Approximate Version of Interval Computation Is Still NP-Hard*

Vladik Kreinovich and Olga Kosheleva
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@utep.edu,olgak@utep.edu
†

Abstract

It is known that, in general, the problem of computing the range of a given polynomial on given intervals is NP-hard. For some NP-hard optimization problems, the approximate version – e.g., if we want to find the value differing from the maximum by no more than a factor of 2 – becomes feasible. Thus, a natural question is: what if instead of computing the exact range, we want to compute the enclosure which is, e.g., no more than twice wider than the actual range? In this paper, we show that this approximate version is still NP-hard, whether we want it to be twice wider or k times wider, for any k.

Keywords: interval computations, NP-hardness, approximate solutions
AMS subject classifications: 65G40, 03D15, 68Q15, 68Q17

1 Formulation of the Problem

Need for interval computations. In practice, we often need to estimate the value of a difficult-to-measure quantity y by using its known relation $y = f(x_1, \ldots, x_n)$ with easier-to-measure quantities $x_1, \ldots, x_n$. This relation is usually described by a continuous function $f(x_1, \ldots, x_n)$.

Measurements are never 100% accurate. The measurement result $\widetilde{x}_i$ is, in general, different from the actual (unknown) value x_i of the corresponding quantity. Often, the only information that we have about the measurement error $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ is the upper bound Δ_i on its absolute value: $|\Delta x_i| \le \Delta_i$; see, e.g., [11]. In this case, once we know the measurement result $\widetilde{x}_i$, the only information that we have about the actual value x_i is that this value belongs to the interval $[\underline{x}_i, \overline{x}_i]$, where $\underline{x}_i = \widetilde{x}_i - \Delta_i$ and $\overline{x}_i = \widetilde{x}_i + \Delta_i$.

*Submitted: June 4, 2020; Revised: May 30, 2021; Accepted: June 3, 2021.
†This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).
The authors are thankful to the anonymous referees for valuable suggestions.

For different values x_i from these intervals, we get, in general, different values of $y = f(x_1, \ldots, x_n)$. It is therefore important to find the range of possible values of y, i.e., find the interval

$$[\underline{y}, \overline{y}] = f([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]) = \{f(x_1, \ldots, x_n) : x_i \in [\underline{x}_i, \overline{x}_i]\}. \tag{1}$$

The problem of computing this range based on the function $f(x_1, \ldots, x_n)$ and intervals $[\underline{x}_i, \overline{x}_i]$ is known as the main problem of *interval computations*; see, e.g., [5, 7, 8].

The main problem of interval computations is known to be NP-hard. It is known that the main problem of interval computations is NP-hard already for polynomials $f(x_1, \ldots, x_n)$. This result was first proven by A. A. Gaganov in [3, 4]; see also [6].

Specifically, the input to this problem consists of:

- a polynomial $f(x_1, \ldots, x_n)$ and
- n intervals $[\underline{x}_i, \overline{x}_i]$ with rational endpoints $\underline{x}_i$ and $\overline{x}_i$.

When we say that a polynomial is given, we mean that we are given an expression obtained from the variables and rational-valued constants by using addition, subtraction, and multiplication. For example, $(1 + x_1) \cdot (1 + x_2) \cdot (1.2 - x_3)$ is such an expression.

The fact that this problem is NP-hard means that – unless P = NP, which most computer scientists believe to be impossible – no polynomial-time (feasible) algorithm can solve all the instances of the interval computation problem. Since no polynomial-time algorithm can always compute the exact range, the currently used polynomial-time algorithms compute *enclosures*, i.e., intervals $[\underline{Y}, \overline{Y}]$ that contain (enclose) the desired range: $[\underline{y}, \overline{y}] \subseteq [\underline{Y}, \overline{Y}]$.

What about an approximate version of this problem? The main problem of interval computation is, in effect, an optimization problem: $\underline{y}$ is the minimum of the functions $f(x_1, \ldots, x_n)$ under the constraints $x_i \in [\underline{x}_i, \overline{x}_i]$, while $\overline{y}$ is the corresponding maximum.

It is known that for many NP-hard optimization problems, their approximate versions can be solved by polynomial-time algorithms; see, e.g., [1]. For example, it is known that the following *knapsack* optimization problem is NP-hard. We are given the prices $p_1, \ldots, p_n$ of n items, their weights $w_1, \ldots, w_n$, and the knapsack's capacity W. We need to find, among all selections $S \subseteq \{1, \ldots, n\}$ that can fit into the knapsack (i.e., for which $\sum\limits_{i \in S} w_i \le W$), the selection with the largest possible overall price $\sum\limits_{i \in S} p_i$. Interestingly, for every $k < 1$, there are polynomial-time algorithms for finding a selection for which the overall price is larger than k times the maximum.

Similar polynomial-time algorithms are known for approximate versions of many other NP-hard optimization problems. A natural question is: can an approximate version of interval computations be solved by a polynomial-time algorithm? Gaganov's result [3, 4] actually shows that computing an enclosure that approximates a bound by *any* given *additive* constant and *limited multiplicative* factor is still NP-hard.

In this paper, we provide a further strengthening of Gaganov's result: namely, we prove that computing an enclosure that approximates a bound by *any* given *multiplicative* factor is still NP-hard.

2 Main Result

Proposition. *For any $k > 1$, the following problem is NP-hard:*

- *given: a polynomial $f(x_1, \ldots, x_n)$ and rational-valued intervals $[\underline{x}_i, \overline{x}_i]$, $i = 1, \ldots, n$,*
- *compute an enclosure $[\underline{Y}, \overline{Y}]$ for the range (1) whose width is no more than k times larger than the width of the actual range $[\underline{y}, \overline{y}]$: $\overline{Y} - \underline{Y} \le k \cdot (\overline{y} - \underline{y})$.*

Proof.

1°. By definition, a problem is NP-hard if every problem from the class NP can be reduced to this problem; see, e.g., [6, 10]. Thus, a usual way to prove that a problem is NP-hard is to show that a known NP-hard problem P_0 can be reduced to this problem. Indeed, in this case, every problem from the class NP can be reduced to P_0, and since P_0 can be reduced to our problem, we can thus conclude that every problem from the class NP can be reduced to our problem as well.

As a known NP-hard problem P_0, we will consider the following *partition problem*:

- given positive integers $s_1, \ldots, s_n$,
- find values $\varepsilon_i \in \{-1, 1\}$ for which $\sum\limits_{i=1}^{n} \varepsilon_i \cdot s_i = 0$.

Equivalently, we want to divide the given set of positive integers into two parts whose sums are equal: the first part is formed by integers s_i with $\varepsilon_i = -1$, the second part is formed by the remaining integers.

2°. We want to reduce a given instance of the partition problem – which is characterized by the sequence of positive integers $s_1, \ldots, s_n$ – to an instance of our problem.

For this purpose, let us first consider, for each instance $(s_1, \ldots, s_n)$, an auxiliary problem – the problem of estimating the range of the variance

$$v(x_1, \ldots, x_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \right)^2 \tag{2}$$

when $x_i \in [-s_i, s_i]$. This auxiliary problem – which is known to be NP-hard [2, 9] – will later be used to form the desired instance of our problem.

The value of the variance is always non-negative, and 0 is attained when all x_i are 0s. Thus, the lower endpoint $\underline{v}$ of the range $[\underline{v}, \overline{v}]$ of the range of the function $v(x_1, \ldots, x_n)$ is equal to 0.

Here, $x_i^2 \le s_i^2$, so $\frac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i^2 \le S \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum\limits_{i=1}^{n} s_i^2$, thus $v(x_1, \ldots, x_n) \le S$. When the corresponding instance of problem P_0 has a solution ε_i, then for $x_i = \varepsilon_i \cdot s_i$, the value S is attained: $v(x_1, \ldots, x_n) = S$. Thus, in this case, $\overline{v} = S$.

Let us show that if the corresponding instance of the problem P_0 does not have a solution, then we have $\overline{v} < S - \frac{1}{4n^2}$. We will prove this by contraposition: namely, we will prove that if $\overline{v} \ge S - \frac{1}{4n^2}$, then the corresponding instance of the problem P_0 has a solution. Indeed, every continuous function attains its maximum at some point in a compact set – in particular, in a box $[-s_1, s_1] \times \ldots \times [-s_n, s_n]$. Thus, there exists a tuple $(x_1, \ldots, x_n)$ for which $v(x_1, \ldots, x_n) \ge S - \frac{1}{4n^2}$, thus

$$S \le v(x_1, \ldots, x_n) + \frac{1}{4n^2}. \tag{3}$$

Then, for each i from 1 to n, we have

$$\frac{1}{n} \cdot s_i^2 + \frac{1}{n} \cdot \sum_{j \neq i} s_j^2 = S \leq \frac{1}{n} \cdot \sum_{j=1}^{n} x_j^2 - \left(\frac{1}{n} \cdot \sum_{j=1}^{n} x_j \right)^2 + \frac{1}{4n^2}.$$

Since $x_j^2 \leq s_j^2$ for all $j \neq i$, we thus get

$$\frac{1}{n} \cdot s_i^2 + \frac{1}{n} \cdot \sum_{j \neq i} s_j^2 \leq \frac{1}{n} \cdot x_i^2 + \frac{1}{n} \cdot \sum_{j \neq i} s_j^2 + \frac{1}{4n^2},$$

hence

$$\frac{1}{n} \cdot s_i^2 \leq \frac{1}{n} \cdot x_i^2 + \frac{1}{4n^2}$$

and

$$x_i^2 \leq s_i^2 \leq x_i^2 + \frac{1}{4n}$$

and so

$$0 \leq s_i^2 - x_i^2 = s_i^2 - |x_i|^2 = (s_i - |x_i|) \cdot (s_i + |x_i|) \leq \frac{1}{4n}.$$

Here, s_i is a positive integer, so $s_i \geq 1$ hence $s_i + |x_i| \geq 1$ and thus,

$$0 \leq s_i - |x_i| \leq \frac{1}{4n} \cdot \frac{1}{s_i + |x_i|} \leq \frac{1}{4n}.$$

The right-hand side is smaller than 1, so, for $\varepsilon_i = \mathrm{sign}(x_i)$ (which is 1 if $x_i > 0$, -1 if $x_i < 0$, and 0 if $x_i = 0$), we get

$$|s_i \cdot \varepsilon_i - x_i| \leq \frac{1}{4n \cdot (s_i + |x_i|)} \leq \frac{1}{4n}.$$

Thus,

$$\left| \sum_{i=1}^{n} s_i \cdot \varepsilon_i - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=1}^{n} |s_i \cdot \varepsilon_i - x_i| \leq n \cdot \frac{1}{4n} = \frac{1}{4}. \tag{4}$$

Also, from (3), taking into account that $x_i^2 \leq s_i^2$, we conclude that

$$\frac{1}{n} \cdot \sum_{i=1}^{n} s_i^2 \leq \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \right)^2 + \frac{1}{4n^2} \leq \frac{1}{n} \cdot \sum_{i=1}^{n} s_i^2 - \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \right)^2 + \frac{1}{4n^2},$$

thus

$$\left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \right)^2 \leq \frac{1}{4n^2}$$

hence

$$\left| \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \right| \leq \frac{1}{2n},$$

thus

$$\left| \sum_{i=1}^{n} x_i \right| \leq \frac{1}{2}. \tag{5}$$

From (4) and (5), we conclude that

$$\left| \sum_{i=1}^{n} s_i \cdot \varepsilon_i \right| \leq \left| \sum_{i=1}^{n} s_i \cdot \varepsilon_i - \sum_{i=1}^{n} x_i \right| + \left| \sum_{i=1}^{n} x_i \right| \leq \frac{1}{4} + \frac{1}{2} < 1.$$

Since the sum $\sum_{i=1}^{n} s_i \cdot \varepsilon_i$ is an integer, this means that this integer is equal to 0, i.e., that the values ε_i indeed solve the given instance of the problem P_0.

3°. Now, let us consider a polynomial

$$f(x_1, \ldots, x_n) = (v(x_1, \ldots, x_n))^N = v(x_1, \ldots, x_n) \cdot \ldots \cdot v(x_1, \ldots, x_n) \quad (N \text{ times}),$$

where N is such that

$$\left(1 - \frac{1}{4n^2 \cdot S}\right)^N < \frac{1}{k},$$

e.g.,

$$N = \left\lceil \frac{\ln(k)}{-\ln\left(1 - \dfrac{1}{4n^2 \cdot S}\right)} \right\rceil + 1.$$

Asymptotically, $N \sim \text{const} \cdot n^2 \cdot S$, so the length of this polynomial's description is bounded by a polynomial of n. Thus, the reduction of the partition problem to our problem is polynomial-time.

If the original instance of the problem P_0 has a solution, then the range $[\underline{y}, \overline{y}]$ of the function $f(x_1, \ldots, x_n)$ is equal to $\left[0, S^N\right]$, and thus, the width of any enclosure $[\underline{Y}, \overline{Y}]$ for this range is at least S^N.

If the original instance has no solutions, then the range $[\underline{y}, \overline{y}]$ is contained in $\left[0, \left(S - \frac{1}{4n^2}\right)^N\right]$, so its width is smaller than or equal to

$$\left(S - \frac{1}{4n^2}\right)^N = S^N \cdot \left(1 - \frac{1}{4n^2 \cdot S}\right)^N < \frac{1}{k} \cdot S^N.$$

Since the width of the enclosure $[\underline{Y}, \overline{Y}]$ is no more than k times the width of the actual range, this width is thus smaller than S^N.

So, if we had an algorithm computing such a no-more-than-k-times wider enclosure $[\underline{Y}, \overline{Y}]$, we would be able to tell whether the original instance of the problem P_0 has a solution or not:

- if $\overline{Y} - \underline{Y} \geq S^N$, then the original instance has a solution,
- otherwise, the original instance does not have a solution.

Thus, we reduced the NP-hard problem P_0 to our problem, hence our problem is also NP-hard.

The proposition is proven.

Remaining open problems. The interval computation problem is NP-hard even if we limit ourselves to quadratic polynomials. In our proof that an approximate version is NP-hard we used polynomials of arbitrary degrees. What if we limit ourselves to quadratic polynomials only? Will the problem still be NP-hard for all k?

Similar questions can be asked about other – non-polynomial – situations when computing the exact range is NP-hard. For example, it is known that if we only know intervals of possible values of all components a_{ij} of a matrix, then computing the range of possible eigenvalues is NP-hard. What if we consider enclosures for this range which are no more than k times wider than the actual range? Will the problem still be NP-hard?

References

[1] G Ausiello, P. Crescenzi, and M. Protasi, "Approximate solutions of NP optimizaton problems", *Theoretical Computer Science*, 1995, Vol. 150, pp. 1–55.

[2] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, "Computing variance for interval data is NP-hard", *ACM SIGACT News*, 2002, Vol. 33, No. 2, pp. 108–118.

[3] A. A. Gaganov, *Computational complexity of the range of the polynomial in several variables*, Leningrad University, Math. Department, M.S. Thesis, 1981 (in Russian).

[4] A. A. Gaganov, "Computational complexity of the range of the polynomial in several variables", *Cybernetics*, 1985, pp. 418–421.

[5] L. Jaulin, M. Kiefer, O. Dicrit, and E. Walter, *Applied Interval Analysis*, Springer, London, 2001.

[6] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.

[7] G. Mayer, *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.

[8] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[9] H. T. Nguyen, V. Kreinovich, B. Wu, and G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Springer Verlag, Berlin, Heidelberg, 2012.

[10] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Massachusetts, 1994.

[11] S. G. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, Springer, New York, 2005.