

The 3SUM and APSP hypotheses are false

Bogdan Grechuk • 13 Oct 2026

<https://functor.network/user/3333/entry/2009>

On October 5, 2026, Josh Alman and Virginia Vassilevska Williams posted a remarkable preprint (Alman and Vassilevska Williams 2026) disproving two long-standing conjectures in theoretical computer science: the *3SUM hypothesis* and the *all-pairs shortest paths (APSP) hypothesis*. These conjectures were central to *fine-grained complexity theory*, which studies the precise running times of algorithms. The authors obtained the first polynomial improvements over the classical quadratic and cubic running times for these two problems.

Let us first recall the problems.

The *3SUM problem* asks whether a given set of n integers contains three distinct elements a, b, c satisfying

$$a + b + c = 0.$$

There is a simple $O(n^2)$ algorithm. First, sort the integers. Then, for each fixed element a , use two pointers to search for two other elements whose sum is $-a$. Each search takes $O(n)$ time, giving $O(n^2)$ time in total.

The *all-pairs shortest paths problem*, or *APSP*, concerns a directed graph with numerical weights assigned to its edges. The length of a path is the sum of its edge weights, and the task is to find the shortest-path distance between every ordered pair of vertices. Negative edge weights are allowed, but we assume that the graph contains no negative-weight cycles, since otherwise some shortest-path distances would not be well-defined.

The classical Floyd–Warshall algorithm solves APSP on a graph with n vertices in $O(n^3)$ time.

Both problems can therefore be solved in polynomial time. However, despite decades of research, nobody had been able to improve either running time by a fixed positive power of n . Algorithms saving logarithmic or other subpolynomial factors were known, but these improvements did not change the leading exponents.

This led to two famous conjectures. The *3SUM hypothesis* asserted that no algorithm could solve 3SUM in time

$$O(n^{2-\varepsilon})$$

for any fixed $\varepsilon > 0$. Similarly, the *APSP hypothesis* asserted that no algorithm could solve APSP in time

$$O(n^{3-\varepsilon})$$

for any fixed $\varepsilon > 0$.

Such algorithms are called *truly subquadratic* and *truly subcubic*, respectively. More precisely, the standard hypotheses concern polynomially bounded integer inputs in the *word RAM model*, in which arithmetic operations on integers occupying $O(\log n)$ bits take constant time.

These conjectures became cornerstones of fine-grained complexity theory. Researchers established numerous *reductions* showing that a sufficiently fast algorithm for one problem would imply a faster algorithm for another. Consequently, assuming the 3SUM or APSP hypothesis, one could obtain conditional lower bounds for the running times of many other algorithms.

Alman and Vassilevska Williams have now disproved both hypotheses.

Theorem 1 (Alman–Vassilevska Williams) *For every fixed constant $c > 0$, there exist deterministic algorithms in the word RAM model that solve:*

- *the 3SUM problem on n integers of absolute value at most n^c in time*

$$O(n^{1.9992});$$

- *the APSP problem on directed graphs with n vertices, integer edge weights of absolute value at most n^c , and no negative-weight cycles, in time*

$$O(n^{2.9995}).$$

The improvements in the exponents are tiny:

$$2 - 1.9992 = 0.0008, \quad 3 - 2.9995 = 0.0005.$$

Nevertheless, they are fixed positive constants, and this is precisely what the two hypotheses asserted was impossible. Moreover, the algorithms are deterministic, so no randomization is needed.

The main idea of the proof is surprisingly simple to formulate: *compute only the entries of a matrix product that are actually needed.*

Let X be an $N \times D$ matrix and Y a $D \times N$ matrix, where D is much smaller than N . Their product is an $N \times N$ matrix with entries

$$(XY)_{ij} = \sum_{k=1}^D X_{ik} Y_{kj}.$$

Suppose, however, that we need to compute only a relatively small collection of these entries, rather than the entire product.

The authors developed a new algorithm which, for

$$D \leq N^{1/18}$$

and matrices with polynomially bounded integer entries, can compute any prescribed collection of at most $N^2/\sqrt{D}$ entries of XY in time

$$O\left(\frac{N^2}{D^{0.063}}\right).$$

This is remarkable because computing the entire product requires at least N^2 operations just to output its entries. Computing the selected entries separately would be much slower, taking up to $N^2\sqrt{D}$ operations.

The new algorithm modifies a classical rectangular matrix multiplication construction of Coppersmith, based on an algebraic identity of Schönhage. The crucial observation is that many intermediate computations can be omitted when only selected entries of the product are required, while the remaining computations can still share enough work to give a substantial speedup.

How does this help with 3SUM and APSP?

The connection is through triangles in graphs. Consider a graph whose vertices are divided into three classes, with N , D , and N vertices, respectively. Let X and Y be the matrices describing the edges between consecutive classes, with entries equal to 0 or 1.

For a vertex i in the first class and a vertex j in the third class, the entry

$$(XY)_{ij}$$

counts the vertices in the middle class adjacent to both i and j . Thus, if i and j are also joined by an edge, this entry counts the triangles containing that edge.

Consequently, computing selected entries of a thin matrix product is equivalent to solving a version of the *All-Edges Sparse Triangle problem* on a graph in which one of the three classes is much smaller than the others.

Another problem, called *Exact Triangle*, asks whether a graph with integer edge weights contains a triangle whose three weights sum to zero. Earlier research established that both 3SUM and APSP can be reduced to Exact Triangle. Alman

and Vassilevska Williams show how to transform Exact Triangle into suitable sparse triangle problems, to which their new matrix algorithm applies. Combining these reductions gives Theorem 1.

The result is particularly striking because the same reductions had previously been used to argue that many problems were probably difficult. Now they work in the opposite direction: a faster algorithm for one problem immediately gives faster algorithms for many others. In fact, the paper also disproves several further hypotheses in fine-grained complexity and gives polynomial speedups for problems involving weighted cliques, matrix products, and graph algorithms.

The authors explicitly acknowledge that the core algorithm was discovered by Claude, an AI model developed by Anthropic. According to the paper, the algorithm emerged during an automated investigation of problems related to cryptography. Anthropic shared it with Alman and Vassilevska Williams, who then simplified, strengthened, and extended the argument, established further applications, and prepared the paper. The authors also report that the main results were formally verified using the Lean 4 proof assistant and Mathlib.

Of course, the new algorithms are primarily of theoretical interest. Their exponents are only marginally smaller than 2 and 3, and the hidden constants are very large. They are not expected to replace the classical algorithms in everyday computations.

Nevertheless, the conceptual significance is enormous. Two conjectures that had shaped algorithmic complexity for decades have been disproved, and a new algebraic technique has emerged that may lead to much larger improvements in the future. At the same time, other central hypotheses, such as the Strong Exponential Time Hypothesis, remain unaffected. The challenge now is to understand how far the new method can be pushed.

References

Alman, Josh, and Virginia Vassilevska Williams. 2026. *Truly Subquadratic 3SUM and Truly Subcubic APSP via Triangles in Sparse Lopsided Graphs*. <https://doi.org/10.48550/arXiv.2610.06783>.