

How to Deal with Infinities: Why

$$x_i / \sqrt{1 + r^2}?$$

Jeffrey L. Vanarsdall, Dario A. Vazquez, and Vladik Kreinovich

Department of Computer Science, University of Texas at El Paso

500 W. University, El Paso, Texas 79968, USA

jlvanarsdall@miners.utep.edu, davazquez5@miners.utep.edu, vladik@utep.edu

1. Formulation of the problem

- Most effective numerical techniques assume that the set of possible alternatives is bounded.
- This includes optimization techniques and techniques for solving systems of nonlinear equations.
- However, in practice, we sometimes have situations in which no such bounds are known.
- A natural idea is to apply some 1-1 transformation T of the whole unbounded n -dimensional space into a bounded domain.
- This way, we will be able to use bounded-domain methods to solve the correspondingly transformed problems.
- There are many possible transformations.
- Which of them should we choose?

2. Formulation of the problem (cont-d)

- In [1], it was shown that good results appear if we use the following transformation:

$$x_i \mapsto T_0(x) = x_i / \sqrt{1 + r^2}, \text{ where } r^2 \stackrel{\text{def}}{=} \sum_i x_i^2.$$

- A natural question is: why this particular transformation was – empirically – most successful?
- In this talk, we provide a possible answer to this question.

3. First requirement: the transformation T should be rotation-invariant

- Let us start enumerating reasonable requirements on T .
- Many practical problems are invariant with respect to rotations R , and this invariance helps to solve them.
- It is therefore reasonable to require that T preserve all rotations, i.e., that:
 - if x is invariant under rotation R , then $T(x)$ should be similarly invariant, and,
 - more generally, if $y = R(x)$, then we should have $T(y) = R(T(x))$.

4. The transformation T should be rotation-invariant (cont-d)

- In particular, each vector x is invariant with respect to all rotations around x .
- Thus, $T(x)$ should be similarly invariant – which means that $T(x)$ should be parallel to x : $T(x) = c \cdot x$ for some scalar c .
- Every two vectors of the same length can be rotated into each other, so c should only depend on the length r of the vector: $x_i \mapsto x_i \cdot c(r)$.

5. Second requirement: T should be fast to compute

- The fastest are hardware supported operations, and only arithmetic operations are hardware supported.
- Computing r requires taking a square root, so we better describe c as depending on r^2 – which is mathematically the same, so $x_i \mapsto x_i \cdot c(r^2)$.
- Let us show that we cannot use only arithmetic operations to compute $c(r^2)$.
- Indeed, then, as one can show by induction, we get a rational function, i.e., a ratio $P(r^2)/Q(r^2)$ of two polynomials.
- If the degree $\deg(P)$ of P is $\geq \deg(Q)$, then for $r \rightarrow \infty$, we have $c(r^2) \sim r^{2a}$ for some $a \geq 0$.
- So $x_i \cdot c(r^2) \rightarrow \infty$, while we want a bounded domain.

6. T should be fast to compute (cont-d)

- If $\deg(P) < \deg(Q)$, then for $r \rightarrow \infty$, we have $c(r^2) \sim r^{-2a}$ for some $a \geq 1$.
- Then $x_i \cdot c(r^2) \rightarrow 0$ when $r \rightarrow \infty$, and we also have $x_i \cdot c(r^2) \rightarrow 0$ when $r \rightarrow 0$, so T is not 1-1.
- Thus, we need to use at least one non-arithmetic operation.
- Square root – which is part of the definition of r – is a natural choice.
- So, the next try is $x_i \mapsto x_i \cdot \sqrt{C(r^2)}$ for some rational function $C(r^2)$.
- Computing $x_i/\sqrt{1+r^2}$ requires – besides computing the square root – one addition and one division.
- We need at least one division – otherwise, C will be a polynomial and we will have $T(x) \rightarrow \infty$, while we want a bounded domain.

7. T should be fast to compute (cont-d)

- If we only have one division and no other operations, then the only choice is $T(x) = x_i/\sqrt{r^2} = x_i/r$ which is not 1-1: $T(2x) = T(x)$.
- Thus, we need at least one operation in addition to division.
- So indeed operation from [1] is the fastest to compute (and one can show that no other operation is as fast).

8. Reference

- H. Schichl, A. Neumaier, M. C. Markót, and F. Domes, “On solving mixed-integer constraint satisfaction problems with unbounded variables”, In: C. Gomes and M. Sellmann (eds.), *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Springer, Berlin, Heidelberg, 2013, pp. 216–233.

9. Acknowledgments

This work was supported in part by:

- National Science Foundation grants 1623190, HRD-1834620, HRD-2034030, and EAR-2225395;
- AT&T Fellowship in Information Technology;
- program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and
- a grant from the Hungarian National Research, Development and Innovation Office (NRDI).