Subquadratic Algorithms for Algebraic Generalizations of 3SUM

Luis Barba[1], Jean Cardinal*[2], John Iacono†[3], Stefan Langerman‡[2], Aurélien Ooms§[2], and Noam Solomon¶[4]

[1]Department of Computer Science, ETH Zürich, Switzerland, luis.barba@inf.ethz.ch
[2]Département d'Informatique, ULB, Belgium, {jcardin,slanger,aureooms}@ulb.ac.be
[3]Department of Computer Science and Engineering, NYU, USA, eurocg2017@johniacono.com
[4]School of Computer Science, TAU, Israel, noam.solom@gmail.com

Abstract

The 3SUM problem asks if an input n-set of real numbers contains a triple whose sum is zero. We consider the 3POL problem, a natural generalization of 3SUM where we replace the sum function by a constant-degree polynomial in three variables. The motivations are threefold. Raz, Sharir, and de Zeeuw gave an $O(n^{11/6})$ upper bound on the number of solutions of trivariate polynomial equations when the solutions are taken from the cartesian product of three n-sets of real numbers. We give algorithms for the corresponding problem of counting such solutions. Grønlund and Pettie recently designed subquadratic algorithms for 3SUM. We generalize their results to 3POL. Finally, we shed light on the General Position Testing (GPT) problem: "Given n points in the plane, do three of them lie on a line?", a key problem in computational geometry.

We prove that there exist bounded-degree algebraic decision trees of depth $O(n^{\frac{12}{7}+\varepsilon})$ that solve 3POL, and that 3POL can be solved in $O(n^2(\log\log n)^{\frac{3}{2}}/(\log n)^{\frac{1}{2}})$ time in the real-RAM model. Among the possible applications of these results, we show how to solve GPT in subquadratic time when the input points lie on $o((\log n)^{\frac{1}{6}}/(\log\log n)^{\frac{1}{2}})$ constant-degree polynomial curves. This constitutes the first step towards closing the major open question of whether GPT can be solved in subquadratic time. To obtain these results, we generalize important tools — such as batch range searching and dominance reporting — to a polynomial setting. We expect these new tools to be useful in other applications. Preprint available on arXiv [5].

*Supported by the "Action de Recherche Concertée" (ARC) COPHYMA, convention number 4.110.H.000023.

†Research partially completed while on sabbatical at the Algorithms Research Group of the Département d'Informatique at ULB with support from a Fulbright Research Fellowship, the Fonds de la Recherche Scientifique — FNRS, and NSF grants CNS-1229185, CCF-1319648, and CCF-1533564.

‡Directeur de recherches du F.R.S.-FNRS.

§Supported by the Fund for Research Training in Industry and Agriculture (FRIA).

¶Supported by Grant 892/13 from the Israel Science Foundation.

1 Introduction

The 3SUM problem is defined as follows: given n distinct real numbers, decide whether any three of them sum to zero. A popular conjecture is that no $O(n^{2-\delta})$-time algorithm for 3SUM exists. This conjecture has been used to show conditional lower bounds for problems in P, notably in computational geometry with problems such as GeomBase, general position [14] and Polygonal Containment [6], and more recently for string problems such as Local Alignment [2] and Jumbled Indexing [4], as well as dynamic versions of graph problems [1, 20], triangle enumeration and Set Disjointness [17]. For this reason, 3SUM is considered one of the key subjects of an emerging theory of complexity-within-P, along with other problems such as all-pairs shortest paths, orthogonal vectors, boolean matrix multiplication, and conjectures such as the Strong Exponential Time Hypothesis [3, 7, 16].

Because fixing two of the numbers a and b in a triple only allows for one solution to the equation $a+b+x=0$, an instance of 3SUM has at most n^2 solution triples. An instance with a matching lower bound is for example the set $\{\frac{1-n}{2},\ldots,\frac{n-1}{2}\}$ (for odd n) with $\frac{3}{4}n^2+\frac{1}{4}$ solution triples. One might be tempted to think that the number of solutions to the problem would lower bound the complexity of algorithms for the decision version of the problem, as it is the case for restricted models of computation [12]. This is a common misconception. Indeed, Grønlund and Pettie [15] recently proved that there exist $\tilde{O}(n^{3/2})$-depth linear decision trees and $o(n^2)$-time real-RAM algorithms for 3SUM.

A natural generalization of the 3SUM problem is to replace the sum function by a constant-degree polynomial in three variables $F\in\mathbb{R}[x,y,z]$ and ask to determine whether there exists any triple (a,b,c) of input numbers such that $F(a,b,c)=0$. We refer to this problem as the *3POL problem*.

For the particular case $F(x,y,z)=f(x,y)-z$ where $f\in\mathbb{R}[x,y]$ is a constant-degree bivariate polynomial, Elekes and Rónyai [10] show that the number of solutions to the 3POL problem is $o(n^2)$ unless f is *special*. Special for f means that f has one of the two special forms $f(u,v)=h(\varphi(u)+\psi(v))$ or

$f(u,v) = h(\varphi(u) \cdot \psi(v))$, where h, φ, ψ are univariate polynomials of constant degree. Elekes and Szabó [11] later generalized this result to a broader range of functions F using a wider definition of specialness. Raz, Sharir and Solymosi [22] and Raz, Sharir and de Zeeuw [21] recently improved both bounds on the number of solutions to $O(n^{11/6})$. They translated the problem into an incidence problem between points and constant-degree algebraic curves. Then, they showed that unless f (or F) is special, these curves have low multiplicities. Finally, they applied a theorem due to Pach and Sharir [19] bounding the number of incidences between the points and the curves. Some of these ideas appear in our approach.

In computational geometry, it is customary to assume the real-RAM model can be extended to allow the computation of roots of constant degree polynomials. We distance ourselves from this practice and take particular care of using the real-RAM model and the bounded-degree algebraic decision tree model with only the four arithmetic operators.

2 Our results

We focus on the computational complexity of 3POL. Since 3POL contains 3SUM, an interesting question is whether a generalization of Grønlund and Pettie's 3SUM algorithm exists for 3POL. If this is true, then we might wonder whether we can beat the $O(n^{11/6}) = O(n^{1.833\ldots})$ combinatorial bound of Raz, Sharir and de Zeeuw [21] with nonuniform algorithms. We give a positive answer to both questions: we show

Theorem 1 *There is a bounded-degree algebraic decision tree of depth $O(n^{\frac{12}{7}+\varepsilon}) = O(n^{1.7143})$ for 3POL. In the real-RAM model, 3POL can be solved in time $O(n^2(\log\log n)^{\frac{3}{2}}/(\log n)^{\frac{1}{2}})$.*

To prove our main result, we present a fast algorithm for the Polynomial Dominance Reporting (PDR) problem, a far reaching generalization of the Dominance Reporting problem. As the algorithm for Dominance Reporting and its analysis by Chan [8] is used in fast algorithms for all-pairs shortest paths, (min,+)-convolutions, and 3SUM, we expect this new algorithm will have more applications.

Our results can be applied to many degeneracy testing problems, such as the General Position Testing (GPT) problem: "Given n points in the plane, do three of them lie on a line?" It is well known that GPT is 3SUM-hard, and it is open whether GPT admits a subquadratic algorithm. Raz, Sharir and de Zeeuw [21] give a combinatorial bound of $O(n^{11/6})$ on the number of collinear triples when the input points are known to be lying on a constant number of polynomial curves, provided these curves are neither lines nor cubic curves. A corollary of our first result is that GPT where the input points are constrained to lie on $o((\log n)^{\frac{1}{6}}/(\log\log n)^{\frac{1}{2}})$ constant-degree polynomial curves (including lines and cubic curves) admits a subquadratic real-RAM algorithm and a strongly subquadratic bounded-degree algebraic decision tree. Interestingly, both reductions from 3SUM to GPT on 3 lines (map a to $(a,0)$, b to $(b,2)$, and c to $(\frac{c}{2},1)$) and from 3SUM to GPT on a cubic curve (map a to (a^3,a), b to (b^3,b), and c to (c^3,c)) construct such special instances of GPT. This constitutes the first step towards closing the major open question of whether GPT can be solved in subquadratic time. Our results also yield efficient algorithms for the problems of counting triples of points spanning unit circles or triangles.

3 Models of Computation

Similarly to Grønlund and Pettie [15], we consider both nonuniform and uniform models of computation. For the nonuniform model, Grønlund and Pettie consider linear decision trees, where one is only allowed to manipulate the input numbers through linear queries to an oracle. Each linear query has constant cost and all other operations are free but cannot inspect the input. In this paper, we consider *bounded-degree algebraic decision trees (ADT)* [23], a natural generalization of linear decision trees, as the nonuniform model. In a bounded-degree algebraic decision tree, one performs constant cost branching operations that amount to test the sign of a constant-degree polynomial for a constant number of input numbers. Again, operations not involving the input are free. For the uniform model we consider the real-RAM model with only the four arithmetic operators.

The problems we consider require our algorithms to manipulate polynomial expressions and, potentially, their real roots. For that purpose, we will rely on Collins cylindrical algebraic decomposition (CAD) [9].

Collins CAD solves any *geometric* decision problem that does not involve quantification over the integers in time doubly exponential in the problem size. This does not harm our results as we exclusively use this algorithm to solve constant size subproblems. Geometric is to be understood in the sense of Descartes and Fermat, that is, the geometry of objects that can be expressed with polynomial equations. In particular, it allows us to make the following computations in the real-RAM and bounded-degree ADT models:

1. Given a constant-degree univariate polynomial, count its real roots in $O(1)$ operations,
2. Given a constant number of univariate polynomials of constant degree, sort their real roots in $O(1)$ operations,
3. Given a point in the plane and an arrangement of a constant number of constant-degree polynomial planar curves, locate the point in the arrangement in $O(1)$ operations.

4 Nonuniform algorithm for explicit 3POL

As a glimpse of our results, we detail here our nonuniform algorithm for explicit 3POL. The other results can be found in the arXiv preprint [5].

Problem (explicit 3POL) *Let $f \in \mathbb{R}[x, y]$ be a bivariate polynomial of constant degree, given three sets A, B, and C, each containing n real numbers, decide whether there exist $a \in A$, $b \in B$, and $c \in C$ such that $c = f(a, b)$.*

Theorem 2 *There is a bounded-degree ADT of depth $O(n^{\frac{12}{7}+\varepsilon})$ for explicit 3POL.*

Idea The idea is to partition the sets A and B into small groups of consecutive elements. That way, we can divide the $A \times B$ grid into cells with the guarantee that each curve $c = f(x, y)$ in this grid intersects a small number of cells. For each such curve and each cell it intersects, we search c among the values $f(a, b)$ for all (a, b) in a given intersected cell. We generalize Fredman's trick [13] — and how it is used in Grønlund and Pettie's paper [15] — to quickly obtain a sorted order on those values, which provides us a logarithmic search time for each cell. Note that it is easy to modify the algorithm to count or report the solutions. In the latter case, the algorithm becomes output sensitive.

$A \times B$ grid partitioning Let $A = \{ a_1 < a_2 < \cdots < a_n \}$ and $B = \{ b_1 < b_2 < \cdots < b_n \}$. For some positive integer g to be determined later, partition the interval $[a_1, a_n]$ into n/g blocks $A_1^*, A_2^*, \ldots, A_{n/g}^*$ such that each block contains g numbers in A. Do the same for the interval $[b_1, b_n]$ with the numbers in B and name the blocks of this partition $B_1^*, B_2^*, \ldots, B_{n/g}^*$. For the sake of simplicity, and without loss of generality, we assume here that g divides n. To each of the $(n/g)^2$ pairs of blocks A_i^* and B_j^* corresponds a cell $A_i^* \times B_j^*$. By definition, each cell contains g^2 pairs in $A \times B$. For the sake of notation, we define $A_i = A \cap A_i^* = \{ a_{i,1} < a_{i,2} < \cdots < a_{i,g} \}$ and $B_j = B \cap B_j^* = \{ b_{j,1} < b_{j,2} < \cdots < b_{j,g} \}$. Figure 1 depicts this construction.

Two useful lemmas follow from this construction:

Lemma 3 *For a fixed value $c \in C$, the curve $c = f(x, y)$ intersects $O(\frac{n}{g})$ cells. Moreover, those cells can be found in $O(\frac{n}{g})$ time.*

Proof. The constant-degree polynomial curve $c = f(x, y)$ is composed of $O(1)$ xy-monotone pieces. Walk and sweep the $A \times B$ grid to locate those pieces. □

Lemma 4 *If the sets A, B, C can be preprocessed in $S_g(n)$ time so that, for any given cell $A_i^* \times B_j^*$ and any given $c \in C$, testing whether $c \in f(A_i \times B_j) = \{ f(a, b) \colon (a, b) \in A_i \times B_j \}$ can be done in $O(\log g)$ time, then explicit 3POL can be solved in $S_g(n) + O(\frac{n^2}{g} \log g)$ time.*

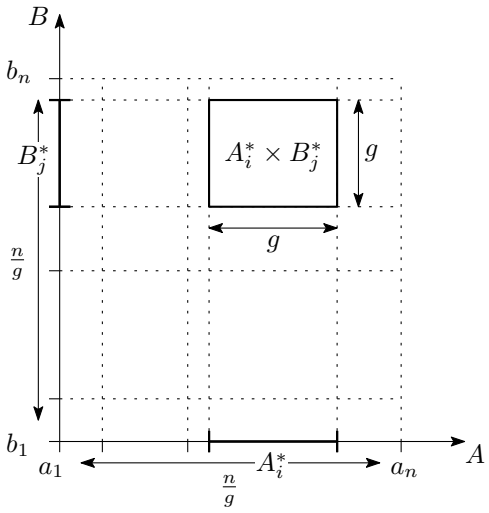

Figure 1: Partitioning A and B.

Remark We do not give a $S_g(n)$-time real-RAM algorithm for preprocessing the input, but only a $S_g(n)$-depth bounded-degree ADT. In fact, this preprocessing step is the only nonuniform part of the algorithm.

Preprocessing All that is left to prove is that $S_g(n)$ is subquadratic for some choice of g. To achieve this we sort the points inside each cell using Fredman's trick [13]. Grønlund and Pettie [15] use this trick to sort the sets $A_i + B_j = \{ a + b \colon (a, b) \in A_i \times B_j \}$ with few comparisons: sort the set $D = (\cup_i [A_i - A_i]) \cup (\cup_j [B_j - B_j])$, where $A_i - A_i = \{ a - a' \colon (a, a') \in A_i \times A_i \}$ and $B_j - B_j = \{ b - b' \colon (b, b') \in B_j \times B_j \}$, using $O(n \log n + |D|)$ comparisons, then testing whether $a + b \leq a' + b'$ can be done using the free (already computed) comparison $a - a' \leq b' - b$. We use a generalization of this trick to sort the sets $f(A_i \times B_j)$. For each B_j, for each pair $(b, b') \in B_j \times B_j$, define the curve $\gamma_{b,b'} = \{ (x, y) \colon f(x, b) = f(y, b') \}$. Define the sets $\gamma_{b,b'}^0 = \gamma_{b,b'}, \gamma_{b,b'}^- = \{ (x, y) \colon f(x, b) < f(y, b') \}, \gamma_{b,b'}^+ = \{ (x, y) \colon f(x, b) > f(y, b') \}$. The following lemma follows by definition:

Lemma 5 *Given a cell $A_i^* \times B_j^*$ and two pairs $(a, b), (a', b') \in A_i \times B_j$, deciding whether $f(a, b) < f(a', b')$ (respectively $f(a, b) = f(a', b')$ and $f(a, b) > f(a', b')$) amounts to deciding whether the point (a, a') is contained in $\gamma_{b,b'}^-$ (respectively $\gamma_{b,b'}^0$ and $\gamma_{b,b'}^+$).*

There are $N := \frac{n}{g} \cdot g^2 = ng$ pairs $(a, a') \in \cup_i [A_i \times A_i]$ and there are N pairs $(b, b') \in \cup_j [B_j \times B_j]$. Sorting the $f(A_i \times B_j)$ for all (A_i, B_j) amounts to locating each point (a, a') with respect to each curve $\gamma_{b,b'}$. We solve this subproblem in $O(N^{\frac{4}{3}+\varepsilon})$ operations using a

duality lemma and a modified version of the algorithm of Matoušek [18] for Hopcroft's problem.

Analysis Combining this new algorithm with Lemma 4 yields a $O((ng)^{4/3+\varepsilon} + n^2 g^{-1} \log g)$-depth bounded-degree ADT for 3POL. By optimizing over g, we get $g = \Theta(n^{2/7-\varepsilon})$, and the previous expression simplifies to $O(n^{12/7+\varepsilon})$, proving Theorem 2.

References

[1] Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *FOCS*, pages 434–443. IEEE Computer Society, 2014.

[2] Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *ICALP (1)*, volume 8572 of *LNCS*, pages 39–51, 2014.

[3] Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *STOC*, pages 41–50. ACM, 2015.

[4] Amihood Amir, Timothy M. Chan, Moshe Lewenstein, and Noa Lewenstein. On hardness of jumbled indexing. In *ICALP (1)*, volume 8572 of *LNCS*, pages 114–125, 2014.

[5] Luis Barba, Jean Cardinal, John Iacono, Stefan Langerman, Aurélien Ooms, and Noam Solomon. Subquadratic algorithms for algebraic generalizations of 3SUM. *ArXiv e-prints*, 2016. arXiv:1612.02384 [cs.DS].

[6] Gill Barequet and Sariel Har-Peled. Polygon containment and translational min Hausdorff distance between segment sets are 3SUM-hard. *Int. J. Comput. Geometry Appl.*, 11(4):465–474, 2001.

[7] Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *ITCS*, pages 261–270. ACM, 2016.

[8] Timothy M. Chan. All-pairs shortest paths with real weights in $O(n^3/\log n)$ time. *Algorithmica*, 50(2):236–243, 2008.

[9] George E. Collins. Hauptvortrag: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata Theory and Formal Languages*, volume 33 of *LNCS*, pages 134–183. Springer, 1975.

[10] György Elekes and Lajos Rónyai. A combinatorial problem on polynomials and rational functions. *J. Comb. Theory, Ser. A*, 89(1):1–20, 2000.

[11] György Elekes and Endre Szabó. How to find groups? (and how to use them in Erdős geometry?). *Combinatorica*, 32(5):537–571, 2012.

[12] Jeff Erickson. Lower bounds for linear satisfiability problems. *Chicago J. Theor. Comput. Sci.*, 1999.

[13] Michael L. Fredman. How good is the information theory bound in sorting? *Theor. Comput. Sci.*, 1(4):355–361, 1976.

[14] Anka Gajentaan and Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995.

[15] Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles. In *Foundations of Computer Science (FOCS 2014)*, pages 621–630. IEEE, 2014.

[16] Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *STOC*, pages 21–30. ACM, 2015.

[17] Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3SUM conjecture. In *SODA*, pages 1272–1287. SIAM, 2016.

[18] Jirí Matoušek. Range searching with efficient hierarchical cutting. *Discrete & Computational Geometry*, 10:157–182, 1993.

[19] János Pach and Micha Sharir. On the number of incidences between points and curves. *Combinatorics, Probability & Computing*, 7(1):121–127, 1998.

[20] Mihai Pătraşcu. Towards polynomial lower bounds for dynamic problems. In *STOC*, pages 603–610. ACM, 2010.

[21] Orit E. Raz, Micha Sharir, and Frank de Zeeuw. Polynomials vanishing on cartesian products: The Elekes-Szabó theorem revisited. In *SoCG*, volume 34 of *LIPIcs*, pages 522–536, 2015.

[22] Orit E. Raz, Micha Sharir, and József Solymosi. Polynomials vanishing on grids: The Elekes-Rónyai problem revisited. In *SoCG*, page 251. ACM, 2014.

[23] Andrew Yao. A lower bound to finding convex hulls. *J. ACM*, 28(4):780–787, 1981.