

USSR is in \mathbf{P}/poly

Nikhil Balaji^{*1} and Samir Datta^{†2}

¹Department of Computer Science and Engineering, IIT Delhi, New Delhi, India

²Chennai Mathematical Institute & UMI ReLaX, Chennai, India

Abstract

The Sum of Square Roots (**SSR**) problem is the following computational problem: Given positive integers a_1, \dots, a_k , and signs $\delta_1, \dots, \delta_k \in \{-1, 1\}$, check if $\sum_{i=1}^k \delta_i \sqrt{a_i} > 0$. The problem is known to have a polynomial time algorithm on the *real RAM* model of computation, however no sub-exponential time algorithm is known in the bit or Turing model of computation. The precise computational complexity of **SSR** has been a notorious open problem [GGJ76] over the last four decades. The problem is known to admit an upper bound in the third level of the *Counting Hierarchy*, i.e., $\mathbf{P}^{\mathbf{P}^{\mathbf{P}^{\mathbf{P}}}}$ and no non-trivial lower bounds are known. Even when the input numbers are *small*, i.e., given in *unary*, no better complexity bound was known prior to our work. In this paper, we show that the unary variant (**USSR**) of the sum of square roots problem is considerably easier by giving a \mathbf{P}/poly upper bound.

1 Introduction

The Sum of Square Roots problem (**SSR**) is the following computational problem: Given positive integers a_1, \dots, a_k , and signs $\delta_1, \dots, \delta_k \in \{-1, 1\}$, check if $\sum_{i=1}^k \delta_i \sqrt{a_i} > 0$. It was first explicitly posed in an influential paper of Garey, Graham and Johnson [GGJ76] on **NP**-hardness of geometric problems such as Euclidean TSP and Euclidean Steiner tree. They remark that while such geometric problems are **NP**-hard “it is not at all apparent that this ... is in **NP**”, since comparing two given tours is reducible to **SSR**, which is not known to be in **NP**. In fact, even when the input graph is presented via an embedding where vertices have integer coordinates, the distance between vertices could still be irrational in the Euclidean metric. Hence comparing lengths of two paths on such a graph, reduces to comparing two linear combinations of square roots for which the best bounds we currently have seem to require exponential precision [BFMS00]. However it has been often observed that in practice that a near-linear bit-precision is sufficient to decide the sign of an **SSR** instance [SY17, Chapter 45], and it is conjectured to be in \mathbf{P} . **SSR** is an important subroutine in Computational Geometry where many problems are known to be solvable in polynomial time relative to **SSR** [HS99]. We refer the interested reader to [DMO06, KS12] and the references therein for a discussion on the status of **SSR** and the stackexchange posts [Erib, Eria] for some interesting discussion on **SSR**-hard problems and related geometric questions. Along with the more general problem, **PosSLP**¹, **SSR** has been frequently used as to show *numerical hardness* in diverse areas such as algorithmic game theory [EY10, UW11, CIJ14], Semidefinite programming [Goe97], probabilistic verification [EKL08, HK15, KW11, BLW13], formal language theory and logic [HKL17, LOW15].

*nbalaji@cse.iitd.ac.in

†sdatta@cmi.ac.in

¹Given an arithmetic circuit using gates $\{+, \times\}$ whose only input is -1 , check if it computes a positive integer.

Prior work

Garey et al. [GGJ76] attribute Andrew Odlyzko to the observation that approximating every square root in the **SSR** instance by rational numbers with $O(k2^k)$ bits and summing them up suffices to infer the correct sign, which yields an **EXP** upper bound for **SSR**. The main idea behind this observation is that $\sum_{i=1}^k \delta_i \sqrt{a_i}$ is an *algebraic integer* whose *minimal polynomial* has degree at most 2^k . Therefore by using standard *root separation bounds* (see Proposition 2), one can infer the sign of any such algebraic integer by a suitable numerical approximation.

Using the root separation bound, Tiwari [Tiw92] proved that **SSR** can be solved in polynomial time on the unit cost arithmetic RAM model of computation via Newton iteration. Implicit in Tiwari’s algorithm is an improved complexity bound: Newton iteration can be used to obtain rational approximation to the square roots up to $O(k2^k)$ bits, which can be represented as pair of **poly**(k)-sized arithmetic circuits (for the numerator and the denominator). Evaluating such circuits can be done in **PSPACE**. We now know from the work of Allender et al. [ABKPM09] that Tiwari’s algorithm is essentially a polynomial time reduction from **SSR** to **PosSLP**, which places **SSR** in $\mathbf{P}^{\text{PPPPPP}}$. This is currently the best known complexity upper bound for **SSR**. A different algorithm achieving the same complexity bound, but via a reduction to *matrix powering* was presented in [ABD14]. All these results seem to inherently depend on root separation bounds. Unfortunately, this line of attack on **SSR** has its limitations: there exist an integer linear combination of k square roots, whose minimal polynomial has degree exactly 2^k ; Indeed consider $\alpha = \sum_{i=1}^k \sqrt{p_i}$ for distinct primes p_i . It is well-known from Galois theory [ST01] that the minimal polynomial of α has degree exactly 2^k . Therefore we require fundamentally new techniques to substantially improve the complexity of **SSR**.

Our work

Many authors have observed [GGJ76, Pap77] that Euclidean TSP is *strongly NP-hard*, i.e., the problem is already **NP-hard** for those graphs where the co-ordinates of the graph are integers of polynomially (in the number of vertices) bounded magnitude. This motivates the following variant, **USSR** : Given $1 \leq a_1, \dots, a_k \leq k$, and $\delta_1, \dots, \delta_k \in \{-1, 1\}$, test if $\sum_{i=1}^k \delta_i \sqrt{a_i} > 0$. We remark here that by the prime number theorem, the magnitude of numbers appearing under the square roots in the example we constructed above, i.e., $\alpha = \sum_{i=1}^k \sqrt{p_i}$ are at most $\mathcal{O}(k \log k)$. Even for this simple “unary” variant, there is no better complexity upper bound known than $\mathbf{P}^{\text{PPPPPP}}$. In this work, we improve the state of affairs considerably. We show that

Theorem 1. **USSR** $\in \mathbf{P/poly}$.

We provide two proofs of Theorem 1, both yielding a **P/poly** (in fact even a $\mathbf{TC}^0/\mathbf{poly}$) upper bound, which is tight modulo the non-uniformity since checking iterated addition of integers is \mathbf{TC}^0 -hard. This is a strong indication that **USSR** should be decidable in **P**. One consequence² of our result is that **USSR** cannot be **NP-complete** unless the polynomial hierarchy collapses to the second level. We would like to point out here that **USSR** is not a *sparse*³ language: For every k where the instances of **USSR** have k positive integers from the range $\{1, \dots, k\}$, there could be potentially 2^k distinct instances of **USSR**. Amazingly, there is still a $k^{\mathcal{O}(1)}$ -length *advice string* that non-trivially helps solve all the *exponentially-many instances*. To the best of our knowledge, this is the first instance of a natural (non-sparse) language that can be solved in **P/poly**, but not known to be solvable in the *polynomial hierarchy* (**PH**).

²However this consequence can be inferred by other means too: since **SSR** is closed under complement, if it was also **NP-complete**, this would imply $\mathbf{NP} = \mathbf{coNP}$, collapsing the polynomial hierarchy.

³If it were, then it would be trivially in **P/poly**

Related work

Cheng [Che06] studied the complexity of **USSR** and gave a subexponential time algorithm when the integers in the instance are bounded by $\mathcal{O}(k \log k)$ in magnitude. Kayal and Saha [KS12] define an analogue of **SSR** for univariate polynomials and obtain an efficient algorithm for deciding this variant. As a byproduct, they are able to show that **SSR** can be solved in **P** for the closely related class of *polynomial integers*. The variant they solve is orthogonal to **USSR** since there can be numbers of small magnitude that are not polynomial integers and vice versa. Both **SSR** and **USSR** have been conjectured [Che06] to admit efficient algorithms, though there has not been any concrete complexity-theoretic evidence before our work. A natural related question, namely whether a sum of square roots can be checked for equality, i.e., $\sum_{i=1}^k \delta_i \sqrt{a_i} = 0$ (as against **SSR** where one wants to infer the sign of the expression) can be solved efficiently in **P** [Blo91].

2 Preliminaries

Here we recall a few relevant propositionpositions about polynomials and integers, which can be found in standard texts on Computer Algebra [VZGG13] and Algebraic Number Theory [ST01]. For a polynomial $P(x) = \sum_{k=0}^n a_k x^k$ with integer coefficients, let $s = \text{size}(P) := \sum_{k=0}^n |a_k|$.

Proposition 2. [Mah64] *Let $P(x) = \sum_{k=0}^n a_k x^k = a_n \cdot \prod_{k=1}^n (x - \alpha_k)$ where $a_n \neq 0$ be an integer polynomial of size s and degree n . Then,*

$$\text{sep}(P) := \min_{\alpha_i \neq \alpha_j} |\alpha_i - \alpha_j| > \frac{\sqrt{3}}{n^{n/2+1} s^{n-1}}$$

Earlier we encountered the linear combination of square roots of primes as a hard instance of **USSR**. Infact, this observation is true also for linear combinations of square roots of square-free numbers. First we define square-free numbers and note that they are abundant.

Proposition 3 (Distribution of Square-free numbers [Sha08]). *A square-free number is a positive integer that is not divisible any perfect square integer other than 1. Let $m(k)$ denote the number of square-free numbers in the set $\{1, \dots, k\}$. Then $m(k)$ grows asymptotically as $\Theta(k)$.*

Since $m = m(k)$ and k have the same growth rate, in the rest of the paper, we will mildly abuse notation and use them interchangeably. The following important propositionerty of square roots of square free numbers will be an important component of our **P/poly** upper bound.

Proposition 4. *Let $\{s_1, \dots, s_m\} \subseteq \mathbb{N}$ be square-free positive integers. Then $\{\sqrt{s_1}, \dots, \sqrt{s_m}\}$ are linearly independent over the rationals. That is, if for some $c_1, \dots, c_m \in \mathbb{Q}$, $\sum_{i=1}^m c_i \sqrt{s_i} = 0$ then we have $c_1 = \dots = c_m = 0$.*

Moreover, the set of square-free numbers generate a field of large extension degree over the rationals.

Proposition 5 ([ST01, Bes40]). *Let $\{s_1, \dots, s_m\} \subseteq \mathbb{N}$ be square-free positive integers. Let $\alpha = \sum_{i=1}^m c_i \sqrt{s_i}$ where $c_i \in \mathbb{Z}$ and $p(x) \in \mathbb{Z}[x]$ be a polynomial such that $p(\alpha) = 0$. Then, $p(x)$ must have degree at least 2^m .*

3 Non-uniform algorithms for USSR

We present two fundamentally different proofs of the **P/poly** upper bound for **USSR**. Both the proofs require a crucial preprocessing step to obtain a *normal form* for **USSR** that we call *Universal USSR* (**U USSR**) consisting of “small” linear combinations of square roots of

square-free integers. Formally, **UUSSR** is the following computational problem: Given (Δ, U) where $\Delta \in [-k^2, k^2]^{m+1}$ and $U = \{\sqrt{s_0} = 1, \sqrt{s_1}, \dots, \sqrt{s_m}\}$, check if $\sum_{i=0}^m \Delta_i \sqrt{s_i} > 0$. We saw in Proposition 5 that square roots of square-free integers are a hard instance of **USSR**. We now observe that they are in fact the hardest subclass of **USSR**: That is, in order to solve **USSR**, it suffices to estimate the sign of linear combinations of square roots of square-free integers. We have the following

Proposition 6. *For every $k \in \mathbb{N}$, given a **USSR** instance (δ, A) , with $\delta \in \{\pm 1\}^k$, $A = \{a_1, \dots, a_k\}$ and $1 \leq a_i \leq k$, there exists a **UUSSR** instance computable in **poly**(k)-time, (Δ, U) where $\Delta \in [-k^2, k^2]^{m+1}$ and $U = \{\sqrt{s_0} = 1, \sqrt{s_1}, \dots, \sqrt{s_m}\}$, and s_1, \dots, s_m are the distinct square-free integers smaller than k such that,*

$$\sum_{i=1}^k \delta_i \sqrt{a_i} > 0 \text{ if and only if } \sum_{j=0}^m \Delta_j \sqrt{s_j} > 0$$

Proof. We design as an intermediate step the **USSR** instance $(\Gamma, [k])$ which consists of all the integers between 1 and k , i.e., $\sum_{\ell=1}^k \Gamma_\ell \sqrt{\ell}$. Now we can partition $[k]$ as

$$[k] = S \sqcup P \sqcup R$$

where

- $S = \{s_1, \dots, s_m\}$ are the square-free numbers.
- P consists of perfect squares.
- R consists of the rest, i.e., $R = [k] \setminus (S \cup P)$.

The sum $\sum_{\ell=1}^k \Gamma_\ell \sqrt{\ell}$ can be rewritten as

$$\sum_{\ell=1}^k \Gamma_\ell \sqrt{\ell} = \sum_{s \in S} \Gamma_s \sqrt{s} + \sum_{p \in P} \Gamma_p \sqrt{p} + \sum_{r \in R} \Gamma_r \sqrt{r}$$

where we set $\Gamma_s, \Gamma_p, \Gamma_r$ to be $\{0, \pm 1\}$ according to the given **USSR** instance (δ, A) as follows. If the number $\ell \in A$, then there must be a corresponding sign for this number in the δ vector, and we set Γ_ℓ to be this sign. Otherwise $\ell \notin A$ and we set $\Gamma_\ell = 0$. To prove the claimed bounds on Δ we proceed as follows.

- Since there will be roughly $\lfloor \sqrt{k} \rfloor$ perfect squares between 1 and k , the contribution of $\sum_{p \in P} \Gamma_p \sqrt{p}$ will be an integer Δ_0 , where $|\Delta_0| \leq k^{3/2} \leq k^2$.
- Any number that is neither square free nor a perfect square can be decomposed in to a square free part and a squared part. For example, if $a_i = c_i^2 s_i$ then we can write $\sqrt{a_i} = c_i \sqrt{s_i}$ where $c_i \leq \sqrt{k}$. Therefore $\sum_{r \in R} \Gamma_r \sqrt{r}$ can be rewritten as $\sum_{s \in S} \Gamma'_s \sqrt{s}$ where $\Gamma'_s \leq \sqrt{k}$.

Therefore we can rewrite the **USSR** instance $\sum_{\ell=1}^k \Gamma_\ell \sqrt{\ell}$ as a **UUSSR** instance $\sum_{j=0}^m \Delta_j \sqrt{s_j}$ where $\forall j \in \{0, 1, \dots, m\}, |\Delta_j| \leq k^2$ as claimed. The reduction is clearly in polynomial (in k) time. \square

Henceforth, we will concern ourselves with only instances of **UUSSR**: Given k , and $\Delta \in [-k^2, k^2]^m$, check if $\sum_{j=0}^m \Delta_j \sqrt{s_j} > 0$. Note that by Proposition 3, $m = \mathcal{O}(k)$. With the help of Proposition 6, we have encoded the input set of numbers A in to the small linear combinations Δ . We are now ready to give a proof of Theorem 1

3.1 Proof of Theorem 1 via Linear Threshold Functions

A Boolean function $f : \mathbb{B}^m \rightarrow \{\pm 1\}$ over a bounded discrete domain \mathbb{B} is said to be a Linear Threshold Function (LTF) if and only if *there exist* real numbers $w_0, w_1, \dots, w_m \in \mathbb{R}$ such that for every $x \in \mathbb{B}^m$, $f(x) = 1$ iff $\sum_{i=1}^m w_i x_i \geq w_0$. LTFs are central objects of study in many areas of theoretical computer science such as circuit complexity, learning theory, analysis of Boolean functions and communication complexity [AB09].

The real numbers w_0, w_1, \dots, w_m are said to be a *realization* of the LTF f . Note that any LTF has infinitely many realizations. The following classical theorem due to Muroga [Mur65] says that any LTF over a *bounded discrete domain* \mathbb{B} , can also be realized with integer weights of *small*⁴ magnitude (i.e., representable using at most $\mathcal{O}(m \log m)$ -bits). In fact the theorem shows that such a representation is only dependent on m and the the maximum absolute value of any element in the discrete bounded domain \mathbb{B} . In particular it is independent of the real numbers w_0, w_1, \dots, w_m .

Theorem 7 (Muroga [Mur65]). *Let b be the maximum magnitude of elements in \mathbb{B} . For any LTF $f : \mathbb{B}^m \rightarrow \{\pm 1\}$ realized using weights $w_0, w_1, \dots, w_m \in \mathbb{R}$, there exists an equivalent integer realization with weights $u_0, u_1, \dots, u_m \in \mathbb{Z}$, where $1 \leq |u_0|, \dots, |u_m| \leq \mathcal{O}((m+1)!b^m)$.*

The proof of Theorem 7 follows from a simple application of Linear Programming and is a fundamental result in Circuit Complexity and Learning Theory. We give an exposition of the proof in the Appendix A for the sake of completeness. The proof of Theorem 1 is now immediate; we simply consider the LTF which is realized using the weights $\Delta_0, \Delta_1 \sqrt{s_1}, \dots, \Delta_m \sqrt{s_m}$ and observe that there is also an integer realization of this LTF that uses small *integer* weights of magnitude at most $m^{\mathcal{O}(m)}$.

Proof. (of Theorem 1)

We invoke Theorem 7 with $\mathbb{B} = \{-1, 0, 1\}$ and $w_0 = \Delta_0, w_1 = \Delta_1 \sqrt{s_1}, \dots, w_m = \Delta_m \sqrt{s_m}$. The LTF $f : \mathbb{B}^m \rightarrow \{\pm 1\}$ is given by $f(x) = 1$ if and only if $\sum_{j=1}^m x_j |\Delta_j| \sqrt{s_j} > \Delta_0$. By Theorem 7, there exist integers u_0, \dots, u_m of magnitude at most $\mathcal{O}((m+1)!) = m^{\mathcal{O}(m)}$ such that for every $x \in \mathbb{B}^m$

$$\sum_{j=1}^m x_j \Delta_j \sqrt{s_j} > \Delta_0 \text{ if and only if } \sum_{j=1}^m x_j u_j > u_0$$

The non-uniform advice for our algorithm will be exactly these $m+1$ numbers u_0, \dots, u_m each of which is $\mathcal{O}(m \log m)$ bits long. Since $m = \mathcal{O}(k)$ by Proposition 3, for every k , the total length of the advice string is $\mathcal{O}(k^2 \log k)$ bits. Given this advice, to decide the sign of any given **UUSSR** instance $\sum_{j=0}^m \Delta_j \sqrt{s_j}$, we can instead now check if $\sum_{j=1}^m u_j > u_0$. Since u_0, \dots, u_m are integers of magnitude at most $\mathcal{O}((m+1)!) = 2^{\mathcal{O}(m \log m)}$, such a computation is just an iterated addition of $\mathcal{O}(m \log m)$ -bit integers which can be performed in polynomial time. \square

Remark 8. *A closer observation of the proofs of Proposition 6 and Theorem 7 shows that in fact **USSR** \in **TC**⁰/**poly**. The reduction from **USSR** to **UUSSR** in Proposition 6 can be implemented in **TC**⁰: Since the numbers are in unary, checking membership in the sets S, P and R reduces to bruteforce division. Similarly checking if $\sum_{j=1}^m u_j > u_0$ is essentially an iterated addition of $\mathcal{O}(k \log k)$ -bit integers. Since division and iterated addition can be performed in **TC**⁰ [HAB02], this shows that both **UUSSR** and **USSR** are in **TC**⁰/**poly**.*

⁴It is easy to see that there always exists *some* integer realization for any LTF. Indeed, consider $\varepsilon = \min_{x \in \mathbb{B}^m} (\sum_{i=1}^m w_i x_i - w_0)$, and for every $i \in [m]$, pick a rational number w'_i in the interval $(w_i - \frac{\varepsilon}{10^m}, w_i + \frac{\varepsilon}{10^m})$. Clearly, for every $x \in \mathbb{B}^m$, $\sum_{i=1}^m w_i x_i - w_0 > 0$ if and only if $\sum_{i=1}^m w'_i x_i - w'_0 > 0$. By clearing the denominators of the rationals, we obtain the integer realization. Notice however that the magnitude of w'_i depend on ε

3.2 Proof of Theorem 1 via Numerical Approximations

We now present a fundamentally different **P/poly** upper bound proof which allows us to solve **UUSSR** provided we know the *approximate magnitudes* of $(m+1)$ *smallest* **UUSSR** instances. The high-level idea behind the proof is follows: Given an instance of **UUSSR** as in Proposition 6 we can interpret the **USSR** problem as estimating the sign of the inner product of a vector in \mathbb{B}^{m+1} with a *fixed vector* in \mathbb{R}^{m+1} , namely the vector of square roots of square-free numbers. Next, observe that there exists a set of $m+1$ vectors in \mathbb{B}^{m+1} which spans \mathbb{R}^{m+1} . Could it be possible that the sign of this inner product can be estimated from the sign/value of the inner products of the basis elements? While this is not possible for an arbitrary basis, we show that there exists a *nice* basis which allows us to do precisely this. Therefore, if this basis along with the *approximate values* of the inner product of the basis vectors is given as advice, we show that we can estimate the sign of **UUSSR** instance.

First we set up some notation. As alluded to earlier, any **UUSSR** instance can be interpreted as an inner product as follows:

$$A = \sum_{j=0}^m \Delta_j \sqrt{s_j} = \langle \vec{\Delta}, \vec{S} \rangle$$

where $\vec{\Delta} = (\Delta_0, \dots, \Delta_m) \in [-k^2, k^2]^{m+1}$ and $\vec{S} = (1, \sqrt{s_1}, \dots, \sqrt{s_m}) \in \mathbb{R}^{m+1}$. By Proposition 4, none of them evaluate to 0 except the inner product with the zero vector. In what follows, we consider only those vectors $\vec{\Delta}$ which have a *positive inner product* on \vec{S} , i.e., $W = \{\vec{\Delta} \in [-k^2, k^2]^{m+1} \mid \langle \vec{\Delta}, \vec{S} \rangle > 0\} \subseteq \mathbb{R}^{m+1}$. Let $|W| = t$. Let $\{v_i\}_{i=1}^t$ be the magnitude of the inner product of the elements of W . By our assumption we have $0 < v_0 v_1, \dots, v_t$ and each of these numbers are distinct since if $v_i = v_j$ it gives rise to a non-trivial linear dependence between the square roots, which is impossible by Proposition 4.

We now construct the unique *lightest* basis for \mathbb{R}^{m+1} from the vectors in W by picking the least (ordered by magnitude of inner product) $m+1$ linearly independent vectors in $[-k^2, k^2]^{m+1}$. It is clear that a basis for \mathbb{R}^{m+1} using these vectors exists and is unique (uniqueness follows from Proposition 4). We denote these vectors by $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_m$, where $\mathbf{b}_i = (b_{i0}, \dots, b_{im}) \in [-k^2, k^2]^{m+1}$. Let $v_0 = \langle \mathbf{b}_0, \vec{S} \rangle, v_1 = \langle \mathbf{b}_1, \vec{S} \rangle, \dots, v_m = \langle \mathbf{b}_m, \vec{S} \rangle$. Given an instance of **UUSSR** of the form $A = \langle \vec{\Delta}, \vec{S} \rangle$, it is clear that we can express $\vec{\Delta}$ in the lightest basis as $\vec{\Delta} = \sum_{i=0}^m c_i \mathbf{b}_i$. The c_i can be obtained by solving a system of linear equations over the rationals. This immediately gives an upper and lower bound on c_i : The maximum entry of the matrix of such a linear system is at most k^2 (which is also the maximum possible entry in any \mathbf{b}_i); Therefore by Cramer's rule we get that for all $i, |c_i| \leq \mathcal{O}((m+1)!k^{2m}) = 2^{\mathcal{O}(m \log km)}$. Since all the entries of the matrix are integers, if $c_i \neq 0$, we also have $|c_i| \geq \frac{1}{2^{\mathcal{O}(m \log km)}}$. Now, we have

$$A = \left\langle \sum_{i=0}^m c_i \mathbf{b}_i, \vec{S} \right\rangle = \sum_{i=0}^m c_i \langle \mathbf{b}_i, \vec{S} \rangle = \sum_{i=0}^m c_i v_i$$

Remark 9. Note that just by observing that $\frac{1}{2^{\mathcal{O}(m \log km)}} \leq c_i \leq 2^{\mathcal{O}(m \log km)}$ we can conclude that there exists $i \in \{0, 1, \dots, m\}$ such that $|v_i| > \frac{1}{2^{\mathcal{O}(m \log km)}}$. That is, all the basis vectors cannot have double exponentially small inner product with \vec{S} . Because, otherwise, they cannot express large instance of **USSR** which are guaranteed to exist, for e.g. $A = \sum_{i=0}^m \sqrt{s_i}$ whose magnitude is larger than m .

Notice that expressing the **UUSSR** instance in the lightest basis immediately yields the following simple lower bound.

Proposition 10.

$$|A| = \left| \sum_{i=0}^m c_i \langle \mathbf{b}_i, \vec{S} \rangle \right| = \left| \sum_{i=0}^m c_i v_i \right| > v_{m_1}$$

where $m_1 \in [m]$ is the largest index such that $c_{m_1} \neq 0$.

Proof. If $|A| < v_{m_1}$, then by definition (of the lightest basis) the basis vector \mathbf{b}_{m_1} can be replaced by $\vec{\Delta}$. \square

From the preceding discussion, it seems like in order to find the sign of the quantity A , it suffices to know the magnitude and sign of all the v_i . The best bounds for v_i are obtained through root separation bounds (Proposition 2), which turn out to be doubly exponentially small as a function of m in magnitude (Proposition 5). Therefore, they need exponential (in m) number of bits to express them unambiguously and any arithmetic with such numbers is bound to result in an exponential time algorithm!

However the lower bound from Proposition 10 gives us some leeway to use approximations to v_i instead of v_i themselves. We will now show that if we have a suitable *rational approximation* \tilde{v}_i to the v_i 's the resulting error is small. Towards this we now assume we work with $\tilde{v}_i = \beta_i 2^{-e_i}$, where β_i is a $\mathcal{O}(m^2)$ -bit rational number in the interval $(1, 2)$. Notice that by standard root separation bounds (see Proposition 2 and 5), one can infer that $|e_i| \leq 2^{\mathcal{O}(m)}$ and hence expressible using $\mathcal{O}(m)$ bits. The $\{\tilde{v}_i = \beta_i 2^{-e_i}\}_{i=0}^m$ are approximations to $\{\langle \mathbf{b}_i, \vec{S} \rangle = \sum_{j=0}^m b_{ij} \sqrt{s_j}\}_{i=0}^m$. We have the following

Proposition 11. $\tilde{A} = \sum_{i=0}^m c_i \tilde{v}_i = \sum_{i=0}^m c_i \beta_i 2^{-e_i} > 0$ if and only if $A = \sum_{i=0}^m c_i v_i > 0$

Proof. If we use \tilde{v}_i instead of v_i , we can upper bound the total accumulated error incurred as follows:

$$\begin{aligned} \left| \sum_{i=0}^m c_i (v_i - \tilde{v}_i) \right| &\leq \sum_{i=0}^m c_i |v_i - \tilde{v}_i| = \sum_{i=0}^m c_i |v_i - \beta_i 2^{-e_i}| \\ &\leq \sum_{i=0}^m (2^{m \log m}) \cdot (2^{-m^2} \cdot 2^{-e_i}) \leq m \cdot (2^{m \log m}) \cdot (2^{-m^2} 2^{-e_{m_1}}) \end{aligned}$$

where the inequality $|v_i - \beta_i 2^{-e_i}| \leq (2^{-m^2} \cdot 2^{-e_i})$ above is obtained by noticing that by definition the approximation \tilde{v}_i agrees with v_i on the mantissa. The last inequality is obtained by noting that v_{m_1} is the basis vector with the largest magnitude that is used in expressing A in terms of the lightest basis. We can now lower bound the ratio of value of the given instance of **UUSSR** to the error accrued is at least

$$\frac{\left| \sum_{i=0}^m c_i v_i \right|}{\left| \sum_{i=0}^m c_i (v_i - \tilde{v}_i) \right|} \geq \frac{2^{-e_{m_1}}}{m 2^{m \log m} 2^{-m^2} 2^{-e_{m_1}}} = 2^{\Omega(m^2)} \quad (1)$$

This means that the total error accrued by using \tilde{v}_i instead of v_i is a very small fraction of the absolute value of A and hence the $\mathcal{O}(m^2)$ -bit approximation to each v_i , namely $\tilde{v}_i = \beta_i 2^{-e_i}$ suffices to compute the correct sign of A . \square

We are now ready to give our second proof of the **P/poly** upper bound.

Proof. We have as our **poly**(k)-length advice string:

1. The lightest basis $\{\mathbf{b}_i \in [-k^2, k^2]^m\}_{i=0}^m$.
2. $\{\tilde{v}_i = \beta_i 2^{-e_i}\}_{i=0}^m$, which are approximations to $\{\langle \mathbf{b}_i, \vec{S} \rangle = \sum_{j=0}^m b_{ij} \sqrt{s_j}\}_{i=0}^m$.

Given this advice, our **P/poly** algorithm for the **UUSSR** instance $A = \langle \vec{\Delta}, \vec{S} \rangle$ is straight forward: Firstly notice that $A \neq 0$ by Proposition 4 and this can be checked in polynomial time [Blo91, HBM⁺10] as well. If $\vec{\Delta}$ is present in the lightest basis or a multiple of one of the basis elements, then the sign is just obtained directly from the advice string in conjunction with the polynomial time tests mentioned above. Therefore, let us assume that Δ is a non-trivial linear combination of the basis vectors. We proceed as follows:

1. Find c_i such that $\vec{\Delta} = \sum_{i=0}^m c_i \mathbf{b}_i$.
2. Return the sign of the appropriate linear combination $\tilde{A} = \sum_{i=0}^m c_i \tilde{v}_i$.

Finding c_i can be done by solving a system of linear equations, and hence is in **P**. However, it is not clear how to return the sign of the linear combination $\tilde{A} = \sum_{i=0}^m c_i \tilde{v}_i$, because this involves arithmetic with exponential-bit numbers! Recall that by the root separation bound, we only ensured that $e_i \leq 2^{\mathcal{O}(m)}$, so \tilde{v}_i 's are still exponential-bit approximations to v_i which admit a succinct representation. We are now left with the task of ensuring that (approximately) computing the linear combination can be done efficiently. For this, let $m_1 \in [m]$ be the largest index such that $c_{m_1} \neq 0$. Also let $m_0 \leq m_1$ be the smallest index such that $e_{m_0} - e_{m_1} \leq m^2$. Let

$$\tilde{V} = \sum_{i=m_0}^{m_1} c_i \beta_i 2^{-e_i} = \left(\sum_{i=m_0}^{m_1} c_i \beta_i 2^{-(e_i - e_{m_1})} \right) 2^{-e_{m_1}} \quad (2)$$

where the quantity inside the large parentheses is a sum of at most m rationals each with at most a $2m^2$ -bit representation, and hence can be computed in **TC⁰**. Essentially we drop all those terms whose exponents are larger than m_0 . It follows that \tilde{V} is a good approximation to \tilde{A} and hence also to A . To this end, notice that we introduce two kinds of errors while making this approximation:

1. The error due to truncation (as argued in Proposition 11). This error is upper bounded in absolute value by $(m_1 - m_0 + 1)2^{-(e_{m_1} + m^2)}$.
2. The error due to dropping of all terms preceding m_0 in Equation 2. This error is upper bounded in absolute value by $(m_0 - 1)2^{-(e_{m_0-1} + m^2)}$.

But we know from the definition of m_0 that $|e_{m_0-1}| > |e_{m_1}| + m^2$. Thus the total error is upper bounded in magnitude by $m_1 2^{-(e_{m_1} + 2m^2)}$. As a fraction of the magnitude of the **UUSSR** instance, this error is small and as argued previously via Equation 1, our approximation using \tilde{v}_i always returns the right answer. \square

Remark 12. *Note that our second proof seems weaker than the first proof since it seems to require solving a system of linear equations which is not known to be implementable in **TC⁰**. However, notice that if we provide the inverse of the $(m+1) \times (m+1)$ matrix $[\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_m]$ as advice, the algorithm above can be implemented in **TC⁰**. Note that this is only an additional $\mathbf{poly}(m) = \mathbf{poly}(k)$ bits long since each entry is a $\mathbf{poly}(k)$ -bit rational number.*

4 Summary and Conclusion

We have presented two non-uniform algorithms for **USSR**. Both can be viewed as “compressing” exponentially many input instances into a polynomial-length advice string such that each instance can be solved *easily* given the advice. Moreover, it is easy to see that the unary nature of **USSR** does not play an important role here. In other words, given any set of k *fixed* positive integers the corresponding 2^k instances of **SSR** can be similarly compressed into a polynomial length advice string from which extracting the solution to an instance is easy. In fact, the compression is possible for the signed sum of any k *fixed* real numbers.

A notable point about the numerical approximation proof is that the number of basis elements b_i is only $m + 1$ while by the root separation bound their exponents e_i are distributed over an exponential range in m . Thus if the e_i 's are “well-distributed” i.e. consecutive e_i 's differ by $\Omega(m^2)$ then we just need to consult the value of *one* basis element to decide on an instance. Prior to our work, there was no difference in the computational complexity between **USSR** and **SSR**. Our work raises a few natural questions:

1. Is there a polynomial time algorithm for **USSR**? Is there a way to use both the proofs in tandem to *verify* the advice? This would give an **NP** algorithm for **USSR**. Even a **PH** algorithm will be a significant improvement to the current state of affairs.
2. Consider the proof via LTFs: Is it possible to estimate the number of vertices in the polytope of the LTF obtained from the **UUSSR** instance? Can we characterize when the vertices will be determined by *easy* **UUSSR** instances?
3. Is it possible to eliminate either the values $\{v_i\}_{i=0}^m$ or the basis $\{\mathbf{b}_i\}_{i=0}^m$ from the advice string in the numerical approximation proof?
4. Is **SSR** \in **P/poly**?
5. Can reasoning similar to Remark 9 be used to give better bounds on distribution on roots of a univariate polynomial, or give improved root separation bounds?

Acknowledgements We would like to thank Nikhil Mande, Mahesh Sreekumar Rajasree, Vikram Sharma, Mahsa Shirmohammadi and James Worrell for many insightful discussions around the Sum of Square Roots problem. N.B was partially supported by the International Emerging Actions grant (IEA'22).

References

- [AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
- [ABD14] Eric Allender, Nikhil Balaji, and Samir Datta. Low-depth uniform threshold circuits and the bit-complexity of straight line programs. In *Mathematical Foundations of Computer Science 2014*, pages 13–24. Springer, 2014.
- [ABKPM09] Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2009.
- [Bes40] Abram S Besicovitch. On the linear independence of fractional powers of integers. *Journal of the London Mathematical Society*, 1(1):3–6, 1940.
- [BFMS00] Christoph Burnikel, Rudolf Fleischer, Kurt Mehlhorn, and Stefan Schirra. A strong and easily computable separation bound for arithmetic expressions involving radicals. *Algorithmica*, 27(1):87–99, 2000.
- [Blo91] J Blomer. Computing sums of radicals in polynomial time. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 670–677. IEEE, 1991.
- [BLW13] Michael Benedikt, Rastislav Lenhardt, and James Worrell. Ltl model checking of interval markov chains. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 32–46. Springer, 2013.

- [Che06] Qi Cheng. On comparing sums of square roots of small integers. In *International Symposium on Mathematical Foundations of Computer Science*, pages 250–255. Springer, 2006.
- [CIJ14] Krishnendu Chatterjee and Rasmus Ibsen-Jensen. The complexity of ergodic mean-payoff games. In *International Colloquium on Automata, Languages, and Programming*, pages 122–133. Springer, 2014.
- [DMO06] Erik D Demaine, Joseph SB Mitchell, and Joseph O’Rourke. The open problems project: Problem 33, 2006.
- [EKL08] Javier Esparza, Stefan Kiefer, and Michael Luttenberger. Solving monotone polynomial equations. In *Fifth Ifip International Conference On Theoretical Computer Science–Tcs 2008*, pages 285–298. Springer, 2008.
- [Eria] Jeff Erickson. Complexity of computing shortest paths in the plane with polygonal obstacles. [cstheory.stackexchange. URL:https://cstheory.stackexchange.com/q/4034/256](https://cstheory.stackexchange.com/q/4034/256) (visited on 2010-12-31).
- [Erib] Jeff Erickson. Sum-of-square-roots-hard problems? [cstheory.stackexchange. URL:https://cstheory.stackexchange.com/q/4053/256](https://cstheory.stackexchange.com/q/4053/256) (visited on 2011-01-01).
- [EY10] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM J. Comput.*, 39(6):2531–2597, 2010.
- [GGJ76] Michael R Garey, Ronald L Graham, and David S Johnson. Some np-complete geometric problems. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 10–22. ACM, 1976.
- [Goe97] Michel X Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79(1-3):143–161, 1997.
- [HAB02] W. Hesse, E. Allender, and D.A.M. Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65:695–716, 2002.
- [HBM⁺10] Paul Hunter, Patricia Bouyer, Nicolas Markey, Joël Ouaknine, and James Worrell. Computing rational radical sums in uniform tc^0 . In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.
- [HK15] Christoph Haase and Stefan Kiefer. The odds of staying on budget. In *International Colloquium on Automata, Languages, and Programming*, pages 234–246. Springer, 2015.
- [HKL17] Christoph Haase, S Kiefer, and Markus Lohrey. Counting problems for Parikh images. In *Leibniz International Proceedings in Informatics, LIPIcs*, volume 83, page 12. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.
- [HS99] John Hershberger and Subhash Suri. An optimal algorithm for euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.
- [KS12] Neeraj Kayal and Chandan Saha. On the sum of square roots of polynomials and related problems. *ACM Transactions on Computation Theory (TOCT)*, 4(4):9, 2012.

- [KW11] Stefan Kiefer and Dominik Wojtczak. On probabilistic parallel programs with process creation and synchronisation. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 296–310. Springer, 2011.
- [LOW15] Antonia Lechner, Joël Ouaknine, and James Worrell. On the complexity of linear arithmetic with divisibility. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 667–676. IEEE, 2015.
- [Mah64] Kurt Mahler. An inequality for the discriminant of a polynomial. *The Michigan Mathematical Journal*, 11(3):257–262, 1964.
- [Mur65] Saburo Muroga. Lower bounds of the number of threshold functions and a maximum weight. *IEEE Transactions on Electronic Computers*, (2):136–148, 1965.
- [Pap77] Christos H Papadimitriou. The euclidean travelling salesman problem is np-complete. *Theoretical computer science*, 4(3):237–244, 1977.
- [Sha08] Harold N Shapiro. *Introduction to the Theory of Numbers*. Courier Corporation, 2008.
- [ST01] Ian Stewart and David Tall. *Algebraic number theory and Fermat’s last theorem*. AK Peters/CRC Press, 2001.
- [SY17] Vikram Sharma and Chee Yap. Robust geometric computation. In *Handbook of Discrete and Computational Geometry, Third Edition*, pages 1189–1223. CRC Press, 2017.
- [Tiw92] P. Tiwari. A problem that is easier to solve on the unit-cost algebraic ram. *J. Complexity*, 8(4):393–397, 1992.
- [UW11] Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in limit-average games. In *International Conference on Concurrency Theory*, pages 482–496. Springer, 2011.
- [VZGG13] Joachim Von Zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

A Muroga’s Proof

Let \mathbb{B} be any bounded discrete domain.

Theorem 13. *Let b be the maximum magnitude of elements in \mathbb{B} . For any LTF $f : \mathbb{B}^m \rightarrow \{\pm 1\}$ realized using weights $w_0, w_1, \dots, w_m \in \mathbb{R}$, there exists an equivalent integer realization with weights $u_0, u_1, \dots, u_m \in \mathbb{Z}$, where $1 \leq |u_0|, \dots, |u_m| \leq \mathcal{O}((m + 1)!b^m)$.*

Proof. Given w_1, \dots, w_n, θ , let X be that subset of \mathbb{B}^n for which $\sum_{i=1}^n w_i x_i \geq 1$. Now consider the linear program with variables z_1, \dots, z_n and the following $|\mathbb{B}|^n$ constraints

$$\begin{aligned} \sum_{i=1}^n z_i x_i &\geq 1 && \text{for } (x_1, \dots, x_n) \in X \\ \sum_{i=1}^n z_i x_i &\leq -1 && \text{for } (x_1, \dots, x_n) \notin X \end{aligned}$$

Clearly, this is a feasible LP because $z_i = w_i/\theta$ is a feasible solution. Therefore it should also have a solution that is a vertex, i.e., a solution with n tight LP constraints of the form $\sum_{i=1}^n z_i x_i = \pm 1$. We can solve such a system of linear equations $X\mathbf{z} = \mathbf{b}$ where $X \in \mathbb{B}^{n \times n}$ and $\mathbf{b} \in \{\pm 1\}^n$. By Cramer's rule, this should give a solution which is the ratio of two determinants. An upper bound on the determinant (and hence any minor) of X is $\mathcal{O}(n!b^n)$. This gives a rational solution for z_i . By clearing the denominators, we get the claimed integer solution. It is clear that such a solution will satisfy all the constraints since it is a vertex of the polytope. \square