6.890 Lecture 21 - Scribe Notes

Aviv Adler, Neil Gurram, and Andrea Lincoln

November 24, 2014

1 3SUM motivation

Throughout this class, we have categorized problems based on whether they were solvable in polynomial time or not. However, it would also be nice to know whether certain polynomial problems are $\omega(n^{c-\epsilon})$, and for which there are algorithms known to solve the problem in $O(n^c)$, for some constant c and any given $\epsilon > 0$.

So, let us consider the following problem. Given n integers, we want to know if any 3 sum to 0. This problem is called $3SUM$. [4] There are two versions of this problem depending on whether the three integers are allowed to be duplicates. This is easily solvable in $O(n^3)$ by trying all triples.

We can also do this in $O(n^2)$ randomized time. Indeed, we can first compute all the pairwise sums and then compute all the pairwise sums and create a hash table of these values; then, we check over all integers if its negation is in the list of integers.

But we can do better; there exists an $O(n^2)$ deterministic algorithm. Start with two copies of the integers in sorted order, which takes $n \log n$ time. Then, in one array we have a pointer at the front of the list, and the other array has a pointer at the end of the list. We do the following for each the n integers in the set. If the sum of the integers at the two pointers and the current integer is smaller than 0, we move the first array's pointer forward; if the sum is larger than 0, we move the second array's point backwards; otherwise, we have the three integers sum to 0, and we are done. If the two pointers crossover, we move onto the next integer. This algorithm clearly takes $O(n^2)$ time.

It is conjectured that no $O(n^{2-\epsilon})$ algorithm exists for this problem, for any $\epsilon > 0$. This is called the $3SUM$-Conjecture.

In various models of computation or with a more limited problem scope, we have algorithms that run in subquadratic time.

If we have that all of the integers are in the range $[-u, u]$, we can solve this problem in $O(n + u \log u)$ time. We have a bunch of subquadratic algorithms as well, but not by factor of ϵ. We can use a model of computation called word RAM to manipulate $\log n$ words in constant time, and it will take total time $O(n^2/(\frac{\log_n}{\log \log n}))^2)$; this is randomized. [2]

This year in fact, we were able to show that there is a randomized algorithm with running time $O(n^2/\frac{\log n}{\log \log n})$ and we have a deterministic algorithm using real RAM that can solve it in $O(n^2/(\frac{\log n}{\log \log n})^{\frac{2}{3}})$. [5]

We also have a decision tree model in which it takes $O(n^{1.5}\sqrt{\log n})$ time to run the model [5]; this is not an algorithm, as we don't know how to compute the tree in subquadratic time. This is one of many problems where we have a better decision tree than algorithms.

1.1 k-SUM and Its Relation to NP

Our 3-SUM problem can easily be generalized to k-SUM, which asks, given n integers, do any k of them sum to 0. 3-SUM is the most popular type of these problems.

In addition, k-SUM is NP-complete. Namely, this is essentially asking the Partition Problem, which is known to be NP-complete. In fact, it is $W[1]$-hard with respect to k. Furthermore, if we assume the exponential time hypothesis, we have there is no $n^{o(k)}$ algorithm, where we are assuming $k \leq n^{0.99}$. [7] Finally, we have that there is a randomized algorithm of $O(n^{\lceil \frac{k}{2} \rceil})$.

We then have the following k-SUM conjecture: there is no $O(n^{\lceil \frac{k}{2} \rceil - \epsilon})$ algorithm for solving this problem. This reduces to the normal 3-SUM conjecture for $k = 3$.

2 3SUM-Hardness

2.1 Definition

This leads us to the notion of $3SUM$-Hardness. We call a problem $3SUM$-hard, if there exists an $O(n^{2-\epsilon})$ algorithm that can solve the problem, then there exists one for $3SUM$ as well.

2.2 3SUM Reductions

This leads us to the notion of $3SUM$-Reductions. Recall we have if A reduces to B, we can solve A using B. So, suppose we reduce an instance x

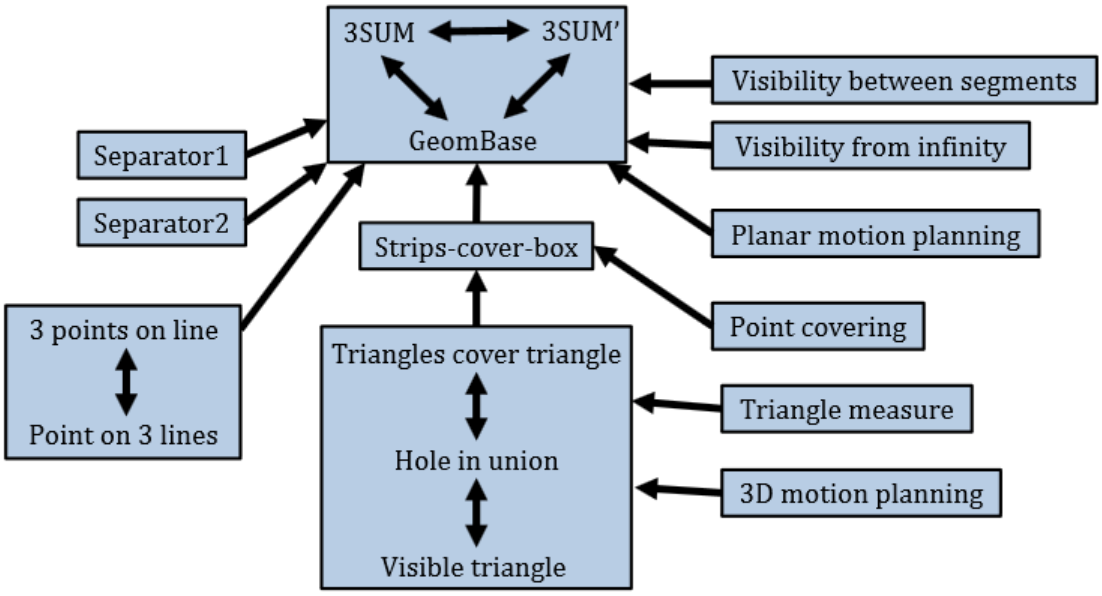

Figure 1: The relationships of problems to 3SUM.

in A to x' in B. If $n = |x|$, and $n' = |x'|$, then if we were to have a $3SUM$ reduction, it is required that we make a $O(1)$-call reduction on $n' = O(n)$, and that our reduction is $O(n^{2-\epsilon})$ for some $\epsilon > 0$.

So, if there is a $3SUM$ reduction from A to B, then we have if A is 3SUM-hard (such as 3SUM), then B is also 3SUM-hard.

3 3SUM-hard Problems

3.1 3SUM-Variants

We have that $3SUM$ with $u = O(n^3)$ is still $3SUM$-hard. This shows we don't need to have the numbers being too large. [6] [2]

$3SUM'$ is the following: given 3 sets A, B, C of n integers, are there $a \in A$, $b \in B$, and $c \in C$ such that $a + b = c$. This problem is also $3SUM$ hard. This is easy to see by a reduction from $3SUM$ with the original instance being S; just put $A = S$, $B = S$, $C = -S$. [4]

We can also show how to reduce from $3SUM'$ to $3SUM$ using very large integers [4]; namely, we can make S to be all the elements of $A +$ *large*,

$B+2$ *large*, and $C-3$ *large*, where *large* just means a large number that is added to each element of the corresponding sets.

3.2 Computational Geometry

The paper by Gajentaan and Overmars has lots of problems in computational geometry that are reducible from $3SUM$.

3.2.1 GeomBase

Our first problem is called GeomBase, which asks the following: given n points in $2D$ with $y \in \{0, 1, 2\}$, does there exist a non-horizontal line hitting 3 points of this set. [4] This is displayed in Figure 2.

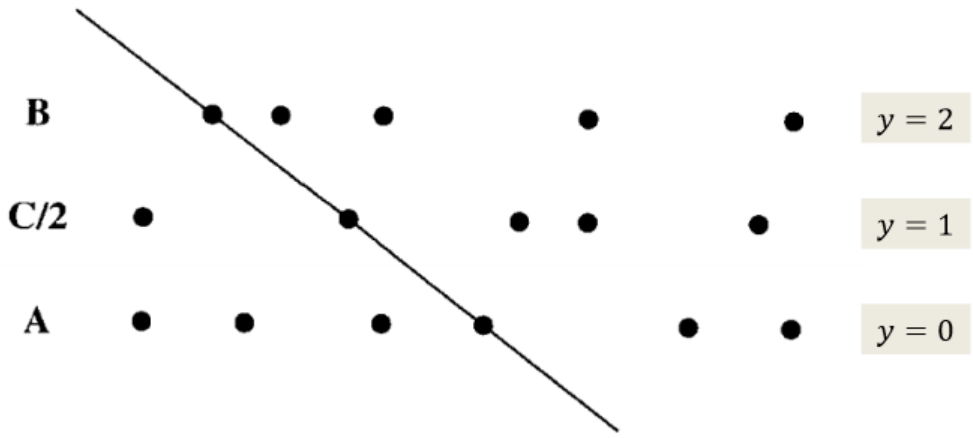

Figure 2: The problem of GeomBase.

We can prove this problem is $3SUM$-hard by reducing from $3SUM'$. Namely, for $a \in A$, we consider the point $(a, 0)$; $b \in B$, the point $(b, 2)$; and $c \in C$, $(\frac{c}{2}, 1)$. Then, it is easy to show that three points lie on a non-horizontal line if and only if there exists $a \in A$, $b \in B$, and $c \in C$, such that $a + b = c$. In fact, this reduction can be used in reverse to show that GeomBase reduces to $3SUM'$. We may have that the points may not be lattice points, but we can always scale the x-coordinates to become integers, and then we can reduce to $3SUM'$.

As an important step to show problems later in this lecture are $3SUM$-hard, we are going to consider a transformation from a GeomBase Instance.

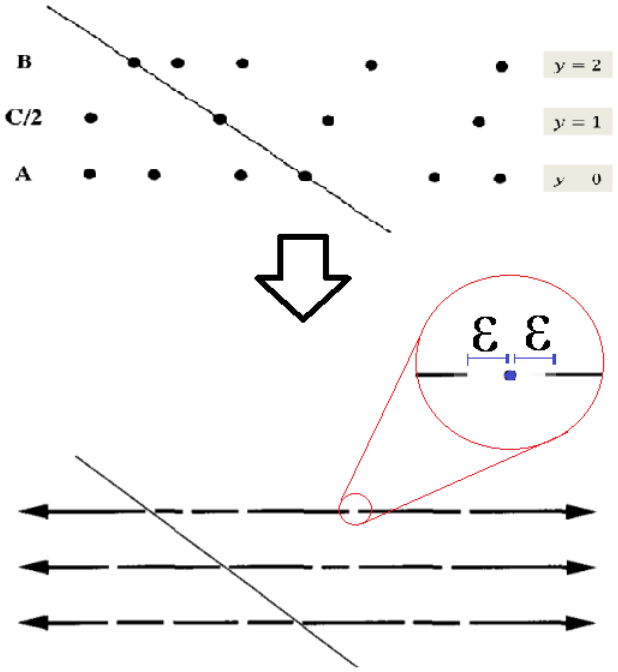

Figure 3: The transfomation to GeomBase'.

We will first construct the three lines $y = 0$, $y = 1$, and $y = 2$. Then, for every point on the lines in the original instance, we draw an ϵ-neighborhood for each of those points that get removed from the horizontal lines. These give a notion of holes. Then, the half-infinite rays can be replaced by finite segments that are bookended by vertical segments intended to block horizontal separating lines. We call this new instance an instance of GeomBase'. This is displayed in Figure 3.

3.2.2 Collinearity

We now begin a shift towards Incidence Problems.

We have a problem of asking whether there exist 3 points on a line, no two of which coincide. [4] This is $3SUM$-hard. We can make a reduction from $3SUM$ to this problem. As displayed in Figure 4, we map from x to (x, x^3) for each x in the $3SUM$ instance, where we assume the three chosen numbers have to be distinct. We can show that three points on the curve will lie on a line if and only if there are three integers summing to 0 in the

original set.

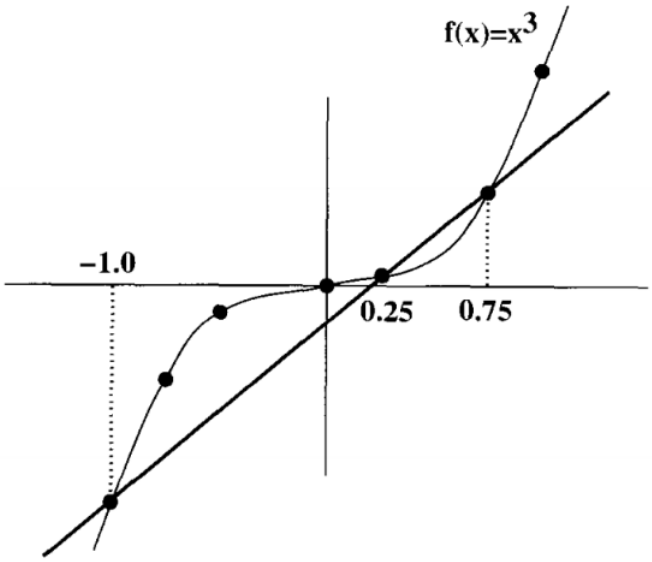

Figure 4: The reduction from 3SAT to collinearity.

Indeed, notice that

$$\frac{b^3-a^3}{b-a} = \frac{c^3-a^3}{c-a} \iff b^2+ba+a^2 = c^2+ca+a^2 \iff (b-c)(b+c+a) = 0 \iff b+c+a = 0,$$

where in the last equality we assume that the three numbers are distinct.

3.2.3 Concurrency

We also can consider the following problem: given n lines, we ask whether any point is on three of the lines. [4] This is the dual of the previous problem.

We will show this problem is $3SUM$-Hard by reducing from Collinearity. Indeed, every point (a, b) can be mapped to the line $ax + by + 1 = 0$. This is called projective plane duality. Then, this preserves point/line incidence; if three points were collinear, the three corresponding lines are incident, and vice versa. Therefore, we have a $3SUM$-reduction.

Incidentally, we can also consider the reverse reduction. If we have any lines that pass through the origin, we can translate all the lines so that none of them pass through the origin. Then, we can consider the projective plane duality defined above, and we have a reduction.

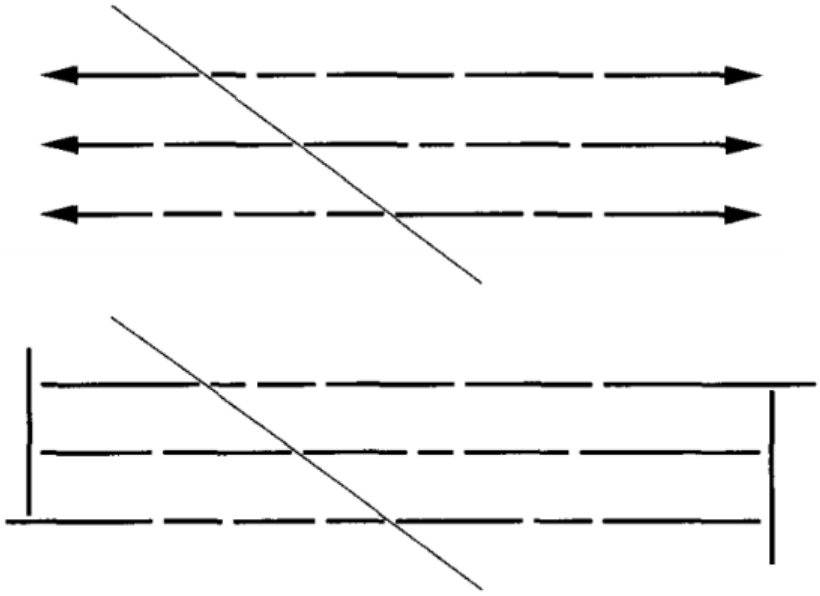

Figure 5: The reduction from GeomBase to Separator, using GeomBase'.

As a side note, all the the d dimensional versions of the problems mentioned so far, are $d+1$-sum hard.

3.2.4 Separator

We can now consider another type of problem. Suppose we are given n line segments in the plane. We ask if there is a line that separates them into 2 nonempty groups. This line is not allowed to intersect any of the segments. [4] This is called the Separator Problem.

We can reduce from GeomBase to Separator. Consider the corresponding GeomBase' Instance. We have we can find a nonhorizontal line hitting 3 points if and only if there is a line separating the corresponding segments into three points (the segments are formed by complement of original graph). This is displayed in Figure 5.

3.2.5 Strips Cover Box

We are now on course with Covering Problems.

We have the following question. Does a union of n strips cover a given

axis-aligned rectangle? A strip is just a fixed region in between two parallel lines. [4]

We can reduce from GeomBase to Strips Cover Box. Consider the corresponding instance of GeomBase', and rotate it 90 degrees. Then, consider the following duality: take the point (m, b) and map it to the line $y = mx + b$. The reverse construction is possible for all non-vertical lines. This is described in Figure 6

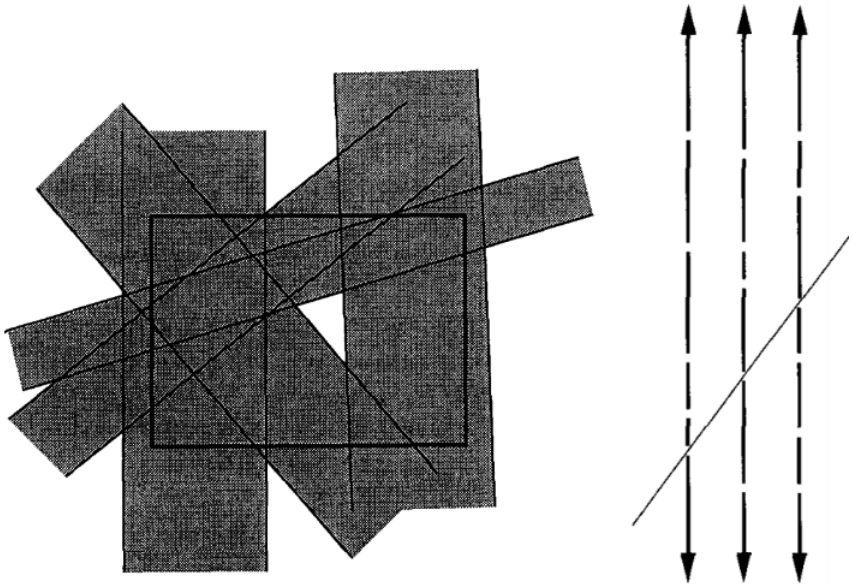

Figure 6: The reduction from GeomBase to Strips Cover Box, using GeomBase'.

Then, observe that points on a a single vertical segment, will get mapped into a strip of lines. Further, we can get the box by considering the 6 halfplanes that occur because of the rays, and then consider the bounding box of the hole formed by the union of these 6 regions Then, we restrict half planes to this rectangle by adding 6 more strips. We then have that if there is a point left behind by these strips, then GeomBase was in the negative.

3.2.6 Triangles Cover Triangle

We can also do the same type of question with Triangles. Do a bunch of triangles cover a triangle? [4] This reduces from Strips Cover Box. We start with a box. Make a triangle that covers the box, and then triangulate

the exterior of the rectangle, but interior of triangle. Then, for each strip, triangulate each intersection. Then, we have the question do a bunch of triangles cover a triangle. We need to make sure we don't blow up complexity when we triangulate regions of strips, but this is fine as triangulation results in $O(1)$-gons.

3.2.7 Hole in Union

We have another problem called Hole In Union. This problem asks does a union of n triangles have a hole? [4] We can reduce from the previous problem of Triangles Cover Triangle. Indeed, let us add very thin triangles that cover the edges of the original triangle instance. We then have that there is a hole if and only if the triangles do not cover the triangle.

3.2.8 Triangle Measure

We have yet another question regarding triangles. Given a set of triangles in the plane, compute the measure of their union. [4] We have a reduction from Triangles Cover Triangle. First, we can add extra triangles to cover the edges of the original triangle instance. Then, we have that the area of the union of the triangles is equal to the area of the original triangle if and only if the triangles cover all the triangles.

3.2.9 Point-Covering

The Point-Covering problem asks given n half-planes, is there a k-way intersection, as in is there a point covered by at least k of the half-planes. [4]

Observe if $k \leq \frac{n}{2}$, we have that the answer is going to always be yes.

But for larger values of k, this becomes harder. So, consider the reduction from Strips-Cover-Box. For each strip, take the 2 half-planes that do not cover the strip. Then, we have any point that does not lie in any of the strips lies in exactly n half-planes. Any other point will lie in fewer half-planes. Then, add 4 more half-planes that are directed inwards into the rectangle with each line overlapping with a rectangle edge. Then, if there is a point that is covered by exactly $n+4$ of the $2n+4$ half-planes, we know the Strips will not cover the Box, and thus we have a reduction.

3.2.10 Visibility Between Segments

We now shift towards Visibility Problems. Our first problem deals with Visibility Between Segments. [4] Given a set of n horizontal line segments in the plane, and two particular segments s_1 and s_2. We ask whether there are points on s_1 and s_2 that can see each other; in other words, is there a segment that, aside from its endpoints on s_1 and s_2 does not intersect any of the n horizontal line segments. This is displayed in Figure 7.

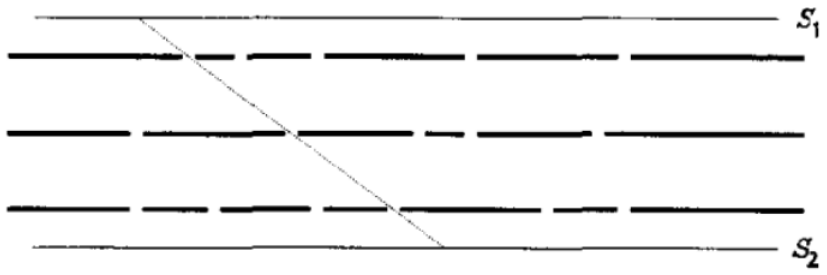

Figure 7: Segment visibility reduction using GeomBase'.

We reduce from GeomBase to Visibility-Between-Segments. We consider the corresponding GeomBase' Instance. Then, add two segments, one above the segments and the other below. These two segments can only see each other if and only if there are three collinear ϵ-neighborhoods corresponding to three collinear points in the original instance.

3.2.11 Visible Triangle

We can then consider a three-dimensional version of the previous problem with triangles. Given a set of n horizontal triangles in $3D$, and one special triangle, T, and a given point in $3D$-space, can we see a point on triangle T. This assumes that the n horizontal triangles are not transparent. [4]

There is a reduction from Triangles Cover Triangle to this problem. Indeed, we first can suppose T has z-coordinate 0, and then make all the other triangles have different heights above T. Then, we can let the point be the point of infinity, and we have that T is visible from infinity if and only if the triangles do not cover T. This is depicted in Figure 8.

We also have a reverse reduction for this problem from this problem to Triangles Cover Triangle.

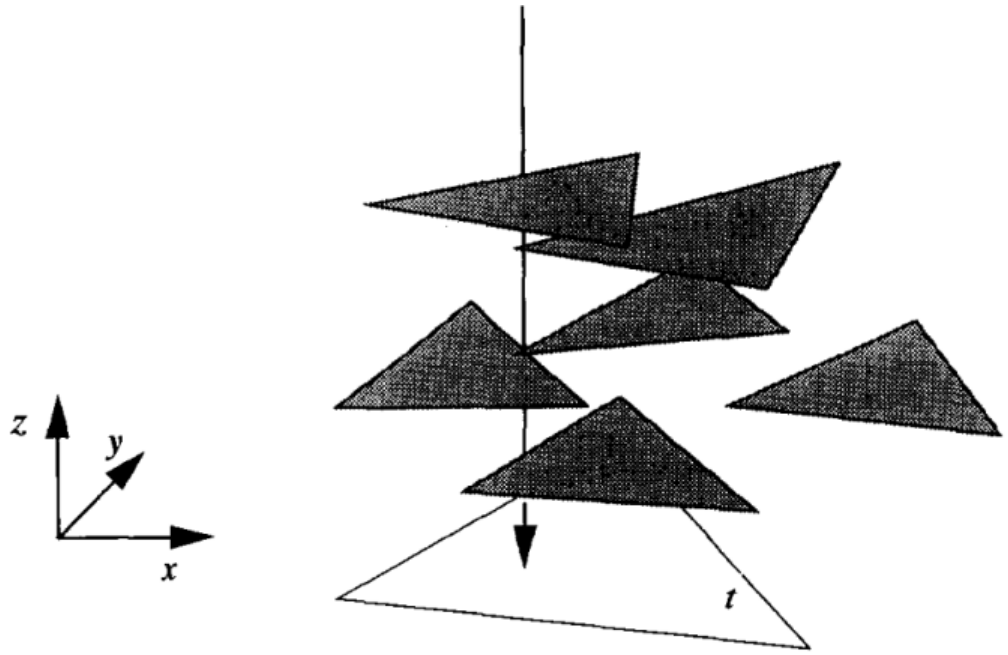

Figure 8: Visible Triangle reduction.

3.2.12 Planar Motion Planning

We also consider a group of Motion Planning Problems. The first problem is called Planar Motion Planning, which asks given given horizontal and vertical segment obstacles, can we move a robot (represented in the form of a line segment) allowing translations and rotations, from a given source to a goal without colliding into any obstacles. [4]

The reduction is evident from the Figure 9.

3.2.13 3D Motion Planning

We can then extend the previous problem into $3D$-space, to get $3D$ Motion Planning. It asks given a set of horizontal non-intersecting triangle obstacles in $3D$-space, and a robot represented as a vertical line segment, can the robot move through the obstacles without collision, using translations only. [4]

There is an algorithm to solve this problem in $O(n^2 \log n) time$.

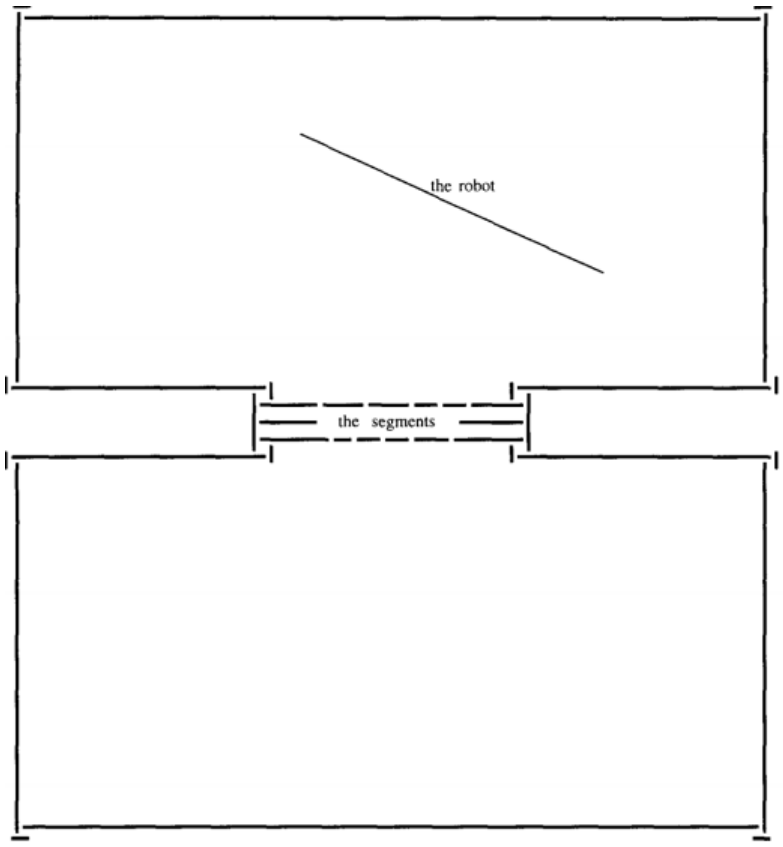

Figure 9: Planar Motion Planning.

This problem can be reduced from Triangles Cover Triangles. Indeed, first we create a cage to prevent the robot from leaving the original triangle T in the original problem instance. Then, we see that we can go from a given source from the top of the cage to the bottom of the cage, if and only if there is a point not covered by a triangle, and we are done. This is depicted in Figure 10.

3.2.14 Fixed Angle Chains

A *fixed angle chain* is a chain of line segments which follow each other at fixed angles.[3]We can imagine this in 3 dimensions as an object where the segments are attached to each other at joints; the joints have a fixed angle but can rotate freely. In 2 dimensions, since the angles are fixed, we can

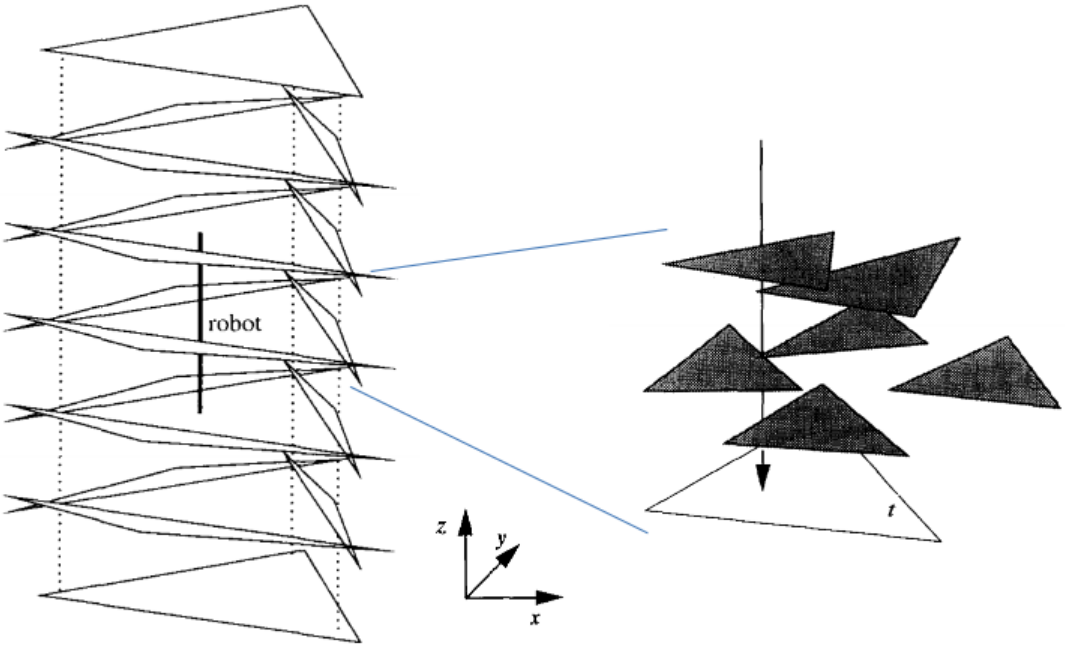

Figure 10: 3D motion planning.

only change whether the fixed angle is a left-hand turn or a right hand turn. Flipping an angle this way flips the whole subsequent structure as if the chain were a rigid body (see Figure 11). The goal of our problem is to find, given a fixed-angle chain in 2 dimensions, if there is an angle which can be flipped to cause a collision (two different points on the chain occupying the same point in the plane). [3]

We reduce from 3SUM' (the A-B-C version, where the goal is to find $a \in A, b \in B, c \in C$ such that $a + b = c$) as follows: given a 3SUM' instance A, B, C, we first find $M = \max(|x| : x \in A \cup B \cup C)$, which is large enough to separate the 3 groups if doubled. We then add $-2M$ to A and $2M$ to $-C$ to get A' and C', and set $B' = -B/2$. We then construct the chain in the following diagram, where the two lines $y = 0$ and $y = 1$ are each broken up by 'teeth' located at the values of the elements of A' and C' respectively; the two lines are joined by a step function where the steps are located at the values of the elements of B'.

It is trivial to verify that, with a careful construction, flipping any edge aside from the vertical edges corresponding to the elements of B' does not cause a collision. Thus, we only need to worry about these vertical segments.

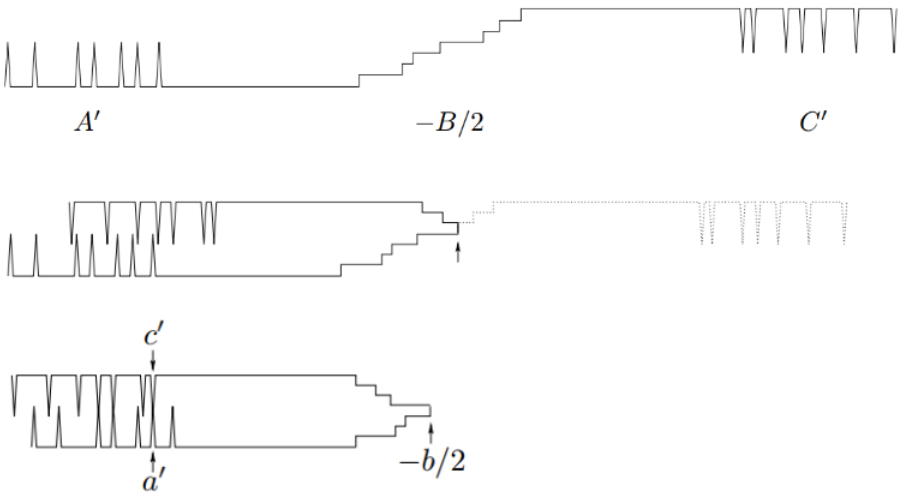

Figure 11: Fixed Angle Chains reduction to 3SUM'.

Consider flipping the segment at $b' \in B'$; the only segments that might come into contact are the 'teeth'. Suppose that the teeth at $a' \in A'$ and $c' \in C'$ come into contact; flipping horizontally at b' reflects everything to the right across the line $x = b'$, and thus causes $c' \to -(c' - 2b')$, so this only happens if $a' = -(c' - 2b')$ for some $a' \in A', b' \in B', c' \in C'$. But $a' = a - 2M$ for some $a \in A$, $b' = -b/2$ for some $b \in B$, and $c' = -c + 2M$ for some $c \in C$. Thus, we can conclude that

$$a' = -(c' - 2b') \implies a - 2M = -(-c + 2M) - b \implies a + b = c$$

so there is a solution to the 3SUM' problem iff there is a solution to the fixed-angle chains problem for this construction, so fixed-angle chains is 3SUM-Hard. [3]

Remark: Setting $C' = C + 2M$ (as opposed to $C' = -C + 2M$) gives a reduction from the 3SUM' variant where the goal is to find $a + b + c = 0$.

The best known algorithm for fixed angle chains is $O(n^3)$. [9]

3.3 Nonquadratic LBs from 3SUM-Hardness

If we assume that the 3SUM Conjecture is true, we can use 3SUM-Hardness to show some nonquadratic lower bounds as well, particularly for problems

on graphs. [6]

Given a graph $G = (V, E)$ with edge weights, suppose we want to find a triangle of a particular weight; we can do this in $O(|E|^{3/2} - \epsilon)$ time only if there is a $O(n^{2-\epsilon})$ algorithm (possibly not the same ϵ in these cases) for 3SUM.

Similarly, finding $|E|$ triangles in G (unweighted this time) in $O(|E|^{4/3} - \epsilon)$ time is 3SUM-Hard.

3.4 Conjectured Cubic Graph Problems

Given a graph $G = (V, E)$ (with or without edge weights), let $\delta(v, w)$ for $v, w \in V$ be the minimum distance between v and w.

We represent some of the relationships between these cubic graph problems in Figure 12.

Reductions to these cubic problems must take $O(n^{3-\epsilon})$.

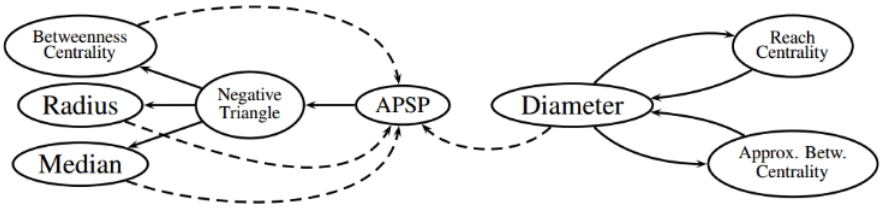

Figure 12: A depiction of the relationships between All Pairs Shortest Paths (APSP), Diameter and other problems.

3.4.1 Diameter

Goal: given G, find $\max_{v,w \in V} \delta(v, w)$ (the *diameter* of G).

Conjecture: there is no $O(|V|^{3-\epsilon})$ algorithm, even for unweighted graphs. [8]

It is known (assuming the ETH) that there is no $(3/2-\epsilon)$-approximation in $O(|E|^{2-\epsilon})$ time.

3.4.2 All-Pairs Shortest Paths (APSP)

Goal: find $\delta(v, w)$ for all pairs $v, w \in V$. Can do this in $O(|V|^3)$ time with the Floyd-Warshall Algorithm (by relaxing all edges $|V|$ times) (this also solves Diameter)

Conjecture: there is no $O(|V|^{3-\epsilon})$ algorithm, even for unweighted graphs.

It is trivial that APSP is at least as hard as Diameter; it is not known if the opposite is true.

We can create notions of Diameter-hard and APSP-hard in the same manner as 3SUM-hard (except now we are saying that a problem is APSP-hard if a solution to it implies a $O(|V|^{3-\epsilon})$ solution to APSP, and the same with Diameter). Multicall reductions are premitted, but with the restriction that if n_* is the size of a reduced instance, then $\sum n_*^{3-\epsilon} = O(n^{3-\epsilon})$.

3.4.3 Negative Triangles

Given a weighted graph the goal is to find a triangle of negative total weight; if we can solve this in subcubic time, we can solve APSP in subcubic time as well. Thus it is APSP-hard (in fact, it's equivalent). [10]

It is also equivalent to listing $|V|^{0.99}$ negative triangles, and to testing the triangle inequality.

3.4.4 New Results: Radius and Median

Radius: find $\min_{v \in V} \max_{w \in V} \delta(v, w)$, i.e. find the vertex which is closest to all other vertices. [1]

Median: find $\min_{v \in V} \sum_{w \in V} \delta(v, w)$, i.e. find the vertex which is closest to all other vertices on average. [1]

Both are APSP-Hard, and in fact equivalent (for both directed and undirected graphs); assuming Strong ETH, there is also no $O(|E|^{2-\epsilon})$ algorithm for sparse graphs.

References

[1] Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams, "Subcubic Equivalences Between Graph Centrality Problems, APSP and Diameter," 26th ACM/SIAM Symposium on Discrete Algorithms (SODA 2015)

[2] Ilya Baran, Erik Demaine, and Mihai Patrascu, "Subquadratic Algorithms for 3SUM," Algorithmica 50(4), 584-596 (2008)

[3] Jeff Erickson, Mark Overmars, and Michael Soss, "Preprocessing Chains for Fast Dihedral Rotations Is Hard or Even Impossible," in *Computational Geometry: Theory and Applications*, April 19, 2002.

[4] Anka Gajentaan and Mark H. Overmars, "On a Class of $O(n^2)$ problems in computational geometry," in *Computational Geometry (5) 1995*, October 14, 1994, pages 165-185.

[5] Allan Gronlund and Seth Pettie, "Threesomes, Degenerates, and Love Triangles," arXiv:1404.0799v3 May 30, 2014.

[6] Mihai Patrascu, "Towards Polynomial Lower Bounds for Dynamic Problems," 42nd ACM Symposium on Theory of Computing (STOC 2010)

[7] Mihai Patrascu and Ryan Williams, "On the possibility of faster SAT algorithms," 21st ACM/SIAM Symposium on Discrete Algorithms (SODA 2010)

[8] Liam Roditty and Virginia Vassilevska Williams, "Subquadratic Time Approximation Algorithms for the Girth," 23rd ACM/SIAM Symposium on Discrete Algorithms (SODA 2012)

[9] Michael Soss and Godfried Toussaint, "Recent Results on Reconfiguring Polygonal Chains in Space." Invited Paper, 957th Meeting of the American Mathematical Society. Toronto. September 2000.

[10] Virginia Vassilevska Williams and Ryan Williams,"Subcubic Equivalences Between Path, Matrix, and Triangle Problems," 2010 Symposium on Foundations of Computer Science. October 23-26, 2010.