

A Survey of 3SUM-Hard Problems

James King
king@cs.ubc.ca

December 20, 2004

1 Introduction

The 3SUM problem is defined as follows: given a set S of n integers, do there exist three elements $\{a, b, c\} \in S$ such that $a + b + c = 0$? This is a linear satisfiability problem. 3SUM can be solved using a simple algorithm with $\Theta(n^2)$ runtime (sort S , then test 3-tuples intelligently) and it is widely believed that $\Omega(n^2)$ is the lower bound for the worst-case runtime of any solution to the 3SUM problem.

Gajentaan and Overmars [GO95] relate a number of problems in computational geometry to the 3SUM problem. They define a problem to be ‘3SUM-hard’ if a constant number of 3SUM instances can solve the problem via a $o(n^2)$ reduction. In section 2 I discuss some of these problems and demonstrate some of the reductions used to prove 3SUM-hardness. In section 3 I discuss lower bounds for the time complexity of the 3SUM problem. In section 4 I discuss related problems and in section 5 I discuss areas of research where there seems to be potential for progress.

2 3SUM-Hard Problems

In this section I discuss problems that have been proven to be 3SUM-hard. For some of the problems I mention I will include the actual reductions to 3SUM. In terms of notation, I will use that of [GO95]. That is, for two problems PR1 and PR2,

$$PR1 \lll_{f(n)} PR2$$

means that every instance of PR1 of size n can be solved using a constant number of instances of PR2, plus $O(f(n))$ additional time.

$$PR1 \stackrel{f(n)}{=} PR2$$

means that $PR1 \lll_{f(n)} PR2$ and $PR2 \lll_{f(n)} PR1$.

A problem PR is called 3SUM-hard if and only if $3SUM \lll_{f(n)} PR$, where $f(n) = o(n^2)$.

There are too many 3SUM-hard problems for me to mention all of them here. This listing includes some of the more significant problems, as well as some of the fundamental 3SUM-hard problems discussed in the original paper [GO95].

3SUM'

Given three sets of integers A , B , and C such that $|A|+|B|+|C| = n$, are there three numbers $a \in A$, $b \in B$, and $c \in C$ such that $a + b = c$?

3SUM' is a variation of the original 3SUM problem. It is equivalent in difficulty to the original 3SUM problem. The reductions are linear, so $3SUM \stackrel{n}{=} 3SUM'$. The reductions consist of building new sets from the originals and modifying them arithmetically and can be found in [GO95].

GEOMBASE

Given n points on horizontal lines given by $y = 0$, $y = 1$, and $y = 2$, is there a non-horizontal line that goes through three points?

GEOMBASE is so named because it is used as an intermediate point for 3SUM hardness proofs of many problems in computational geometry, and is essentially the base geometric form of the 3SUM problem. $3SUM' \stackrel{n}{=} GEOMBASE$ [GO95], and the reductions are quite simple. To reduce GEOMBASE to 3SUM', for each integer $a \in A$ we create a point with coordinates $(a, 0)$. For each $b \in B$ we create a point with coordinates $(b, 2)$ and for each

$c \in C$ we create a point with coordinates $(\frac{c}{2}, 1)$. Now we have an instance of GEOMBASE. Clearly any non-horizontal line that goes through three points corresponds to some $a \in A, b \in B, c \in C$ such that $(a + b)/2 = c/2$, or $a + b = c$. Therefore we have $3SUM' \lll_n GEOMBASE$. We can essentially do the same reduction in reverse to show that $GEOMBASE \lll_n 3SUM'$, so we have $3SUM' =_n GEOMBASE$.

3-POINTS-ON-LINE

Given a set S of n points on the plane, are any three of them colinear?

Consider the curve given by $y = x^3$. Simple arithmetic tells us that three points (a, a^3) , (b, b^3) , and (c, c^3) are colinear if and only if $a + b + c = 0$. Therefore reducing 3-POINTS-ON-LINE to 3SUM is simply a matter of projecting the numbers from our 3SUM instance from the x -axis to the curve $y = x^3$. This tells us that $3SUM \lll_n 3\text{-POINTS-ON-LINE}$ [GO95].

Note that this problem is a special case of the problem of determining whether a set of *any* type of two dimensional objects in the plane has three elements that can be intersected by a single line. These problems, too, are therefore 3SUM-hard. Gomez et al. [GRT97] discuss the problem of finding, removing, then replacing degeneracies in input so as to find exact solutions to problems with degeneracies. This problem is 3SUM-hard for many types of degeneracies.

MINIMUM-AREA-TRIANGLE

Given a set S of n points on the plane, what is the area of the smallest triangle formed by any three of these points?

This problem 3SUM-hard because it is at least as hard as 3-POINTS-ON-LINE. Clearly a set S of points contains three colinear points if and only if the area of the smallest triangle is 0.

SEPARATOR

Given a set S of n line segments, is there a line that divides S into two nonempty subsets?

We can show that SEPARATOR is 3SUM-hard by reducing it to GEOMBASE [GO95]. For our reduction we need to create a set of line segments such that gaps between our line segments correspond to points in our instance of GEOMBASE. For each of our three original lines, we sort the points on it then create line segments between adjacent points such that there is an infinitesimally small gap at each point. After doing this for each line, we can create a ‘cap’ of extra line segments at each end of our set such that a line can only split the set of segments by passing through three of the gaps that correspond to our original points (see [GO95] for details). Thus SEPARATOR solves GEOMBASE after a $O(n \log n)$ time reduction. Therefore $3\text{SUM} \lll_{n \log n} \text{SEPARATOR}$, and SEPARATOR is 3SUM-hard.

STRIPS-COVER-BOX

Given a rectangle and n infinite strips (the space between two parallel lines), does the union of the strips cover the rectangle?

3SUM-hardness is proven in [GO95].

TRIANGLES-COVER-TRIANGLE

Given a set S of n triangles and a triangle t , does the union of the triangles in S cover t ?

3SUM-hardness is proven via a reduction of two instances of TRIANGLES-COVER-TRIANGLE to an instance of STRIPS-COVER-BOX. The rectangle in the original problem is split in half to turn the instance of STRIPS-COVER-BOX into two instances of ‘Strips-cover-triangle’. Each of these can in turn be solved by an instance of TRIANGLES-COVER-TRIANGLE by cutting off the irrelevant part of each strip to make a finite rectangle, then splitting each rectangle in half to make two triangles. STRIPS-COVER-BOX returns TRUE if and only if both instances of TRIANGLES-COVER-TRIANGLE return true. The

reduction is linear, so we have STRIPS-COVER-BOX \lll_n TRIANGLES-COVER-TRIANGLE.

PLANAR-MOTION-PLANNING

Given a set of line segments in the plane and a line segment robot, can the robot be moved from a start configuration to a goal configuration using translation and rotation?

This is about as easy as motion planning problems get, so its 3SUM-hardness is quite significant. It can be proven by reducing it to GEOMBASE. The reduction is very similar to the reduction of GEOMBASE to SEPARATOR; we build the same construction of line segments, then build large ‘cages’ of line segments both above our top line and below our bottom line. The starting configuration is inside the upper cage and the goal configuration is inside the lower cage. Our robot is long enough that, because of the cages, the robot can move from the start configuration to the goal configuration only by moving through a triplet of gaps that a line would fit through in SEPARATOR. Our reduction is the same as that of GEOMBASE to SEPARATOR, plus constant time, so we have GEOMBASE $\lll_{n \log n}$ PLANAR-MOTION-PLANNING and PLANAR-MOTION-PLANNING is 3SUM hard.

DIHEDRAL-ROTATION

Given a polygonal chain of n edges in 3-space and a dihedral rotation, does the rotation cause the chain to intersect itself?

The dihedral rotation problem arises especially when dealing with polymers in physics. A dihedral rotation involves rotating half of the chain rigidly around a given edge by a given angle. Soss et al. [SEO02] prove that preprocessing a static chain of length n and answering n queries is a 3SUM-hard problem. They also prove that, for a dynamic chain in which a successful query actually performs the rotation, performing n queries is 3SUM-hard, which suggests that no amount of preprocessing can yield sublinear query time.

Soss et al. reduce DIHEDRAL-ROTATION to 3SUM' by creating a polygonal chain with 'combs' of sharp 'teeth': one comb with teeth corresponding to elements in A and one with teeth corresponding to elements in C . In between the combs is a 'staircase' created using the elements in B such that certain rotations in the staircase are only legal if 3SUM' is *not* satisfied. See [SEO02] for more details.

POLYGON-CONTAINMENT

Given two polygons P and Q , can P be translated such that it lies completely within Q ?

3SUM-hardness for this problem was proven by Barequet and Har-Peled [BHP99]. In the same paper they offer proofs of 3SUM-hardness for variations of this problem, such as when the polygons must be convex but rotation is allowed. Reductions for the proofs use SEGMENTS-CONTAIN-POINTS as an intermediate problem. That asks, given a set P of n real numbers and a set Q of $O(n)$ disjoint real number intervals, can the points be shifted by some real number v such that, for every number $p \in P$, $p + v$ lies in one of the intervals in Q ? This problem is 3SUM-hard and some polygon containment problems (including the ones given) can be reduced to it in $o(n^2)$ time. It should be noted that some of these containment problems actually belong to a harder complexity class [Bar96] of which every 3SUM-hard problem is a member.

3 Lower Bounds

Lower bounds have not yet been proven for general models of computation. It is widely believed that $\Omega(n^2)$ is a tight lower bound for the worst-case runtime of any algorithm that solves 3SUM, and that there is therefore a similar bound for anything that solves a 3SUM-hard problem.

3SUM can certainly be solved in $O(n^2)$ time, and can even be solved faster in special cases. If all of the numbers are integers in the range $(-M, M)$ for some M , 3SUM' can be solved in time $O(n + M \log M)$. Recall that the same then applies to 3SUM, since we know that $3SUM \equiv_n 3SUM'$. The technique finds the bit vectors representing the three sets A , B , and C , then uses a fast

Fourier transform to calculate the integer vector representing the multiset $A + B$. It then compares that vector to the bit vector of C .

This technique is due to Seidel and appears via a personal communication with Erickson in [Eri95]. The problem is that it works only on a restricted subset of 3SUM instances. This is problematic not only in theory but also in practice. Also, it uses a fairly powerful model of computation; this fact is not extremely problematic in a practical sense, but does make the result less significant theoretically.

Erickson [Eri95] provides a lower bound for any algorithm that solves every instance of 3SUM. The bound is the desired tight $\Omega(n^2)$ bound, but only works in a weak model of computation. This is the algebraic decision tree model with restricted queries. All queries must be based on the sign of a linear combination of three elements of the set. This model of computation is sufficient for solving many 3SUM-hard problems. However, some 3SUM-hard problems, such as MINIMUM-AREA-TRIANGLE, cannot be solved at all using this model of computation. MINIMUM-AREA-TRIANGLE can be solved in $O(n^2)$ time [CGL83], but all solutions to do so involve more complex queries.

Erickson's lower bounds rely on an adversarial argument. He proves that, for any algorithm solving 3SUM using the restricted model of computation, there exists an input with $\Omega(n^2)$ 'critical 3-tuples'. Each critical 3-tuple is a query such that, if the algorithm has not yet performed it, the adversary is in a position where it can change the input such that the answers of previously made queries do not change, but our critical query must be performed to solve the 3SUM instance correctly. Since $\Omega(n^2)$ critical 3-tuples must be tested, the algorithm must run in $\Omega(n^2)$ time.

4 Related Problems

3SUM is actually a linear satisfiability problem for an equation in 3 variables. Closely related is the more general linear satisfiability problem for an equation in r variables for some r . Hernández Barrera [Bar96] investigates the complexity class related to this problem for $r = 4$ rather than $r = 3$ (essentially the class of 4SUM-hard problems). This class includes several

geometric problems, notably computing the Minkowski sum of two polygons and POLYGON-CONTAINMENT, as defined earlier.

Erickson's lower bound for 3SUM is a specific case of the more general lower bound he proves. Under the same model of computation mentioned before (though queries are now linear combinations of r elements), the linear satisfiability problem for an equation in r variables cannot be solved faster than $\Omega(n^{\lceil r/2 \rceil})$. This bound is tight for all odd r and is within a logarithmic factor for all even r .

5 Open Problems

Improved Lower Bounds

Though Erickson made significant progress towards proving a lower bound of $\Omega(n^2)$, it would be a significant achievement to extend his result to a more general model of computation. This would prove a true lower bound of $\Omega(n^2)$ for all 3SUM-hard problems. It is my belief that Erickson's lower bound will gradually be strengthened to include stronger and stronger models of computation. This, however, is a well studied problem and progress towards a better lower bound has been slow since Erickson's result.

Seemingly 3SUM-Hard Problems

There are still many problems that appear to be 3SUM-hard, but for which no lower bounds better than $\Omega(n \log n)$ have been proven. Proving that these problems are 3SUM-hard would offer a better lower bound, if only for a weak model of computation, and would open the door for further lower bound improvements.

The optimum binary search tree problem, for example, can be solved in $O(n^2)$ time [Knu71]. It is widely believed that this is also a lower bound, but no lower bound better than $\Omega(n \log n)$ has been proven. Knuth's algorithm uses dynamic programming, which is a much more powerful tool than what is required for a $O(n^2)$ 3SUM algorithm. Also, Knuth's algorithm requires $\Theta(n^2)$ storage, which is not required to solve 3SUM in $O(n^2)$ time (no optimum binary search tree algorithm has been found that uses $o(n^2)$ space

and polynomial time). These facts suggest that the problem is harder than 3SUM, though this has not been proven. Optimum binary search tree is just one of many problems that are believed to have $\Theta(n^2)$ worst-case runtime but are not known to be 3SUM-hard. Proving 3SUM-hardness would improve the best known lower bounds for these problems.

References

- [Bar96] A. Hernández Barrera. Finding an $o(n^2 \log n)$ algorithm is sometimes hard. In Frank Fiala, Evangelos Kranakis, and Jörg-Rüdiger Sack, editors, *Proceedings of the 8th Canadian Conference on Computational Geometry (CCCG-96)*, volume 5 of *International Informatics Series*, pages 289–294, Carleton University, August 12–15 1996. Carleton University Press.
- [BHP99] Gill Barequet and Sarel Har-Peled. Polygon-containment and translational min-hausdorff-distance between segment sets are 3sum-hard. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 862–863, N.Y., January 17–19 1999. ACM-SIAM.
- [CGL83] Bernard Chazelle, Leo J. Guibas, and D. T. Lee. The power of geometric duality. In *24th Annual Symposium on Foundations of Computer Science*, pages 217–225, Tucson, Arizona, 7–9 November 1983. IEEE.
- [Eri95] Jeff Erickson. Lower bounds for linear satisfiability problems. In *Proceedings of the 6th Annual Symposium on Discrete Algorithms*, pages 388–395, New York, NY, USA, January 1995. ACM Press.
- [GO95] Gajentaan and Overmars. On a class of $O(n^2)$ problems in computational geometry. *CGTA: Computational Geometry: Theory and Applications*, 5, 1995.
- [GRT97] Gomez, Ramaswami, and Toussaint. On removing non-degeneracy assumptions in computational geometry. In *CIAC: Italian Conference on Algorithms and Complexity*, 1997.
- [Knu71] Donald E. Knuth. Optimum binary search trees. *Acta Informatica*, 1(1):14–25, January 1971.
- [SEO02] Michael Soss, Jeff Erickson, and Mark Overmars. Preprocessing chains for fast dihedral rotations is hard or even impossible. April 19 2002.