Visibility representations

7.1 Introduction

The tessellation representation of a planar st-graph *G* seems rather awkward to the human eye, mainly because it represents edges as faces, and faces as edges, while everybody is tought to assume the opposite. The visibility representations can be seen as an enhancement of the tesselation representation, that correct this. In visibility representations, the vertices are displayed as horizontal segments, the edges as vertical segments that link two horizontal segments without crossing another, and the faces as the area that is defined between horizontal segments and vertical lines.

The visibility representations that we just described are based on a form of visibility called weak-visibility. We can define more strick forms of visibility called e-visibility and s-visibility, to obtain more visibility representations that satisfy more constrains. We shall formally refer to the forms of visibility and their representations, after the description of the weakest form of visibility, and an algorithm to construct visibility representations for it.

Definition 7.1: *Weak Visibility representation G of a planar st-graph G is the set of horizontal vertex-segments G(?) that represent the vertices of G, and vertical edge-segments G(u,?) that represent the edges of G, such that*:

1. *no two vertex-segments overlap*
2. *no two edge-segments overlap*
3. *each edge-segment G(u,?) has its bottom endpoint on G(u), its top endpoint on G(?), and does not intersect any other vertex-segment.*

A visibility representation of a planar st-graph *G* can be drown directly inside the tesselation representation of *G*. Actually, if we shrink leftwards each rectangular area that represents an edge at the tesselation representation to a line segment, we obtain the visibility representation from the tesselation representation. This is one way to prove that the visibility representation of a planar st-graph exists. This can also be proved directly from the construction principles of the Algorithm 7.1 that computes such a representation.

Theorem 7.1: *Let G be a planar st-graph with n vertices. Then, Algorithm 7.1 Visibility constructs in* $O(n)$ *time a visibility representation of G with integer coordinates and* $O(n^2)$ *area.*

Proof: From the construction of the algorithm and Lemma 4.4 we can observe that any two vertex-segments are separated by a horizontal or vertical strip of at least unit width. In the representation constructed by the algorithm, we can observe that no two faces intersect except for their common edges.

We will make an example of this algorithm. In Figure 7.1 is a planar st-graph *G* and its dual. At the top of Figure 7.2 we have the dual G^* of G' and its optimal weighted topological numbering. This numbering provides the x-coordinates for the visibility representation. At the right of Figure 7.2 we have *G* and it's optimal weighted topological numbering which provides the y-coordinates for the visibility representation.

We will make an example for a specific vertex, say $v2$ and find the horizontal vertex-segment of it's visibility representation according to step 5 of the algorithm. This vertex has $left(v2) = f3$, $right(v2) = f7$, the x-coordinate of $f3$ according to the optimal weighted topological numbering of the dual of G is $X(f3) = 2$, while the of $f4$ is $X(f4) = 6$. So, according to step 5 the vertex-segment should be drawn from 2 to 5. The y-coordinate of the vertex-segment is directly provided by the optimal weighted topological numbering of G.

For a specific edge, say ($v8$,$v6$), we follow the step 4 of the algorithm. The y-coordinates of ($v8$,$v6$) can be directly obtained from the optimal weighted topological numbering of G, specifically $y_{bottom} = 1$, $y_{top} = 2$, as it is shown in Figure 7.2. Edge ($v8$,v6), has *left*($v8$,v6) = $f4$, while it has *right*($v8$,v6) = $f6$. The x-coordinate is the number if the left facet of the edge, thus the number that is provided by the optimal weighted topological numbering of $f4$, specifically $x = 3$.

Algorithm 7.1: Visibility

Input: planar st-graph G with n vertices.
Output: visibility representation G of G with integer coordinates and area $O(n^2)$.

G= Visibility(G)

1) Construct the G^* as the dual graph of G.
2) Assign unit weights to the edges of G and compute an optimal weighted topological number Y of G.
3) Assign unit weights to the edges of G^* and compute an optimal weighted topological numbering X of G^*0.
4) Draw the vertival segments:

for each *edge e* **in** *p* **do**

draw G(e) as the vertical segment with:

$x(\Gamma(e)) = X(left(e));$

$y_B(\Gamma(e)) = Y(orig(e));$

$y_T(\Gamma(e)) = Y(dest(e));$

endfor

5) Draw the horizontal segments:

for each *vertex ?* **do**

draw G(?) as the horizontal segment with:

$y(\Gamma(u)) = \Upsilon(u);$

$x_L(\Gamma(u)) = \Upsilon(left(u));$

$x_R(\Gamma(u)) = \Upsilon(right(u)) - 1;$

endfor

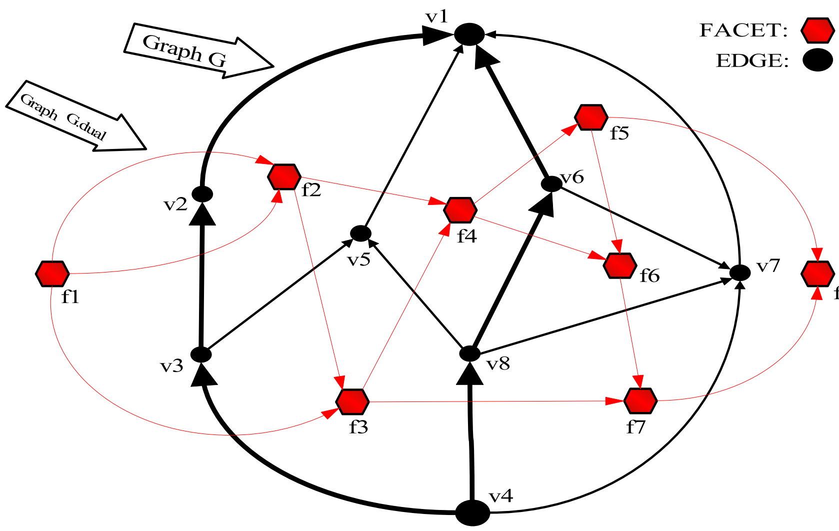


Figure 7.1: graph G and its dual

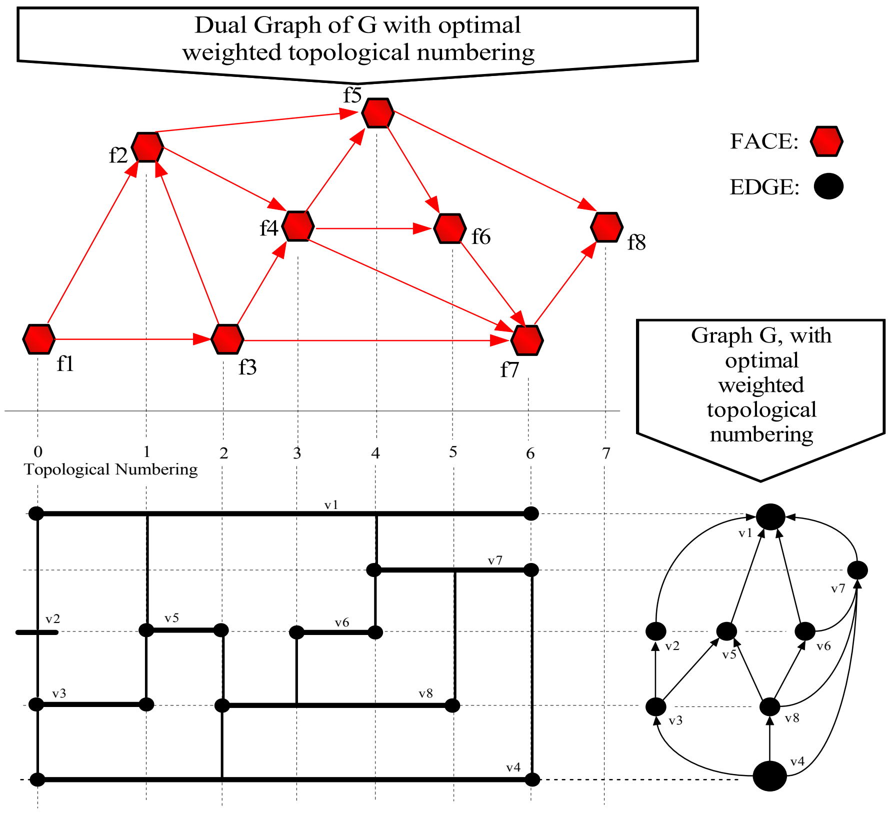


Figure 7.2: Visibility representation of graph G.

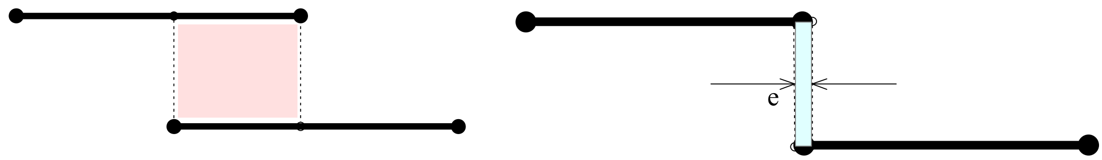
7.1 Forms of visibility representations.

The visibility representation described in the previous section is a rather loose one. We may require additional constrains to be fulfilled. While some of these additional constrains appear for practical reasons at VLSI layer compaction strategies, some other are interested for theoretical purposes. In order to understand the three forms of visibility representation, we must first define two forms of visibility which defines if two vertex-segments are visible or not.

Definition 7.2: *Two vertex-segments are called visible in a visibility representation, when they can be joined by a rectangular area, orthogonal to them, which does not intersect any other segment.*

Definition 7.3: *Two vertex-segments are called e-visible in a visibility representation, when they can be joined by a rectangular area of width e, orthogonal to them, which does not intersect any other segment, and e may tend to zero.*

A schematic representation of these two forms of visibility is in Figure 7.3. The difference between visibility and e-visibility, is that visibiliry requires that the rectangular area which links the two segments has a fixed non-zero width, while in e-visibility, this width can approach zero. From this definition, we can observe that when two segments are visible, the are also e-visible, while the opposite does not stand. Thus, in Figure 7.3a, the two segments are visible and e-visible, while in Figure 7.3b, the two segments are e-visible, but not visible, since the rectangular area, orthogonal to them, which links them, has a non fixed width that tends to zero. So, e-visibility is more generic since it includes visibility.



(a) (b)

Figure 7.3: (a) two visible and e-visible segments, (b) two e-visible segments

Now we can provide the formal definitions of the three forms of visibility. The first form, the w-visibility is the case we described at the previous section, does not pose any restrictions to how the segments are visible according to their neighbors. If we want only neighbor vertices to be visible, something usefull in VLSI layour compaction, we have to move to the other two forms of visibility representations. The second form is the e-visibility, which requires that every two neighbor vertice-segments must be e-visible and that every two vertice-segments which are e-visible must be neightbors. The third form of visibility is s-visibility which requires that every two neighbor vertice-segments must be visible and that every two vertice-segments which are visible must be neighbors. Obviously, s-visibility is more generic than w-visibility.

Definition 7.4: *A w-visibility representation for a graph G = (V,E) is a mapping of vertices of G into nonoverlapping horizontal segments (called vertex-segments), and of edges of G into vertical segments (called edge segments) such that, for each edge (u,?) in E, the associated edge-segment has its endpoints on the vertex segments corresponding to u and ?, and it does not cross any other vertex-segment.*

Definition 7.5: *An e-visibility representation for a graph G is a w-visibility representation with* *the additional property that two vertex-segments are e-visible if and only if the corresponding vertices of G are adjacent.*

Definition 7.6: *An s-visibility representation for a graph G is a w-visibility representation with additional property that two vertex-segments are visible if and only if the corresponding vertices of G are adjacent.*

If a graph admits any of these three visibility representations, then it is planar, since a planar embedding of it can be immediately obtained from the visibility representation by shrinking each vertex-segment to a point.

Here in Figure 7.4, there is an example of a visibility representation of each type. in Figure 7.4a, is the example graph G. in Figure 7.4b, there is the w-visibility representation of G. As it can be observed, two vertex segments are visible even if they are not neighbors. in Figure 7.4b, there is the e-visibility representation of G. Note that the vertex-segments *v*3, *v*4 are represented by intervals that do not include the left point of the segment. The inclusion of that point on the intervals will make *v*2 *e-visible* with *v*3, and *v*3 *e-visible* with *v*4, but at neither case, *visible* with *v*4. This is the case in Figure 7.4d, where *v*2 "sees" *v*3, and *v*3 "sees" *v*4, at a single point, thus through a narrow width rectangular area, of width that approaches zero.

The question that normally arises after the definition of the three types of visibility, is, given a planar st-graph, does a w-visibility or e-visibility or s-visibility representation exists for it? Obviously, the w-visibility representation exists for every planar st-graph, and the algorithm that constructs such a representation was seen at the previous section. The following will answer this question for the e-visibility and the w-visibility case.

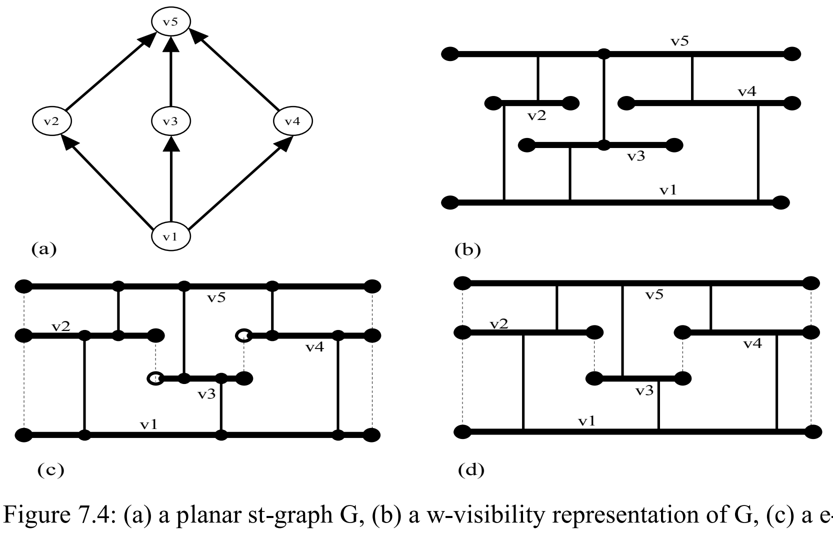


Figure 7.4: (a) a planar st-graph G, (b) a w-visibility representation of G, (c) a e-visibility representationn of G, (d) a s-visibility representation of G

Lemma 7.1: *If the graph G admits an e-visibility representation, then there exists a planar* *embedding of G, such that all cutpoints appear on the boundary of the external face.*

The proof of this lemma is beyond the purposes of this lecture notes, and can be found at the bibliography. An example of a graph that does not allow a e-visibility representation of it is in Figure 7.5. As it can be seen, there is way to embed this graph with an embedding that brings the internal cutpoint to an external face. The internal cutpoint will always be inside an internal face.

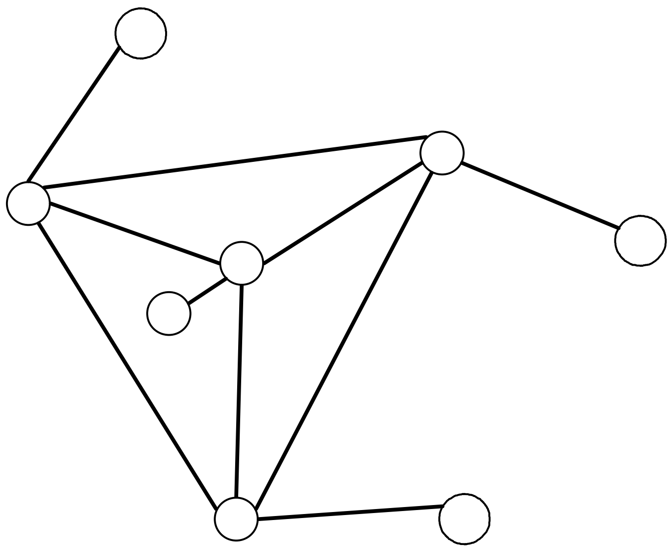


Figure 7.5: A planar st-graph G that does not allow a e-visibility representation of it.

Lemma 7.2: *If there is a planar embedding of a planar st-graph G=(V,E), with all the cutpoints (if any) at the external face, then an e-visibility representation can be constructed in O(|V|).*

This lemma tells as that there is an algorithm that computes an e-visibility representation of a planar st-graph *G*, if such a representation exists, according to Lemma 7.1. From Lemma 7.1 and Lemma 7.2 we can conclude to the following theorem.

Theorem 7.1: *A graph G admits an e-visibility representation if and only if there is a planar embedding for G, such that all cutpoints appear on the boundary of the same face.*

And the following corollary comes straightforward:

Corollary 7.1: *Let G' be the graph obtained from G by adding a new vertex and connecting it to all cutpoints of G. Then G admits an e-visibility representation, if and only if G' is planar.*

It is harder to answer if there is an s-visibility representation for a planar st-graph *G*. For the s-visibility representation we can conclude to a lemma is similar to Lemma 7.1.

Lemma 7.3: *If a graph G admits a s-visibility representation, then there exists a planar embedding of G, such that all cutpoints appear on the boundary of the external face.*

This lemma obviously does not answer the question of the existance of a visibility representation for a given planar st-graph. The following theorem does answer, but only for 2-connected graphs.

Theorem 7.2: *A 2-connected graph G admits an s-visibility representation with bottommost vertex-segment s and topmost vertex-segment t, if and only if there is a strong st-numbering for G.*

We remind that strong st-numbering of *G* is the st-numbering on a planar embedding *G'* of *G*, such that s and t appear on the boundary of the external face, and for every internal face f of *G'*,the vertices *l*(*f*) and *h*(*f*) are joined by the arc [*l*(*f*),*h*(*f*)]. Note that *l*(*f*) and *h*(*f*) is lowest and highest numbered vertices on the boundary of a face of G'.

For 4-connected graphs, we can also answer the existance question, and additionaly provide an algorithm with complexity $O(|V|^3)$.

Theorem 7.3: *Every 4-connected planar graph G = (V,E) admits an s-visibility representation which can be computed in time* $O(|V|^3)$.

The proof of the above lemmas and theorems is beyond the scope of this material.

7.3 Constrained Visibility Representations

In some cases there is a need to give more emphasis to certain paths of a planar graph. These paths often called critical, should discriminate from all the other paths, on the visibility representation of the planar graph. Such a visibility representation could be used, for example, for having a quick inspection of critical paths on a workflow graph. Critical paths in that case they have the potential of delaying a project, thus they should be emphasized in a visibility representation. A way to emphasize certain paths in a visibility representation is to align their edges to the same horizontal coordinate. Such a visibility representation is called constrained visibility representation. This representation can be used as a starting point for obtaining orthogonal and polyline drawings with interesting properties, as will be shown in the next section.

It is easy to construct such an algorithm, based on the algorithm of the visibility representation. Let *G* be a planar st-graph with *n* vertices. The key idea is to construct a new planar st-graph *G'* that has an extra facet for each critical path. This can be done by dublicating each critical path. The visibility representation of that graph will have the edge segments of the left side of the boundary of each extra facet, vertically aligned. By removing the right copy of every edge of the dublicated path, and joining the copies of the dublicated vertices, we have each critical path aligned to one x coordinate. The new facet for every critical path can be inserted directly to the dual of *G*, as a new vertex, at it will be shown, after the following definitions:

Definition 7.1: *Two paths p1 and p2 of a planar st-graph G, are said to be non intersecting, if they do not share any edge, and do not cross at common vertices.*

This means that if the two paths *p1* and *p2* have a common vertice, which is the case in Figure 1, then two consecutive vertices, in the clockwise or anticlockwise order, must belong to the same path. For example in Fig. 1, two paths are intersecting if *e*1 and *e*3 belong to the same path, while *e*2 and *e*4 belong to another.

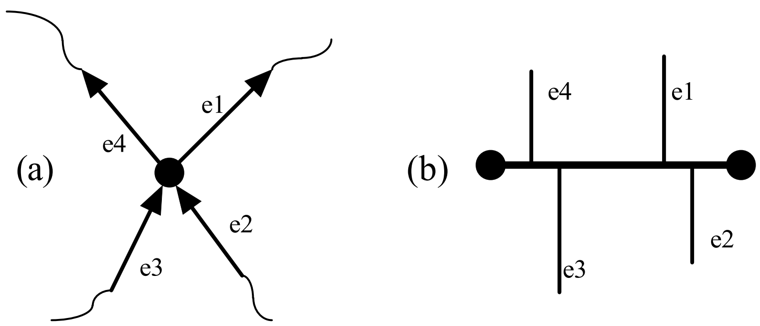


Figure 7.1: Two paths meet on an edge and it's visibility representation's segment.

Lemma 7.1: *A Constrained Visibility representation of a planar st-graph exists if and only if the critical paths do not intersect.*

Proof: If the two paths intersect then the edges at the visibility representation could not be aligned. Take the case in Figure 1. Let *e*2, *e*4 belong to one path and *e*1, *e*3 to another, then if *e*1 is at the right, *e*3 must be at the left of *e*2, thus *e*1 and *e*3 could never be on the same side to share the same x coordinate.

In order to align the vertical edges of a critical path of the visibility representation to the same *x* coordinate we must insert a facet that restricts the *x* coordinate of each edge at the critical path. This could be done easily to the dual graph of *G*, by inserting a new vertex that receives all incoming edges from their sources, and sends all outcoming nodes to their targes, for each critical path.

Without any loss of generality, let every edge of *G* to be a single edged critical path. Let *?* be the set of nonintersecting critical paths. For example in Figure 2, the critical paths are { (*v*4*?* *v*3*?* *v*2*?* *v*1), (*v*4*?* *v*8*?* *v*6*?* *v*1) }, so the set *?* has the following paths:

? = { (*v*4*?* *v*3*?* *v*2*?* *v*1), (*v*4*?* *v*8*?* *v*6*?* *v*1), (*v*3*?* *v*5), (*v*8*?* *v*5), (*v*5*?* *v*1), (*v*8? *v*7), (*v*4*?* *v*7), (*v*7*?* *v*1), (*v*6*?* *v*7) }

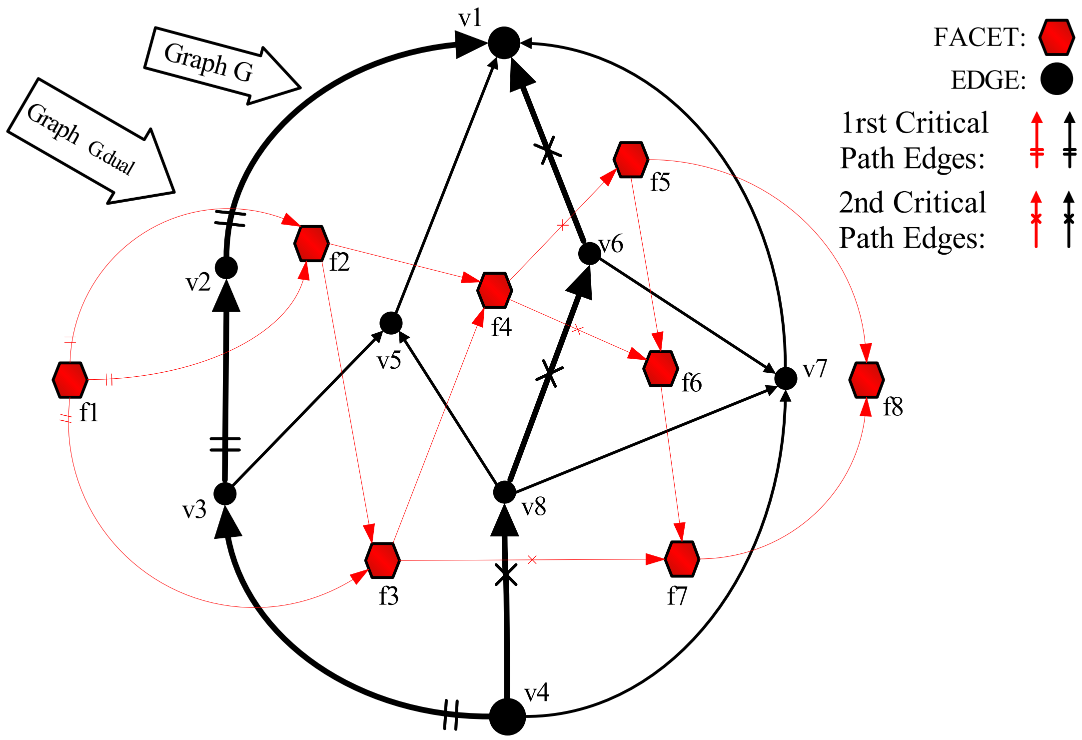


Figure 7.2: A graph *G* and it's dual *G**.

Notice that each edge of *G* uniquely defines an edge at its dual *G**. So, the set *?* of edges of *G* uniquely defines a set of edges at *G**. Now, for each path *p* that belongs to *?* , a new facet is inserted at *G**. Let the new graph formed by this procedure be *Gp*. *Gp* is constructed by forcing the facets at the left side of each critical path to link to the facets at the right side of that critical path, through the new facet inserted for that critical path. The *Gp* graph of graph *G* presented in Figure 2 is drawn in Figure 3. Note that all facets around path (*v*4? *v*3? *v*2? *v*1) are linked to the destination facets through the new rectangular facet '*x*1', and that all facets around path (*v*4? *v*8? *v*6? *v*1) are linked through '*x*2'. Single edge critical paths, are linked through *x*3,*x*4,*x*5,*x*6,*x*7,*x*8,*x*9 new circular facets.

Since *G* is a planar st-graph, it's dual, *G** is also a planar st-graph. The insertion of new facets at *G** doesn't change either the planarity of *G** nor the fact that there is an st-numbering for it. Every edge *e* of *G*, belongs to a path *p* of *?* , and has a left and a right facet. Therefore, every internal facet of *G* has some path to it's left and some path to it's right. No path is to the left of *s** or to the left of *t**. Hence, *Gp* contains no directed cycles, has one source *s** and one sink *t**. Clearly, *Gp* is directed and planar. Finally, notice that both *s** and *t** are on the external face of *Gp*. Therefore, *Gp* is also a planar st-graph. More formally:

Definition 7.2: *Let G be a planar st-graph, and G.dual, it's dual. Let ? be a set of non-intersecting paths that covers the edges of G. We construct Gp as the graph with the vertex set* $F \cup \Pi$ *and edge set { (f, p) | f = left(e) for some edge e of path p }* $\cup$ *{ (p,g) | g = right(e) for some edge of path p }.*

Lemma 7.2: *The digraph Gp constructed by the definition above is a planar st-graph.*

The topological numbering of *G* and *Gp* will actually provide the y and x coordinate of the horizontal vertices and the vertical edges of the contrained visibility representation. Now we can present the algorithm that constructs the constrained visibility representation:

Algorithm 7.1: Constrained Visibility

Input: planar st-graph *G* with n vertices; set *?* of non-intersecting paths covering the edges of *G*.
Output: constrained visibility representation *G* of *G* with integer coordinates and area $O(n^2)$.

G = Constrained_Visibility(*G*,*?*)

1) Construct *Gp*.
2) Assign unit weights to the edges of G and compute an optimal weighted topological numbering *Y* of *Gp*, such that *X(s) = 0.*
3) Assign half-unit weights to the edges of Gp and compute an optimal weighted topological numbering *X* of *Gp*, such that *X(s*) = -0.5.*
4) Draw the vertical segments:

    for each *path p* **in** *?* **do**
        for each *edge e* **in** *p* **do**
            draw G(e) as the vertical segment with:
            $x(\Gamma(e)) = X(p);$
            $y_B(\Gamma(e)) = Y(orig(e));$
            $y_T(\Gamma(e)) = Y(dest(e));$
        endfor
    endfor

5) Draw the horizontal segments:

    for each *vertex ?* **do**
        draw G(?) as the horizontal segment with:
        $y(\Gamma(u)) = \Upsilon(u);$
        $x_B(\Gamma(u)) = \min_{u \in p}\{X(p)\};$
        $x_T(\Gamma(u)) = \max_{u \in p}\{X(p)\};$
    endfor

An example of this algorithm is shown on Figure 3. At the upper side in Figure 3 there is graph *Gp* topologicaly ordered according to step 3 of the algorithm. At the right side of Figure 3 a graph *G* is topologicaly ordered according to step 2. The indexes of the extra facets of *Gp* provide the *x* coordinates and the indexes of the vertices of *G* provide the *y* coordinates, for the vertical edges and the horizontal segments, as it is stated in steps 4, 5. The operations at step 3 are quite clear, while the operations at step 5 need some explaining.

At step 5 we compute a horizontal segment of the constrained visibility representation for each edge of *G*. The *y* coordinate of that segment is the index provided by the topological numbering of *G*. The horizontal segment starts at the minimum *x* coordinate among the *x* coordinates of all the paths p of ? that contain this edge, and ends at the maximum x coordinate among the x coordinates of all the paths p of ? that contain this edge. For example, the edge $v8$ of G is contained in paths ($v4$? $v8$? $v6$? $v1$), ($v8$? $v5$), ($v8$? $v7$). These paths have x coordinates 3, 2, 5 respectively. Thus the horizontal segment will be:

$$[x_L, x_R] = [\min\{3,2,5\}, \max\{3,2,5\}] = [2, 5].$$

Note that x_L could be equal to x_R, as it happens at vertex $v2$ of G in Figure 3.

Theorem 7.1: *Let G be a planar st-graph with n vertices, and let ? be a set of nonintersecting paths covering the edges of G. The algorithm Constrained_Visibility computes in O(n) time a visibility representation of G with integer coordinates and O(n^2) area, such that the edges of every path p in ? are vertically aligned.*

The optimal weighted topological numbering ensures that the area will be $O(n^2)$. A non optimal weighted topological numbering would require more area. Also, the weights need not be set to unit. An arbitrary positive weighting of G will not affect the algorithm.

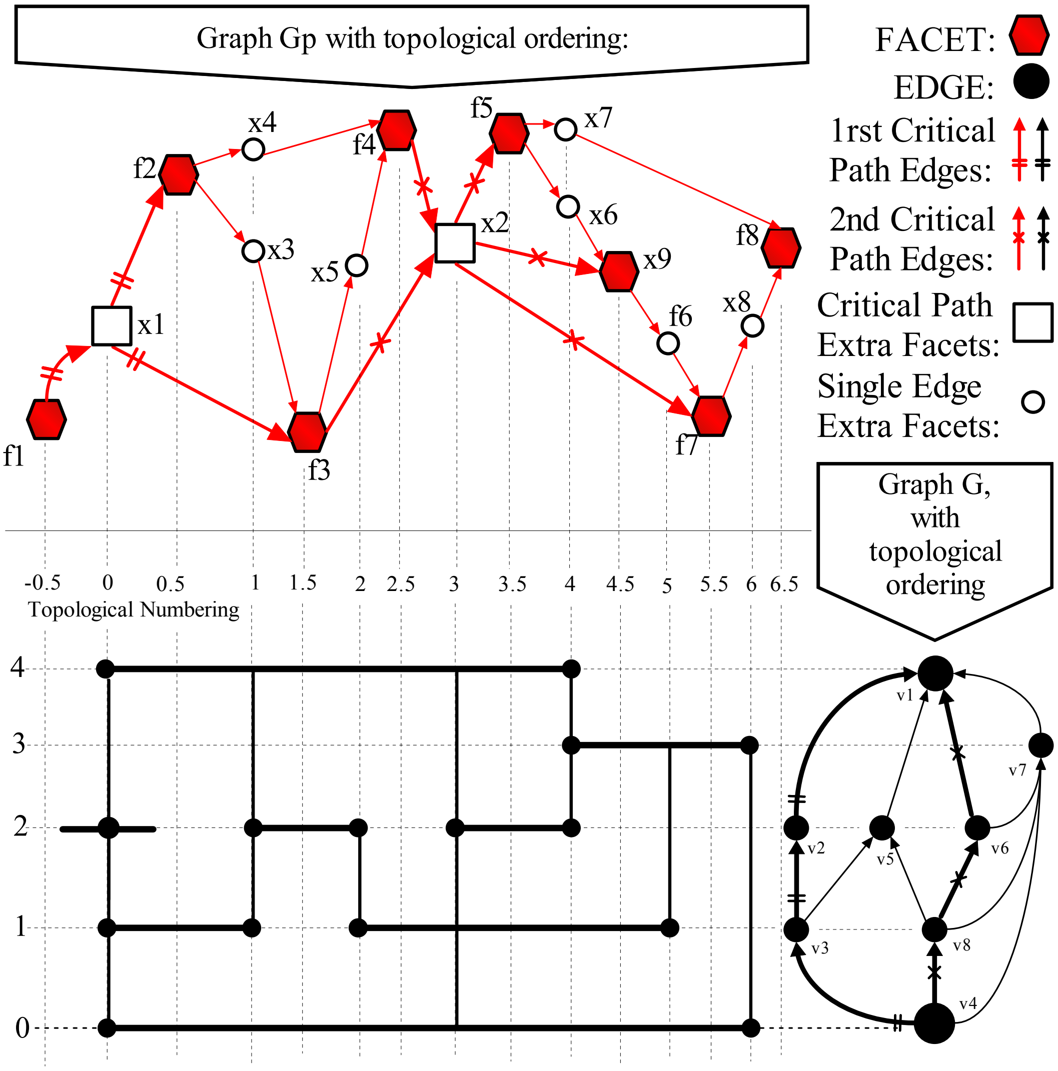


Figure 7.3: Constrained visibility representation of G.