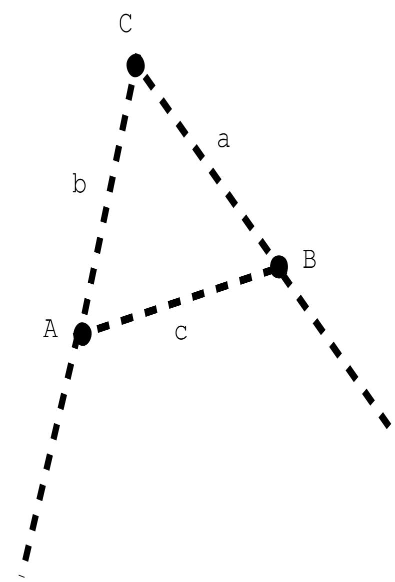

Figure 1: Part of a triangulation

1 Closest Obstacle Calculation

For our purposes, we must find the diameter of the larget circular unit that can move between two (unconstrained) edges of a triangle in a Constrained (Delaunay) Triangulation. For example, between edges a and b in Figure 1 above. Luckily this is equivalent to finding the closest obstacle to the vertex joining these two edges (vertex C in the diagram) in the region extending between the edges as shown. An obstacle can be a vertex or a point on a constrained edge.

There are three cases possible within a triangle which can determine the closest obstacle in this region. The first such case is that either angle at vertex A or at vertex B is a right angle or obtuse (Section 1.1), the second arises when these angles are accute and edge c is constrained (Section 1.2), and finally the last possibility is when angles A and B are accute and edge c is unconstrained (Section 1.3).

Pseudocode for the algorithm which determines the closest obstacle between edge a and b in triangle T is given in code listings 1, 2, and 3. A brief discussion on the complexity of this algorithm is given in Section 1.4, and the proofs that this technique is equivalent to finding the maximum radius of a unit with a valid path through this triangle are given in Section 2.

In these proofs, we assume circular units, but this can apply to other shapes as well. The maximum allowable size through a series of adjacent triangles yields a winding path though them and can be used to determine the

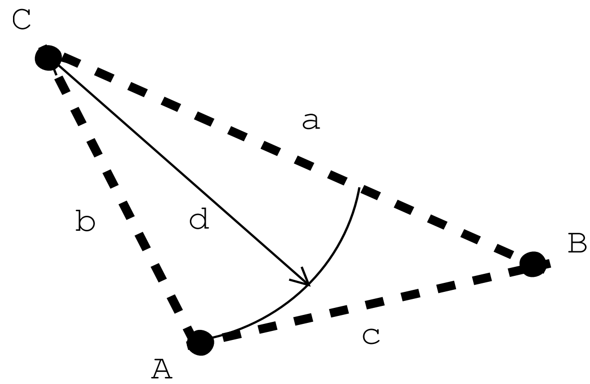

Figure 2: Case 1: angle at vertex A is obtuse

throughput of smaller units, or the maximum allowable size of a rectangular unit, for example.

The techniques used assume that at the very least, each vertex in a triangulation represents a contraint. That is, if one was representing an environment, one would only add a vertex to the triangulation either because it was an endpoint for a constrained edge, or if it were a point obstacle.

This is intuitive because adding unneeded vertices would only complicate the triangulation and slow down subsequent algorithms. If such vertices were added, however, these methods might incorrectly determine the maximum diameter of a unit through a path in the case where a path for the true largest possible unit would pass through such an unconstrained vertex.

In each case, the path for the unit of maximum diameter is determined as an arc hugging vertex C. While the unit need not always follow this path to traverse the triangle, it is true that a unit couldn't successfully traverse some other path and not this one, as is proven in Section 2.

1.1 Case 1: Angle A or B is Right or Obtuse

When the angle at either vertex A or B is right or obtuse, the first case applies. Assume, without loss of generality, that the angle at vertex A is right or obtuse. It follows that edge b is shorter than edge a. Thus the maximum allowable diameter d of a circular unit between edges a and b in this triangle is the length of edge a. See Figure 2 for a visual explanation.

This follows from the fact that the closest point on a line to a point is where a line passing through the point intersects the line at a right angle. Any other line intersecting both the point and the line is necessarily longer than this, as is shown below.

In Figure 3, the point c is the point where a line passing through point p intersects with line ab at a right angle. For any other point c' on the line ab,

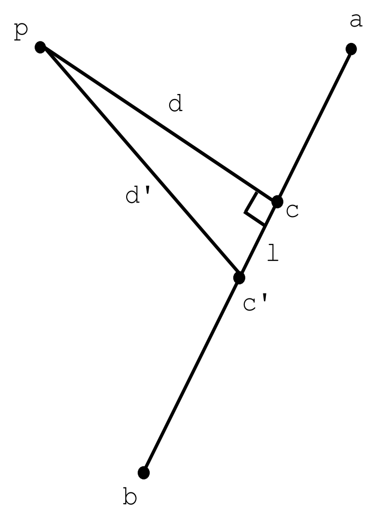

Figure 3: The closest distance of a line to a point

it will be some positive distance l away from c. Thus if the distance from p to c is d, then the distance (d') from p to c' is $\sqrt{d^2 + l^2} > d$. Thus c is the closest point to p on line ab, as desired.

Similarly, consider Figure 4. The length of segment b is $\sqrt{|CP|^2 + |PA|^2}$. For any point $A' \neq A$ along the segment between vertex A and vertex B, the length of the segment between C and A' would be $\sqrt{|CP|^2 + (|PA| + |AA'|)^2} > \sqrt{|CP|^2 + |PA|^2}$ since $|AA'| > 0$. Thus vertex A is the closest point on segment AB to vertex C, and since there can be no obstacles in this triangle (any obstacles would have been incorporated in the triangulation), it follows that the closest obstacle is $|b|$ away from vertex C and this is the maximum diameter of a unit that can traverse from edge a to edge b in this triangle.

1.2 Case 2: Edge c is Constrained

In the case that the interior angles at both vertices A and B are accute, the point on the line passing through the vertices A and B that is closest to vertex C lies between A and B as shown in Section 1.1 above. In the case that edge c is constrained, this point is an obstacle. This situation is shown in Figure 5.

As described in Section 1.1, since there can be no obstacles within the triangle, the closest point on edge c to vertex C (when edge c is constrained) represents the closest obstacle to vertex C in the triangle. Assuming the distance between vertex C and the point P on segment AB which makes

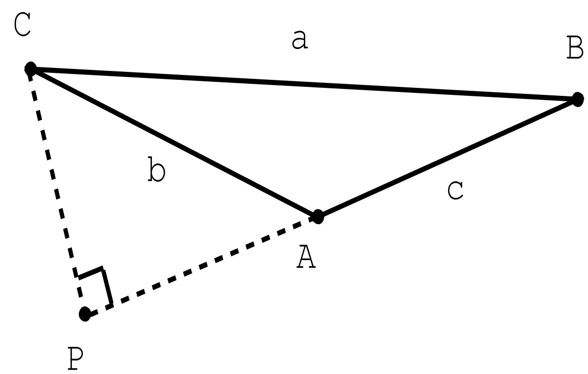

Figure 4: Triangle with one obtuse angle

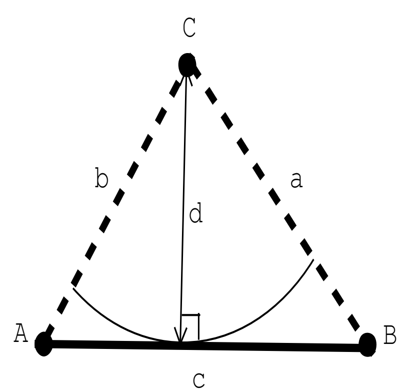

Figure 5: Case 2: angles at vertices A and B are accute and edge c is constrained

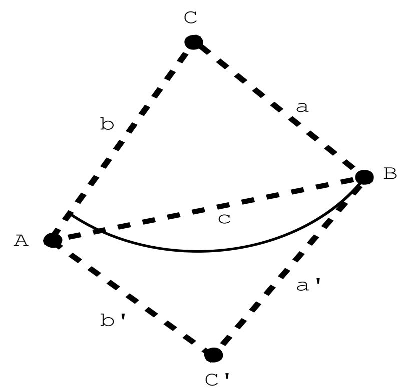

Figure 6: Vertex B is the closest obstacle to vertex C

CP perpendicular to AB, is d, the diameter of the largest circular unit that can traverse the triangle from edge a to edge b is d, as desired.

1.3 Case 3: Edge c is Unconstrained

In the case where edge c is not constrained and the angles at both vertices A and B are accute, the situation gets slightly more complex, as the closest point on segment AB to vertex C no longer represents an obstacle.

Vertices A and B are still obstacles, and thus the maximum unit diameter that can traverse this triangle from edge a to edge b is bounded above by both $|a|$ and $|b|$. Figure 6 shows a case where the shorter of edges a and b is the distance to the closest obstacle. However, since there may still be obstacles on the opposite side of edge c from vertex C closer to C than either A or B, we must consider these possibilities.

What must occur then is a search that is bounded by the closest obstacle found so far. Thus this search begins by searching across edge c to the triangle opposite this edge, bounded above by $min\{|a|, |b|\}$ and continues as described below.

When the search enters a triangle via an edge, it checks the other two edges as follows. We will say each edge is the segment between two vertices U and V. First of all, an edge will only be considered if both angles $\angle CVU$ and $\angle CUV$ are accute, that is, if the closest point on the line passing through U and V to vertex C lies between these two points. Figure 7 shows how considering such a segment could falsely determine the closest obstacle to vertex C. If this criterion is not met, search along this branch ends.

Next, we consider the distance from vertex C to the closest point on edge UV. If this distance is greater than the current upper bound, search along this branch returns because further search will not yield a closer obstacle than the closest already found. This situation is depicted in Figure 8. If this

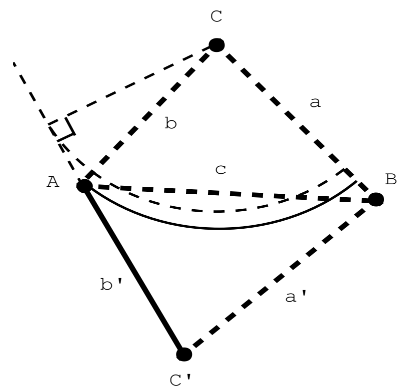

Figure 7: Edge b' should not be considered because $C'AC$ is obtuse

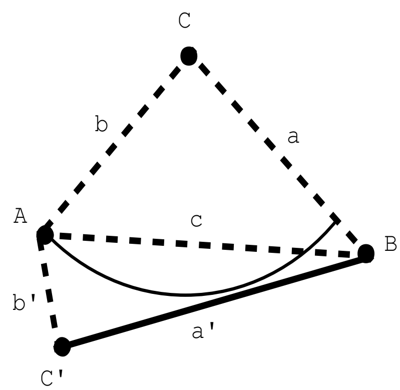

Figure 8: Edge a is farther from vertex C than A, so it is not considered

distance is less than the current upper bound and edge UV is constrained, then the current upper bound is updated to reflect this new distance, and search returns from this branch. A case where a constrained edge determines the closest obstacle to vertex C is shown in Figure 9. Finally if the distance is less than the current upper bound and UV is unconstrained, search continues across this edge.

1.4 Complexity

Of course, we desire to know that this algorithm's complexity will be reasonable if we wish to use it in certain domains. While a proof of the upper bound of this algorithm's running time would be quite involved, we can make a number of observations to see that it will not be unmanageable.

First of all, both cases 1 and 2 require constant running time. Case 1

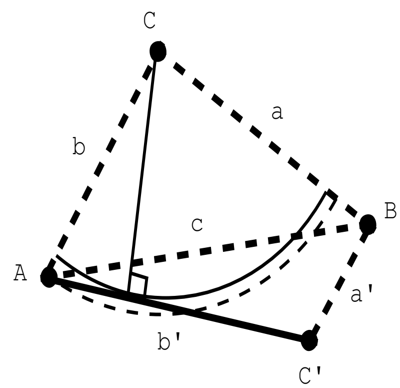

Figure 9: Edge b' becomes the closest obstacle to vertex C

Algorithm 1 DistanceBetween(Vertex C, Edge e)

A , $B \leftarrow$ EndpointsOf(e)
if $A_x = B_x$ **then**
  $a \leftarrow 1$
  $b \leftarrow 0$
  $c \leftarrow -(C_x)$
else
  $rise \leftarrow B_y - A_y$
  $run \leftarrow B_x - A_x$
  $intercept \leftarrow A_y - (\frac{rise}{run})A_x$
  $a \leftarrow rise$
  $b \leftarrow -run$
  $c \leftarrow run \times intercept$
end if
return $\frac{|a \cdot C_x + b \cdot C_y + c|}{\sqrt{a^2+b^2}}$

Algorithm 2 SearchWidth(Vertex C, Triangle T, Edge e, Distance d)

```
U, V ← EndpointsOf(e)
if IsObtuse(C, U, V) ∨ IsObtuse(C, V, U) then
  return d
end if
d′ ← DistanceBetween(C, e)
if d′ > d then
  return d
else if IsConstrained(e) then
  return d′
else
  T′ ← TriangleOpposite(T, e)
  e′, e″ ← OtherEdges(T′, e)
  d ← SearchWidth(C, T′, e′, d)
  return SearchWidth(C, T′, e″, d)
end if
```

Algorithm 3 CalculateWidth(Triangle T, Edge a, edge b)

```
C ← VertexBetween(a, b)
c ← EdgeOpposite(C, T)
A ← VertexOpposite(a, T)
B ← VertexOpposite(b, T)
d ← min{Length(a), Length(b)}
if IsObtuse(C, A, B) ∨ IsObtuse(C, B, A) then
  return d {Case: 1}
else if IsConstrained(c) then
  return DistanceBetween(C, c) {Case: 2}
else
  return SearchWidth(C, T, c, d) {Case: 3}
end if
```

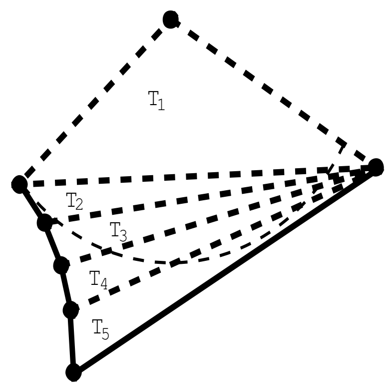

Figure 10: Searching across all triangles to find the width of triangle T_1

occurs quite frequently in triangulations, and when there are few or no point obstacles (triangulated vertices not representing endpoints of constrained edges) case 2 will occur even more often.

Also one can notice that unless vertex C is very obtuse, case 3 cannot result in a very long search. In most cases, the bound of the search ($min\{|a|, |b|\}$) will not exceed the value at the start of the search (the distance of edge c from vertex C) by very much. For a search to traverse many triangles, particularly when this difference is small, the triangles across edge c would have to be very thin. This is uncommon to see in most triangulations and a very rare case in Delaunay Triangulations. Such a case is illustrated in Figure 10.

Although searching a triangle can conceivably expand search across two other edges resulting in an exponential search, we must remember that in the worst case, the search is still limited by the number of triangles in the triangulation and thus could not be worse than linear. Next one must note the conditions under which search could branch in such a way. These conditions are rare to find in a triangulation, and nearly (if not completely) impossible in a Delaunay Triangulation.

As an example, observe Figure 11. In this case the algorithm for finding the width between edges a and b in triangle T will result in a search across edge c and into triangle T', and then across *both* edges a' and b'. However, in a Constrained *Delaunay* Triangulation, edge c would have been replaced by the edge shown in grey, and thus this effect would not have occurred.

Finally, although the worst case is linear in the number of triangles in the triangulation for searching a single triangle, observing the properties of the other triangles searched will reveal that they will almost or always fall under case 1 or 2 when searched. Because of this, the complexity of running this algorithm on *all* the triangles in the triangulation is likely linear and not quadratic.

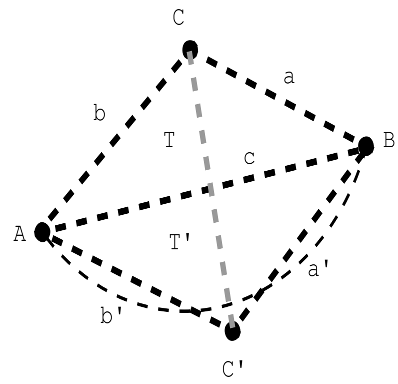

Figure 11: Finding the width of triangle T results in a branching search

This can be seen in Figure 10 where finding the width of triangle T_1 will result in a search through triangles T_2, T_3, T_4, and T_5. However, the algorithm run on all the other triangles will not result in any search at all. Thus the cost of finding the widths of all these triangles is linear in the number of triangles.

2 Maximum Unit Diameters

In this section, we show that the distance to the closest obstacle between edges a and b is equivalent to the diameter of the largest circular unit with a valid path between these edges. In Section 2.1, we define what constitutes a valid path and go through some preliminary proofs that we will use in the sections that follow.

In Section 2.2, we prove that whenever the closest obstacle between edges a and b is at distance d from vertex C then there exists a valid path between these edges for a circular unit of diameter d, thus the method is sound.

Then in Section 2.3, we prove that for every valid path for a unit of a given diameter, there will be no obstacles within this distance of vertex C between edges a and b, and thus the method is complete. These combined prove the equivalence of the distance between vertex C and the closest obstacle between edges a and b and the diameter of the largest unit with a valid path between these edges.

Finally, in Section 2.4 we show that the path found using this method given the radius of a unit is equivalent to the minimum length path between any points along these edges while avoiding obstacles. This will become a useful result later on.

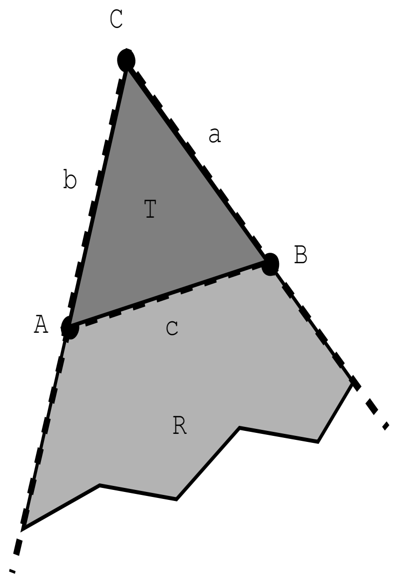

Figure 12: Region R for triangle T when moving between edges a and b

2.1 Definitions

Consider the region which exists between the rays extending from vertex C toward A and from C toward B. Call this region R. Figure 12 illustrates this region for a triangle. We will say there is a valid path from edge a to edge b for a circular unit of radius r in triangle T if and only if there exists a path p such that:

1. One end of p is on edge a
2. The other end of p is on edge b
3. p is contained entirely in region R
4. For each point on p, there is no obstacle (vertex or point on a constrained edge) which is closer than r
5. For each point on p, there exists a point within distance r which is inside triangle T

Point 5 above warrants some discussion. It is possible to find a path possessing all the above properties except 5, however because the unit completely leaves the triangle T, we consider the path to instead go from edge a to edge c, continuing somehow through other triangles, and then returning to triangle T going from edge c to edge b.

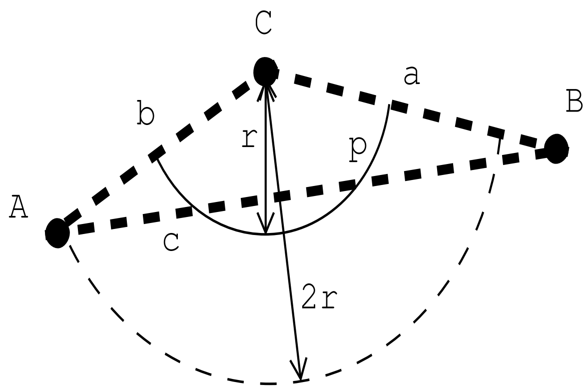

Figure 13: A valid path that is always within r of T

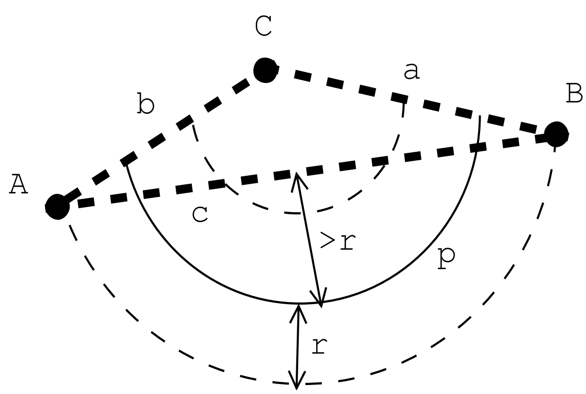

Figure 14: An invalid path that is $> r$ away from T at some points

The reason it is not required for the path to stay entirely within T is because sometimes a path might exist where the unit is always partially within T but the path itself might cross edge c. Suppose the triangle opposite edge c is T'. If we required the path to be entirely within T, such a path would require finding a path from a to c in T, from c to c in T', and then from c to b in T. This would complicate the problem unnecessarily and so such a path is only considered to be going through triangle T.

An example of a path p that leaves T but still satisfies this requirement is shown in Figure 13, and an example of a path that is invalid because it is $> r$ away from Triangle T at some points is shown in Figure 14.

Furthermore, we must make an exception to the rule for point 4. When the unit is crossing the edge a - that is, for points on the path for which some segment of length r extending from it intersects edge a - we do not consider obstacles in the region opposite edge a from region R. Similarly,

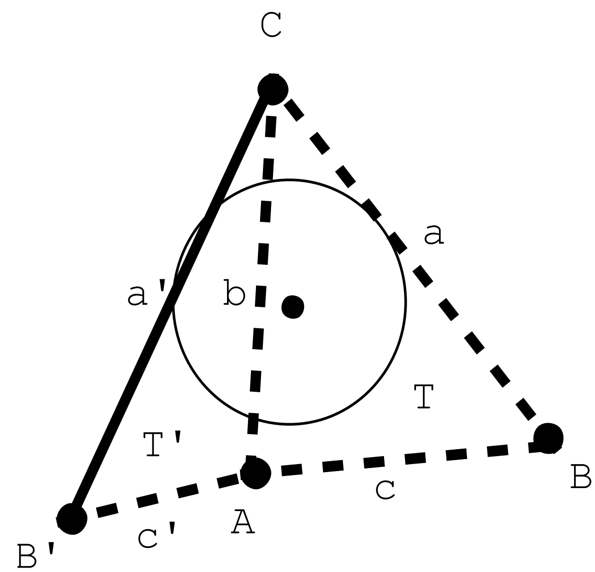

Figure 15: Obstacle outside of region R interfering with a path inside of it

when the unit crosses b, we do not consider obstacles in the region opposite edge b from R.

This is because the ultimate goal of finding these paths through individual triangles is to combine the paths through adjacent triangles together to form a single path through a triangulation. Thus obstacles on the opposite side of a will be the considered when finding a path through the triangle which shares edge a, and similarly for b. An example of such an obstacle interfering with a path outside of region R is given in Figure 15.

With that being said, sometimes an obstacle outside of region R can interfere with a path through T. If, at some point along a path, the unit is crossing the boundary of R outside either edge a or edge b, then those obstacles are not being considered in the paths through the triangles sharing those edges and thus must be considered by T.

The arc method, which states that the distance to the closest obstacle in region R from vertex C is the diameter of the largest circular unit with a valid path between edges a and b in triangle T, uses a special kind of path, which we call an *arc path*. Equivalently, we may refer to such a path as one which "hugs" vertex C.

An arc path for a unit of radius r is a path p such that each point on p is at distance r from vertex C. This forms an arc of radius r between edges a and b in region R. It also hugs vertex C by keeping the unit at the minimum distance at which requirement 4 is not violated by this vertex. Figure 16 shows an example of such a path.

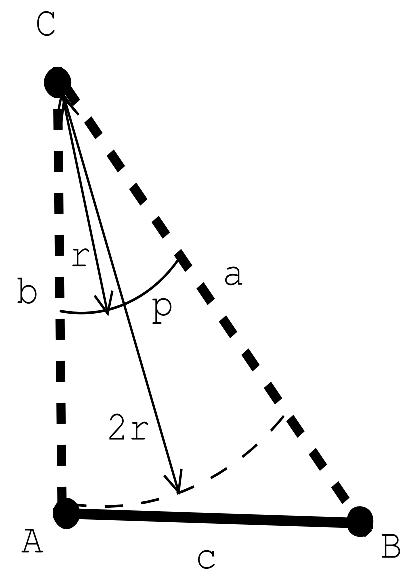

Figure 16: An example of an arc path

Arc paths are used because by their very definition, they satisfy requirements 1, 2, 3, and 5, leaving only requirement 4 to be verified. This final requirement for a unit of radius r is met for an arc path exactly when the closest obstacle to vertex C in region R is at distance $\geq 2r$ from vertex C. We prove that this method finds valid paths correctly wherever any are prenent in the sections that follow.

For the purpose of using it in the proof of the arc method's completeness in Section 2.3, we will now show that an arc path will not cross the boundary of R other than through edges a and b. We will prove this by contradiction, assuming we have a valid path which crosses this boundary and showing it is not an arc path.

For a path hugging vertex C to be valid, the arc around C of radius $2r$ in region R must be free of obstacles. Because vertices are considered to be obstacles, both edges a and b must have length at least $2r$ in order for this path to be valid.

Without loss of generality, we will simply consider the boundary of R by edge b. The proof extends identically to the boundary by edge a. Now, the unit must cross the boundary of R not at edge b, and not partially at edge b since that would imply the unit overlaps vertex A, which is an obstacle, and the path would violate requirement 4 and be invalid.

Thus the closest point, which we will call w, on that boundary of R to some point on the path, which we will call p, must be outside of edge b. Thus since the segment from w to p must be perpendicular to the boundary

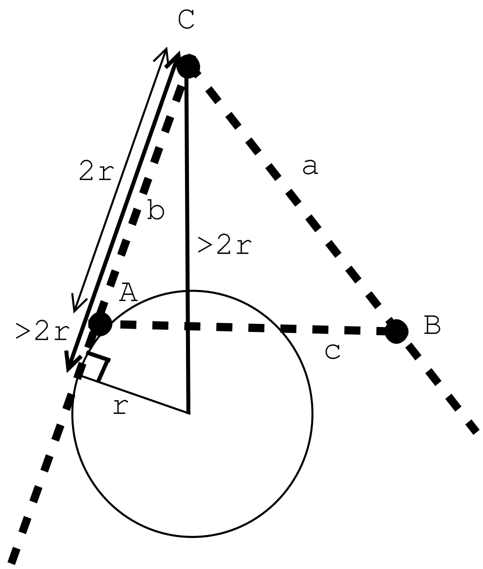

Figure 17: A unit crossing the boundary of region R below edge b

of R, then p must be farther from vertex C than w. Also we know that w is farther from vertex C than vertex A, and vertex A is at least $2r$ from vertex C. Thus p would have to be farther than $2r$ from vertex C, which violates our definition that an arc path always be at distance r from vertex C. A diagram of this proof is found in Figure 17.

Thus an arc path will not cross a boundary of region R outside of edges a and b, as desired. By corollary, to verify requirement 4 of an arc path, we need only consider obstacles within region R.

2.2 Soundness

Here, we will prove that if there is no obstacle within $2r$ of vertex C in region R, then there is a valid path through triangle T from edge a to edge b. In particular, we will prove that there is such a path hugging vertex C because by the very definition of these arc paths, this already satisfies requirements 1, 2, 3, and 5 in Section 2.1 above.

It remains to prove that if no obstacle is within distance $2r$ from vertex C - that is, there is no obstacle inside the arc of radius $2r$ around vertex C in region R - then the arc path satisfies requirement 4. This we will prove below.

Since the unit travelling along the path hugging vertex C has radius r, every point on this path will be at distance r from C, by the definition of these arc paths. For requirement 4 not to be met, there must be an obstacle

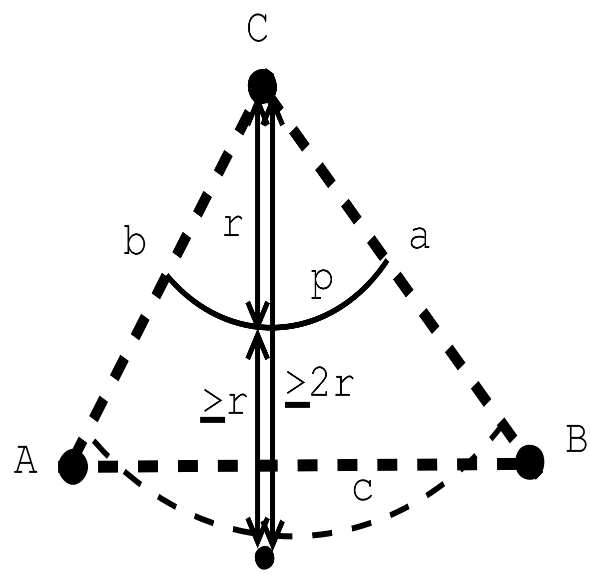

Figure 18: Using the triangle inequality to prove soundness

within distance of less than r from one of these points.

However, because of the triangle inequality, we know that an obstacle that is $\geq 2r$ from vertex C cannot be $< r$ from any point that is at distance r from that same point. This is depicted in Figure 18 for clarity. Thus, if there are no obstacles within distance $2r$ of vertex C in region R, there are none within distance r of any point along the path from edge a to b in triangle T hugging vertex C. Therefore, requirement 4 is met and the path is valid, and finally, the arc method is sound.

2.3 Completeness

In order to prove that if there exists a path from edge a to edge b in the triangle T for a unit of radius r then there are no obstacles within distance $2r$ of vertex C in region R, we will prove the contrapositive. Assume there is an obstacle at distance $< 2r$ from vertex C in region R.

We will consider the point p to be the closest obstacle to vertex C in region R. Such a point must exist in region R because by definition (Section 2.1), only obstacles in region R can interfere with an arc path from edge a to edge b through triangle T.

Now consider the line segment going from vertex C to this point p. We know that the length of this line segment is $< 2r$. For the remainder of the proof, we will consider this point to be the only obstacle in region R. This is a relaxed constraint, and certainly, if no path exists with this single point

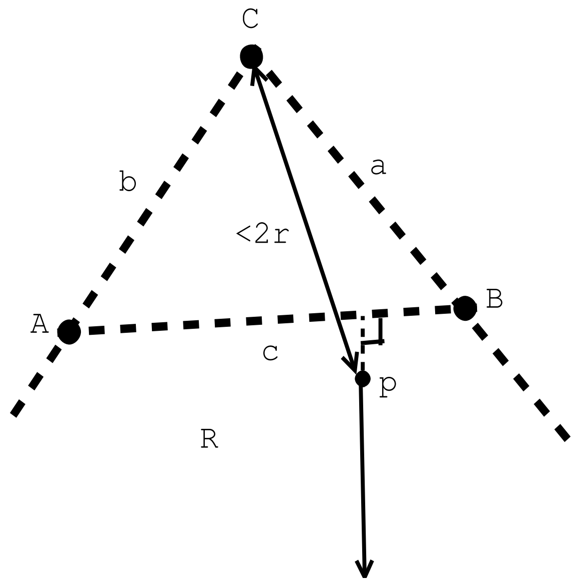

Figure 19: Partitioning of region R into 2 sub-regions

obstacle, no path exists with any amount of obstacles including p.

Next, consider a segment between vertex C and p, and a ray extending from p perpendicular to and away from edge c of the triangle. One can see that these partition R into two sub-regions. This partitioning can be observed in Figure 19. Thus, we can see that any valid path travelling from edge a to edge b in region R must cross either the segment between C and p, or the ray extending from p, since edge a and edge b are in different sub-regions of R and a valid path must be entirely within R by requirement 3. We will cover both of these cases below.

First, assume the path of a unit with radius r crosses the segment between C and p at some point u. Since the segment between C and p has length $< 2r$, we know u must be at distance $< r$ from either vertex C or p (or both). Thus if the path crosses any point on this segment, it comes within distance r of some obstacle, and thus fails to meet requirement 4 above. This can be seen in Figure 20 where the above unit is interfering with vertex C (its centre is distance $< r$ from C) and the lower unit with point p.

Similarly, if the path crosses the ray extending from p perpendicular and away from edge c, it must cross at some point v on the ray that is at least distance r from p, otherwise it would fail to meet requirement 4 and would be an invalid path. Figure 21 shows that a unit crossing this ray is indeed at distance $> r$ from triangle T. We also use that there cannot be an obstacle within a triangle in a triangulation: obstacles are either represented

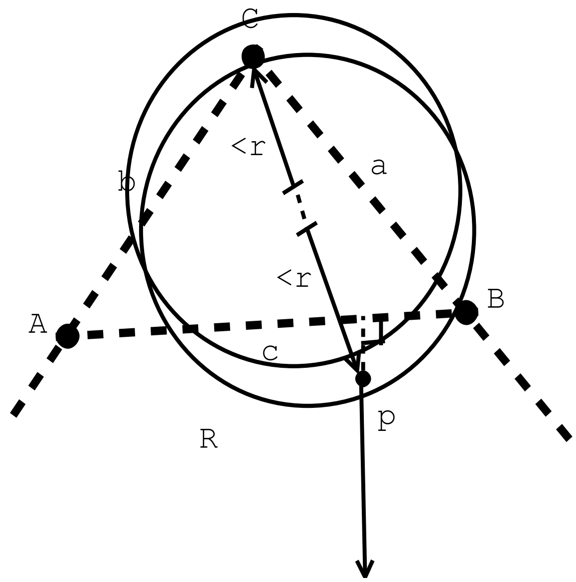

Figure 20: A unit trying to pass between C and p

as vertices or edges of disjoint triangles.

Thus the point v where the path crosses the ray must be at least r away from p. Since the ray is perpendicular to c - the closest edge of T to p - the point that is distance r away from v that is closest to triangle T is along that ray. Since v is at least r away from p, which itself is outside of T, it follows that at point v, there is no point within distance r which is inside triangle T. Thus requirement 5 is violated and the path is invalid.

Finally we conclude that since there is nowhere that a path from edge a can cross the partition to edge b and remain valid, all such paths are invalid. Thus if a valid path exists for moving a circular unit of radius r from edge a to edge b in triangle T, there must be no obstacles within distance $2r$ of vertex C in region R, as desired.

2.4 Minimal Length

Here we will prove that the arc path with radius r between two edges in a triangle is actually the shortest valid path for a unit of that radius between those two edges in the triangle. This result will become useful when we begin searching for paths in Constrained (Delaunay) Triangulations.

We know that a unit of radius r cannot come within r of a vertex without invalidating its path (as we assume all vertices in a triangulation represent constraints). This means that any valid path must have all points at least at distance r from vertex C (again when considering paths between edge a

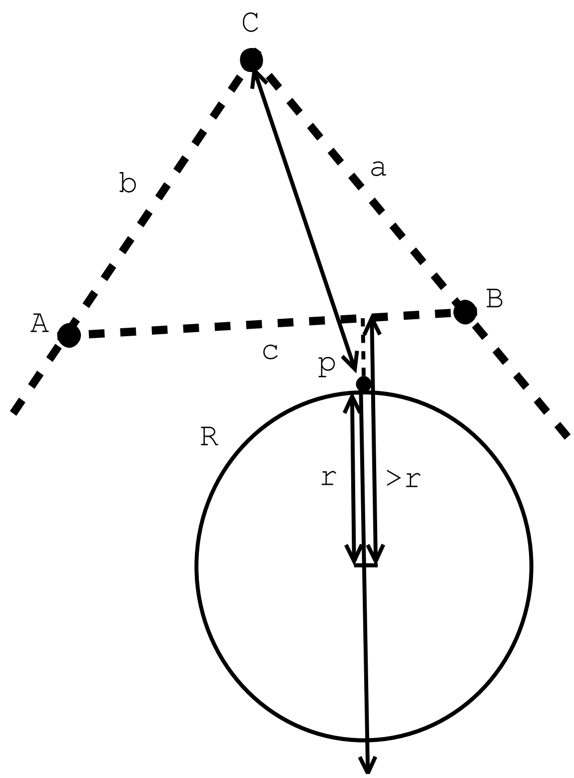

Figure 21: A unit trying to pass below point p

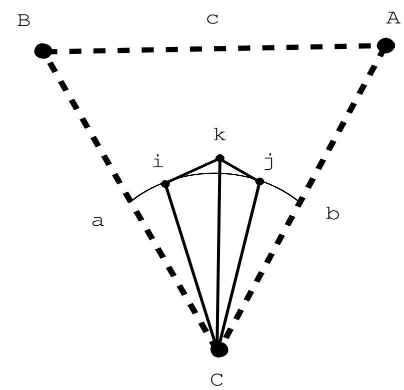

Figure 22: An alternate path departing from the arc path in the middle

and edge b).

The paths where all points are exactly distance r from vertex C is indeed the arc path with radius r, so it remains to prove this path is shorter than paths with points $> r$ away from vertex C. This we will prove below.

There are two cases to consider for this proof: one where the alternate path meets edges a and b at the same points as the arc path (distance r away from vertex C), and one where the alternate path meets these edges at different points. We will consider the former possibility first.

If the path touches both edges a and b at the same place as the arc path and departs from it at some point in between, we know that the alternate path must depart from the arc path at some point which we will call i and rejoin it at some point which we will call j, with some point not on the arc path (at distance $> r$ from vertex C) in between, which we will call k. This configuration is shown in Figure 22.

Consider a path that follows the arc path to point i, goes in a straight line to point k, and then to point j, and continues following the arc path through the triangle. Because the shortest distance between any two points is a straight line, we know that this path is at least as short as any other path through these points. It remains to prove that the arc path between points i and j is shorter than the path from i to k to j. This we prove below.

Take first the triangle formed by vertex C, and points i and k as shown in Figure 23. Consider the angle $\angle iCk$ to be θ. We know the arc in this triangle has length $r\theta$ so we must show segment ik has length $> r\theta$. We know the segment Ci has length r and segment Ck has length $> r$, we must find the length of segment ik.

We know that $\angle Cik$ is either a right angle or obtuse, because otherwise segment ik would be closer than r to vertex C at some points. Also, the more obtuse this angle is, the longer the segment ik would have to be given we know the length of segment Ci and angle θ. Thus we will assume $\angle Cik$

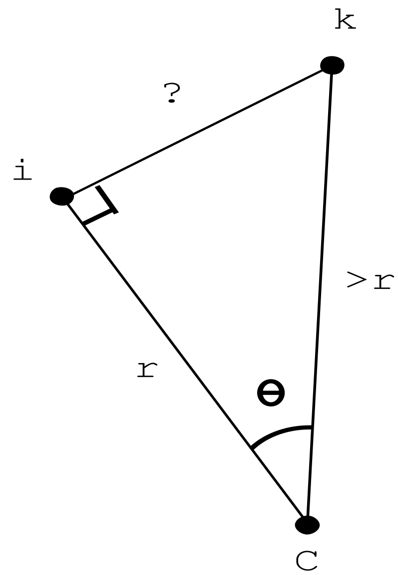

Figure 23: One section of the alternate path

is a right angle, and the real length of segment ik will be at least that which we find using this assumption.

We can calculate the length of segment ik to be $r \cdot tan(\theta)$, so if this section of the arc is shorter than segment ik, then we have $r \cdot tan(\theta) \geq r\theta$ or $tan(\theta) \geq \theta$, which is true for $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$, which is all we require. Thus segment ik is longer than that portion of the arc, and this applies identically to segment kj as well. We have shown, then, that the arc path is the shortest of valid paths through the triangle that pass through points at distance r away from vertex C on edges a and b.

The proof concerning paths that do not meet both edges a and b at distance r from vertex C, follows similarly. Consider one side where the alternate path and the arc path do not meet an edge at the same place. Without loss of generality, assume this is edge a. We will also assume that the arc path and the alternate path have at least one common point. The extension of this proof to alternate paths that do not meet the arc path at all will be given afterward.

Let the closest such common point to edge a be point i and the point on edge a that this alternate path meets be point k. This is shown in Figure 24 Assume now that the alternate path between i and k is a straight line because any path between i and k must be at least as long as the straight-line path between them. Now the proof that segment ik is longer than the corresponding section of the arc path is identical to the proof above.

Now we will show that this extends to alternate paths that do not share

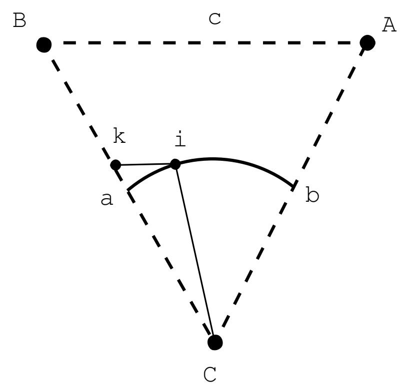

Figure 24: Alternate path departing from the arc path on one side

any common points with the arc path. For a path that does not touch the arc path at any point, consider an arc path of minimum radius r' that does touch this path. We know $r' > r$ because if $r' < r$ then part of this path would be at distance $< r$ from vertex C and would thus be invalid.

From the proof above, the length of this alternate path is greater than that of the arc path with radius r'. Since this path is longer than the arc path of radius r ($r' > r \Rightarrow r'\theta > r\theta$ where $\theta = \angle ACB$), we have that the alternate path is longer than the arc path with radius r.

Thus we have shown that the arc path of radius r between two edges of a triangle is the shortest valid path for a circular unit of radius r between those edges of that triangle, as desired.

3 Combining Triangles into a Path

As stated in Section 2, the purpose of finding the diameter of the largest circular unit that can traverse a triangle from one edge to another is so we can determine the diameter of the largest circular unit that can traverse a path of adjacent triangles. We can also use this knowledge to guide the search for a path for a circular unit of a certain diameter. In this section, we will give several results relevant to this goal.