Tolerance Graph G for the example set of interval-tolerance pairs*

Bounded interval tolerance graphs fall within the family of trepezoid graphs [10] where each vertex of the graph maps to a trepezoid and there is an edge between vertex $x$ and vertex $y$ in the graph whenever the intersection between the mappings of $x$ and $y$ is nonempty.

**Definition 2.2-Trepezoid Representation**
An interval-tolerance pair $(I_x, t_x)$ has **T**repezoid **R**epresentation $TR(x)$ defined as follows: given two parallel lines $L_1$ and $L_2$, if $l(x)$ and $r(x)$ denote the left and right end points of interval $I_x$ on the real line, then the trapezoid associated with vertex $x$ is the convex hull of points $l(x)$, and $r(x) - t_x$ on line $L_1$ and the points $l(x) + t_x$, and $r_x$ on line $L_2$.
For the example set of interval-tolerance pairs in Figure 1 the corresponding trepezoid representation appears in Figure 3.
Bounded interval tolerance graphs entertain a canonical square embedding defined in [3]. Each vertex of the graph maps to a square embedding and there is an edge between vertex $x$ and vertex $y$ in the graph whenever the intersection between the shadows of the mappings of $x$ and $y$ on axis-1 or on axis-2 is nonempty.

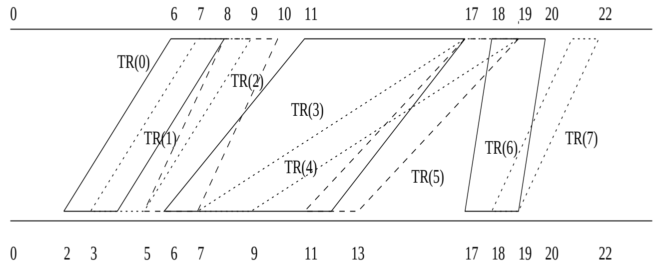


**Figure 3. Trepezoid representation for the example set of interval-tolerance pairs**

**Definition 2.3-Square Embedding Representation**
An interval-tolerance pair $(I_x, t_x)$ has **S**quare embedding **R**epresentation $SR(x)$ in two dimensional rectilinear plane defined as follows: if $l(x)$ and $r(x)$ denote the left and right end points of interval $I_x$ on the real line, then the lower left corner $ll(x)$ of the square embedding $SR(x)$ is at the coordinates $(l(x), l(x)+t_x)$ and the upper right corner $ur(x)$ is at coordinates $(r(x)-t_x, r(x))$. We will use $\mathcal{SR}$ to refer to the set of square embedding representations corresponding to tolerance representation $< \mathcal{I}, \mathcal{T} >$ (Definition 2.1).
For the example set of interval-tolerance pairs in Figure 1 the corresponding square embeddings appear in Figure 4.
An orthogonal polygon is *orthogonally convex* if its intersection with any vertical or horizontal line has at most one connected component. Clearly a square is an orthogonally convex polygon.

**Definition 2.4- Orthogonal Convex Hull of Square Embedding Representation**
*Orthogonal convex hull* $\mathcal{H}(\mathcal{SR})$ of a set $\mathcal{SR}$ of square embedding representations is the smallest orthogonally convex polygon that encloses the set of square embedding representations $\mathcal{SR}$.
For the set of square embeddings $\mathcal{SR}$ in Figure 4 the orthogonal convex hull $\mathcal{H}(\mathcal{SR})$ appears in Figure 5.

**Definition 2.5- Interval Realizer**
The pair of intervals $I_x^1 = [l(x), r(x) - t_x]$ and $I_x^2 = [l(x) + t_x, r(x)]$ are called *interval realizers* of $I_x$. In geometric terms, $I_x^1$ is mapping of $I_x$ to interval $[l(x), r(x) - t_x]$ on axis-1 and $I_x^2$ is mapping of $I_x$ to interval $[l(x) + t_x, r(x)]$ on axis-2. Clearly $| I_x^1 | = | I_x^2 |$.

## 3 Characteristics

The complement graphs of bounded interval tolerance graphs correspond to *partial order with interval dimension* $\leq$ *2* [3]. Which means that, the partial order can

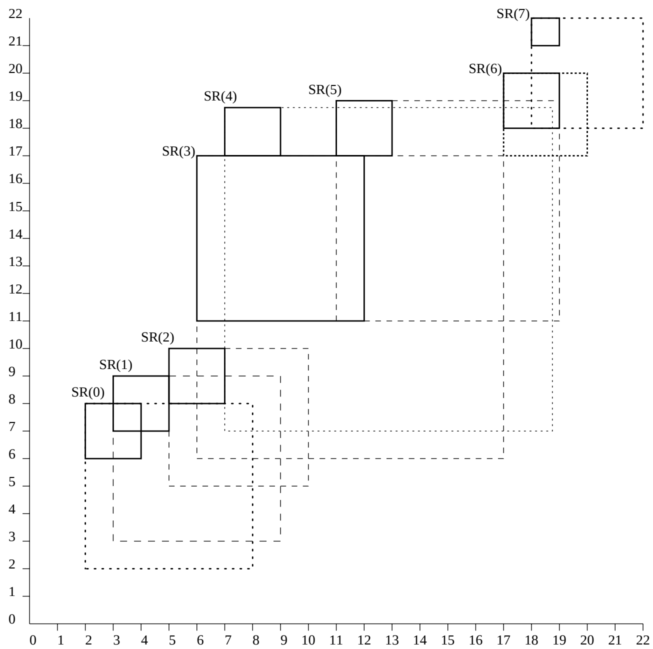


**Figure 4. Square embeddings for the example set of interval-tolerance pairs**

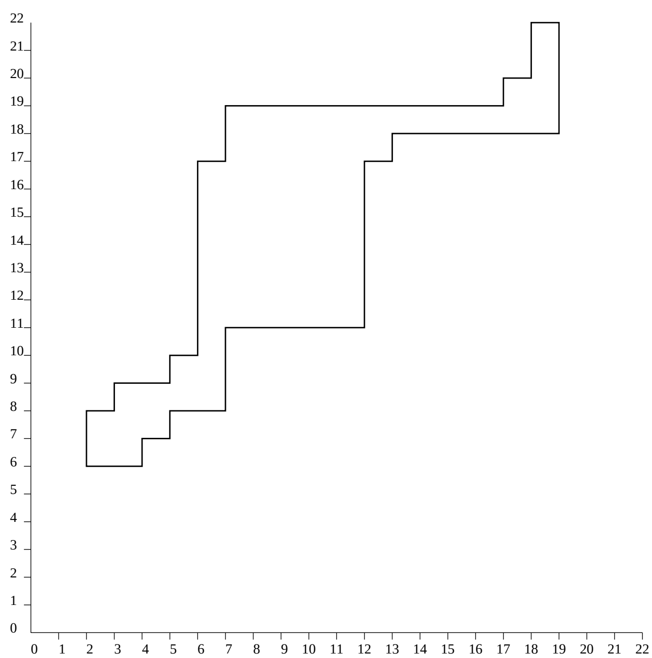


**Figure 5.** *Orthogonal convex hull of square embedding representation of Figure 4*

be recovered from a family of two or less interval orders. When the two interval orders are mapped to two parallel lines we obtain the trepezoid representation (definition 2.2) and when the two interval orders are mapped to two axes in the rectilinear plane we obtain the square embedding (definition 2.3).

If $G^1$ is a interval graph represented by interval order $\mathcal{I}^1$ (intervals of type $I^1_x$ in definition 2.4) and if $G^2$ is a interval graph represented by interval order $\mathcal{I}^2$ (intervals of type $I^2_x$ in definition 2.4). Then the bounded interval tolerance graph is $G = G^1 \cup G^2$ and since $G^1$ and $G^2$ are co-comparability graphs of interval order $P^1$ and $P^2$ respectively, the comparability graph of $P^1 \cap P^2$ is the complement of the bounded tolerance graph $G$. Two lemmas follow immediately.

**Lemma 3.1** *If $C^1$ and $C^2$ are the sets of cut vertices for interval graphs $G^1$ and $G^2$ respectively, then the set of cut vertices $C$ for the graph $G = G^1 \cup G^2$ is $C = C^1 \cap C^2$.*

**Lemma 3.2** *If $B^1$ and $B^2$ are the sets of bridges for interval graphs $G^1$ and $G^2$ respectively, then the set of bridges $B$ for the graph $G = G^1 \cup G^2$ is $B = B^1 \cap B^2$.*

In the remaining section we first show that the geometrical relationship between placement of square embeddings corresponds to the partial order $P^1 \cap P^2$. This property of bounded interval tolerance graphs allows us to work with geometrical representation instead. Next, we describe how orthogonal convex hull (definition 2.4) of square embeddings can be used to derive efficeint algorithms for bounded interval tolerance graphs.

If **l**ower **l**eft corner of a square embedding $SR(x)$ is denoted by $ll(x)$ and **u**pper **r**ight corner is denoted by $ur(x)$ the square embedding $SR(x)$ is said to be "less than" ($<$) square embedding $SR(y)$ if $ur(x) < ll(y)$ that is, $(r(x) - t_x < l(y)$ and $r(x) < l(y) + t_y$. In other words, the square embedding $SR(x)$ is physically to the "left and below" the square embedding $SR(y)$. When two square embeddings $SR(x)$ and $SR(y)$ are not related by $<$ order then they are said to be "incomparable". If $SR(x)$ and $SR(y)$ are incomparable, it will be denoted by $SR(x) \parallel SR(y)$. In this notation the relation between the square embeddings of Figure 4 are denoted as: $SR(0) \parallel SR(1)$, $SR(0) < SR(2)$, $SR(0) < SR(7)$ etc.

The following theorem establishes that if there is an edge $(x, y)$ in the bounded tolerance graph i.e., $| I_x \cap I_y | \geq \min\{t_x, t_y\}$ then the square embeddings $SR(x)$ and $SR(y)$ are incomparable in the partial order $P^1 \cap P^2$.

**Theorem 3.3** *If* $| I_x \cap I_y | \geq \min\{t_x, t_y\}$ *for the interval-tolerance pairs* $(I_x, t_x)$ *and* $(I_y, t_y)$ *then the square* $SR(x)$ *and* $SR(y)$ *are incomparable.*

*Proof:*
**Case 1.** ($| I_x \cap I_y | \geq \min\{t_x, t_y\}$) and $| I_x \cap I_y | < I_x, I_y$
Without loss of generality, let $\min\{t_x, t_y\} = t_x$. Therefore $| I_x \cap I_y | \geq t_x$ and either **(a).** $| I_x \cap I_y | \geq t_y$ or **(b).**$| I_x \cap I_y | < t_y$
Square region embeddings for the two cases Case 1(a) and Case 1(b) are of the form illustrated in Figure 6. In either case $SR(x) \parallel SR(y)$. The alternate case when $\min\{t_x, t_y\} = t_y$ is analogous.

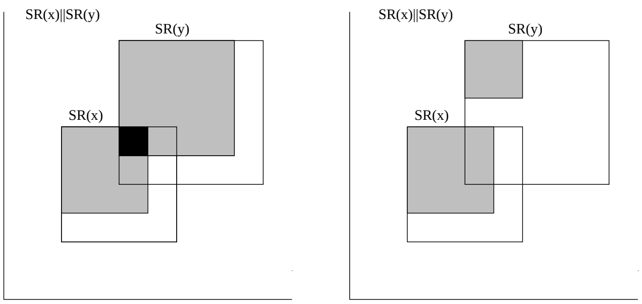


**Figure 6.** *Case 1(a) and Case 1(b)*

**Case 2.** ($| I_x \cap I_y | \geq \min\{t_x, t_y\}$) and $I_x \cap I_y \supseteq I_y$. Square embeddings for this case are of the form illustrated in Figure 7. In this case as well $SR(x) \parallel SR(y)$. The alternate case when $I_x \cap I_y \supseteq I_x$ is analogous.

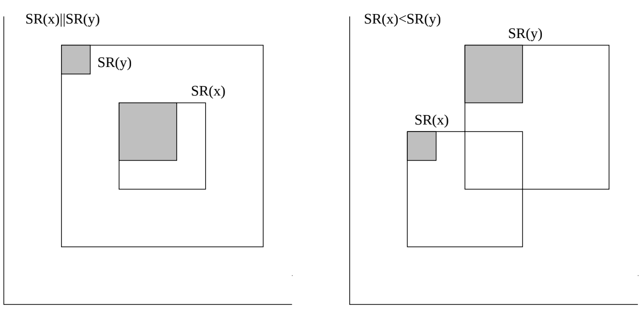


**Figure 7.** *Case 2 and Case 3*

**Case 3.** ($| I_x \cap I_y | < \min\{t_x, t_y\}$)
Without loss of generality let $\min\{t_x, t_y\} = t_x$. Therefore, from the case condition, $| I_x \cap I_y | < t_x$ and also that $| I_x \cap I_y | < t_y$ The square embeddings for this case are of the form illustrated in Figure 7. In this case alone $SR(x) < SR(y)$.

It should be noted here that even though an arrangement of square embeddings such as in Case 2 of the proof are permissible for tolerance graphs in general, they can not exist for bounded tolerance graphs. This is due to the fact that $I_y^2 > I_x^2$ and at the same time $I_y^1 < I_x^1$. This implies that $y > x$ in interval order $P^2$ and simultaneously $y < x$ in the interval order $P^1$ hence the order in $P = P^2 \cap P^1$ can not be recovered, which leads to a contradiction in the definition [3] of bounded tolerance graphs.

**Corollary 3.4** *There is an edge* $(x, y)$ *in bounded interval tolerance graph* $G$ *if and only if at least one of the following conditions holds:*
*(i).* $I_x^1 \cap I_y^1 \neq \emptyset$
*(ii).* $I_x^2 \cap I_y^2 \neq \emptyset$

We say that there is an edge $(x, y)$ in the graph whenever 'shadows' of square embeddings $SR(x)$ and $SR(y)$ overlap either on axis-1 or on axis-2.
In the following Lemma we generalize the notion of shadow of square embedding to the shadow of a orthogonal convex hull (Definition 2.4) and establishe the test conditions for any two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ of $G$ to be disconnected.

**Lemma 3.5** *There is an edge* $(x, y) \in E$ *in a bounded interval tolerance graph* $G = (V, E)$ *between any subgraph* $G_1 = (V_1, E_1), x \in V_1$ *and* $G_2 = (V_2, E_2), y \in V_2$ *if and only if shadow of orthogonal convex hull* $\mathcal{H}(P_1)$ *and* $\mathcal{H}(P_2)$ *overlap either on axis-1 or on axis-2. Where* $\mathcal{H}(P_1)$ *and* $\mathcal{H}(P_2)$ *are orthogonal convex hulls of square embeddings of vertices in* $V_1$ *and in* $V_2$ *respectively.*

*Proof :*
If there is an edge $(x, y)$ between subgraphs $G_1 = (V_1, E_1), x \in V_1$ and $G_2 = (V_2, E_2), y \in V_2$ then from Lemma 3.4 either $I_x^1 \cap I_y^1 \neq \emptyset$ or $I_x^2 \cap I_y^2 \neq \emptyset$ and therefore, $\mathcal{H}(P_1)$ and $\mathcal{H}(P_2)$ must overlap either on axis-1 or on axis-2.
Also, if $\mathcal{H}(P_1)$ and $\mathcal{H}(P_2)$ overlap either on axis-1 or on axis-2 there exists some square embeddings $SR(x)$ and $SR(y)$ whose shadows overlap otherwise it will not be smallest orthogonally convex polygon and therefore the lemma.

## 4 Algorithms

### 4.1 Cut Vertices, Bridges

A vertex $v$ in the graph is a cut vertex whenever deletion of the square embedding $SR(v)$ from orthogonal convex hull results in dividing the convex hull into more than one orthogonal polygons. Lemma 3.5 provides the basis for detecting this condition. If a discontinuity results in the shadow on axis-1 and also on axis-2 from the deletion of a square emedding $SR(v)$ the vertex $v$ is a cut vertex otherwise it is not. In other words, if a vertex $v$ is a cut vertex in the interval graph $I^1$ and a cut vertex also in the interval graph $I^2$ it is a cut vertex for the tolerance graph.
The algorithm assumes that the interval tolerance graph is connected and its square embedding representation is available as input.

**Algorithm Cut_Vertex** $(G, \mathcal{I}, \mathcal{T})$

**begin**

step–i **in parallel** Construct orthogonal convex hull $\mathcal{H}(\mathcal{SR})$ of the of square embeddings $\mathcal{SR}$ by *stitching* square embeddings of individual squares.

step–ii For each vertex $x$ of graph $G$ **in parallel** do

- i **if** $delete(I_x^1)$ results in discontinuity in the shadow on axis-1 set $flag_x^1$;
- ii **if** $delete(I_x^2)$ results in discontinuity in the shadow on axis-2 set $flag_x^2$;
- iii **if** $(flag_x^1 \bigwedge flag_x^2)$ then declare $x$ as a cut vertex.

**end.**

A pair of interval trees [13] $t^1$ and $t^2$, one each for interval realizer $\mathcal{I}^1$ and $\mathcal{I}^2$ are used as data structures to perform operations in step-ii. Each interval tree $t^i$ has exactly $2n$ leaves and $\log n$ height.
The algorithm begins by finding the orthogonal convex hull $\mathcal{H}(\mathcal{SR})$ of the square embeddings. Next, each processor dedicated to a vertex $x$ independently detects the effect of deleting one square embedding $SR(x)$ on the shadow of $\mathcal{H}(\mathcal{SR})$ on axis-1 and on axis-2. Whenever a discontinuity is discovered, after deletion, in the shadow on both axis-1 and on axis-2 the processor declares $x$ as a cut vertex.
With $O(n)$ processors it requires no more than $O(\log n)$ time to construct a *orthogonal convex hull* of the $n$ square embeddings. Answering the question in Step-ii can be completed in $O(\log n)$ time giving the overall complexity.
Parallel algorithm for finding bridges in interval graphs from [6] is used as a function to derive bridges in bounded interval tolerance graphs. The correctness of the algorithm is based on Lemma 3.2.
To recall an important results from [6], we define the *density* of an interval graph $G$ at a point $k$ (written $d_k$) to be the number of intervals which contain $k + \epsilon$ where $0 < \epsilon < 1$. The *density sequence* is the sequence $(d_1, d_2, \ldots, d_{2n}$ for the $2n$ end points of the interval representation. Note that $\mid d_j - d_{j-1} \mid = 1$ for all $j$. We say that interval $x$ *contributes* to $d_k$ if $l(k) \leq k \leq r(k)$.

**Preposition 4.1** *[6] There is a one-to-one correspondence between bridges in interval graph $G$ and $(1,2,1)$-subsequence of the density sequence. For each $(1,2,1)$-subsequence $(d_{j-1}, d_j, d_{j+1})$, the corresponding bridge is the pair of intervals contributing to $d_j$.*

In brief our algorithm proceeds as follows. First, we compute the density sequence for the intervals $\mathcal{I}^1$ (and similarly for $\mathcal{I}^2$). Next we determine the bridges for interval graph $G^1$ (and similarly for $G^2$) by finding the $(1,2,1)$-subsequence in the density sequence. The set of edges which are bridge both in interval graph $G^1$ and in interval graph $G^2$ are declared as bridges of the bounded interval tolerance graph.

**Algorithm Bridges** $(G, \mathcal{I}, \mathcal{T})$

**begin**

step–i Computer density sequence $d^1$ for interval representation $\mathcal{I}^1$.

step–ii Look for $(1,2,1)$-sub-sequences in $d^1$ to obtain, a list of bridges $B^1$.

step–iii Computer density sequence $d^2$ for interval representation $\mathcal{I}^2$.

step–iv Look for $(1,2,1)$-sub-sequences in $d^2$ to obtain, a list of bridges $B^2$.

step–v The set of bridges for the graph $G$ are $B^1 \bigcap B^2$.

**end.**

All the steps can be completed in $O(\log n)$ time, using $O(n/\log n)$ processors on a CREW P-RAM model. Computation of density sequence is indeed an application of parallel prefix sum. The set intersection in step-v can be completed within $O(\log n)$ time, using $O(n)$ processors giving the overall complexity.

## 4.2 Hamiltonian Path

In this section we outline parallel algorithms to find Hamiltonian Circuit in a bounded interval tolerance graph. This algorithm generalizes both the Hamiltonian path algorithm for interval graphs [7] and for the permutation graphs [8].
On the CREW P-RAM model these algorithms execute in $O(\log n)$ time using $O(n)$ processors.
Without loss of generality, we assume that corners of no two square embeddings share a common coordinate.
Let $Ov(SR(i))$ denotes the set of square embedding representations whose shadow overlaps with the shadow of $SR(i)$ either on axis-1 or on axis-2. Formally,

$$Ov(SR(i)) = \{SR(j) \in \mathcal{SR}\}$$

such that

$$\begin{array}{l} l(i) \leq l(j) \leq r(i) - t_i \bigvee \\ l(i) + t_i \leq l(j) + t_j \leq r(i) \end{array}$$

Let

$$ll^* = \min \left\{ \begin{array}{l} \min\{l(j) : SR(j) \in \mathcal{SR}\}, \\ \min\{l(i) + t_i : SR(i) \in \mathcal{SR}\} \end{array} \right\}$$

and

$$ur^* = \max \left\{ \begin{array}{l} \max\{r(j) - t_j : SR(j) \in \mathcal{SR}\}, \\ \max\{r(i) : SR(i) \in \mathcal{SR}\} \end{array} \right\}$$

The 'first' and 'last' squares in $\mathcal{SR}$, denoted as $First(\mathcal{SR})$ and $Last(\mathcal{SR})$, are defined as as follows:

$$First(\mathcal{SR}) = SR(i) \in \mathcal{SR} \mid ll(SR(i)) = ll^*$$

and

$$Last(\mathcal{SR}) = SR(j) \in \mathcal{SR} \mid ur(SR(j)) = ur^*$$

For each square embedding representation $SR(i) \in \mathcal{SR}$ we define 'farthest square with overlapping shadow', denoted $Farthest(SR(i))$, as follows:

$$Farthest(SR(i)) = SR(j) \mid l(i) \leq l(j) \leq r(i) - t_i$$

A 'path of square embedding representations' is a sequence $SR(i_1), SR(i_2), \ldots, SR(i_k)$ of distinct squares in $\mathcal{SR}$ with overlapping shadows, in symbols: $SR(i_j) \in Ov(SR(i_{j-1}))$, $j = 2, 3, \ldots, k$.
A 'Hamiltonian path' is a path of square embedding representations which includes every square in $\mathcal{SR}$.
Finally, $G$ is said to be *connected* if between any two square embeddings in $\mathcal{SR}$ there exists a path joining them.

**Lemma 4.1** *A bounded interval tolerance graph $G$ is connected if and only if $\bigcup_1^n [ll(i), ur(i)]$ forms a single rectangle $[ll^*, ur^*]$.*

**Proof:** Assume that $G$ is connected. Then there must exist a path joining $First(\mathcal{SR})$ and $Last(\mathcal{SR})$. Thus

$$\bigcup_{i=1}^{n} [ll(i), ur(i)] = [ll^*, ur^*]$$

Conversely, assume

$$\bigcup_{i=1}^{n} [ll(i), ur(i)] = [ll^*, ur^*]$$

and $G$ is not connected. Then there must exist two squares which cannot be joined with any path. Thus there exists a square $c$, $ll^* < c < ur^*$ which does not overlap any other square: a contradiction.

**Theorem 4.2** *A proper bounded interval tolerance graph $G$ has a Hamiltonian path if and only if it is connected.*

**Proof:** Obviously if $G$ has a Hamiltonian path, it is connected. Conversely, Assume $G$ is connected. Then there must exist a path $p = SR(i_1), SR(i_2), \ldots, SR(i_m)$, $m \leq n$, such that $First(\mathcal{SR}) = SR(i_1)$, and $Last(\mathcal{SR}) = SR(i_m)$.

Let $K = \mathcal{SR} - \{SR(i_1), SR(i_2), \ldots, SR(i_m\}$. Observe that since $G$ is proper, for each $SR(k) \in K$ there must exist two squares of $p$ both overlapping with $SR(k)$ (either on dimension-1 or on dimension-2).

Consider iteratively each pair $(SR(i_j), SR(i_{j+1}))$, $j = 1, \ldots m-1$, of squares in $p$. If $SR(h) \in Ov(SR(i_j))$ and $SR(h) \in Ov(SR(i_{j+1}))$ then

$$\begin{aligned} p_j \; = \; & SR(i_1), SR(i_2), \ldots, SR(i_j), \\ & SR(h), SR(i_{j+1}), SR(i_{j+2}) \ldots, SR(i_m) \end{aligned}$$

is still a path. The above reasoning can be iterated until the set $K$ is emptied. The resulting path is then a Hamiltonian path of the entire graph.

```
Algorithm Hamiltonian_Path(G, SR)

begin
step–i  p = SR(i_1), SR(i_2), ..., SR(i_m), where

        First(SR) = SR(i_1),
        Last(SR) = SR(i_m),
        Farthest(SR(i_j)) = SR(i_{j+1}), for j = 1, ..., m − 1

step–ii  K = SR − {SR(i_1), ..., SR(i_m)}
step–iii  while (K ≠ ∅)
     {
     for each (SR(i_j), SR(i_{j+1})) do in parallel
        If
                (SR(h) ∈ K)∧
                (SR(h) ∈ Ov(SR(i_j)))∧
                (SR(h) ∈ Ov(SR(i_{j+1})))
        then
        {

          p  =  SR(i_1), ..., SR(i_j),
                SR(h), SR(i_{j+1}) ..., SR(i_m)

        K = K − SR(h)
        }
     }
end.
```

The algorithm assumes that the tolerance interval graph $G$ is connected and its square embedding representation $\mathcal{SR}$ is available as input. A pair of interval trees [13] $t^1$ and $t^2$, one each for interval realizer $\mathcal{I}^1$ and $\mathcal{I}^2$ are used as data structures to evaluate condition in step-iii. Each interval tree has $O(n)$ leaves and $O(\log n)$ height. With $O(n)$ processors the algorithm can be executed in $(\log n)$ time.

## 5 Conclusion

We have presented parallel algorithms to find cut vertices, bridges, and Hamiltonian path in bounded tolerance graphs which include both interval graphs and permutation graphs as subclasses. The algorithm takes advantage of the fact that bounded tolerance graphs have canonical square embedding in the rectilinear plane. This approach transforms the original graph problem to a problem in computational geometry.

## References


[1] Giri Narasimhan, and Rachael Manber *Stability Number and Chromatic Number of Tolerance Graphs*, Univ. of Wisconsin Madison Computer Science Department Tech Report, 1988.

[2] Martin Charles Golumbic, Clyde L. Monma, and William T. Trotter, *Tolerance Graphs*, Discrete Applied Mathematics, 1984, 157-170.

[3] Stefen Felsner, *Tolerance Graphs and Orders*, Tech. Report No. 309/1991, Technische Universitat Berlin, Germany.

[4] Bogart, K.P., Isaak, G., Langley, L. and Fishburn, P.C., *Proper and Unit Tolerance Graphs*, Tech. Report No. 91-74/1991, DIMACS.

[5] Tsin, Y.H. and Chin, F.Y., *Efficient Parallel Algorithms for a class of graph theoretic problems*, SIAM J. of Computing, 13:580-599, 1984.

[6] Sprague, Alan P. and Kulkarni, K.H., *Optimal Parallel Algorithms for finding cut vertices and bridges in interval graphs*, SIAM J. of Computing, 1991.

[7] Bertossi, Alan A., *Finding Hamiltonian Circuits in Proper Interval Graphs*, Information Processing Letters, 1983.

[8] Sastry, Shanthi P., Jayakumar N., and Veni Madhavan C.E. *Domination and Hamiltonian Circuit problem in Permutation Graphs* , Springer Verlag Lecture Notes in Computer Science, 1991.

[9] Olariu, S. Schwing, J.L. and Zhang, J., *Optimal Parallel Algorithms for problems modelled by a family of Intervals*, In Proc. of 28th Annual Allerton Conf. on Comm. Control and Computing, pp. 282-291, 199.

[10] Dagan, I., Golumbic, M.C. and Pinter, R.Y., *Trapezoid Graphs and their coloring*, Discrete Applied Mathematics, 21 (1988), 35-46.

[11] Giovanni De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw Hill Pub., (1994).

[12] Selim G. Akl, Kelly A. Lyons *Parallel Computational Geometry*, Prentice Hall, (1993).

[13] Franco P. Preparata, Michael Ian Shamos, *Computational Geometry An Introduction*, Springer-Verlag, (1985).

[14] Andreae, T., Henning, U., Parra, A., *On a problem concerning Tolerance Graphs*, Discrete Applied Mathematics, 1993, 73-78.

[15] Langley, Larry L., *The Predecessor Successor Relation and a recognition algorithm for orders of dimension two*, Tech. Report No. PMA-TR92-102, Department of Mathematics and Computer Science, Dartmouth College.

[16] Bogart, Kenneth P., and Trenk, Ann N., *Bipartite Tolerance Orders*,

[17] David Kelly , and William T. Trotter, *Dimension Theory for Ordered Sets*, 1984, 171-211, Proceedings of the NATO Advanced Study Institute Aug. 28 1981, D. Reidel Co. Pub.

[18] Springsteel, Fredrick and Stojmenovic, Ivan, *Parallel General Prefix Computations with Geometric, Algebraic, and other Applications*, International Journal of Parallel Programming, 485-503, Vol.18, No. 6. 1989.

[19] Joseph O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, (1987).