3-1-2022

Why Optimization Is Faster than Solving Systems of Equations: A Qualitative Explanation

Siyu Deng
The University of Texas at El Paso, sdeng@miners.utep.edu

Bimal K. C
The University of Texas at El Paso, bkc@miners.utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:

Technical Report: UTEP-CS-22-44

Why Optimization Is Faster than Solving Systems of Equations: A Qualitative Explanation

Siyu Deng, Bimal K C, and Vladik Kreinovich,

Abstract Most practical problems lead either to solving a system of equation or to optimization. From the computational viewpoint, both classes of problems can be reduced to each other: optimization can be reduced to finding points at which all partial derivatives are zeros, and solving systems of equations can be reduced to minimizing sums of squares. It is therefore natural to expect that, on average, both classes of problems have the same computational complexity – i.e., require about the same computation time. However, empirically, optimization problems are much faster to solve. In this paper, we provide a possible explanation for this unexpected empirical phenomenon.

1 Formulation of the problem

Practical problems: a general description. In many practical situations, we need to make a decision:

- decide what controls to apply,
- decide which proportion of money to invest in stocks and in bonds,
- decide the proper dose of a medicine, etc.

Siyu Deng
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: sdeng@miners.utep.edu

Bimal K C
Computational Science Program, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: bkc@miners.utep.edu

Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: vladik@utep.edu

What are possible options. Let us describe such problems in precise terms. Let $x_1,\dots,x_n$ be parameters describing possible decisions.

We usually have some constraints on the values of these parameters. Many of these constraints are equalities: $f_i(x_1,\dots,x_n)=0$ for some functions $f_i(x_1,\dots,x_n)$, $1\le i\le m$.

Sometimes, there is only one possible solution. In some practical situations, there are so many constraints that these constraints uniquely determine the values x_i. In this case, to find x_i, we need to solve the corresponding system of equations

$$f_1(x_1,\dots,x_n)=0,$$

$$\dots,$$

$$f_m(x_1,\dots,x_n)=0.$$

Sometimes, there are many possible solutions. In other situations, constraints do not determine the solution uniquely. In this case, we must select the best of possible solutions.

Usually, the quality of a possible solution $x_1,\dots,x_n$ can be described in numerical terms, as $f(x_1,\dots,x_n)$. The corresponding function $f(x_1,\dots,x_n)$ is known as *objective function*. Thus, we must select the possible solution that maximizes the value of the objective function.

In other words, we need to solve an optimization problem.

Unconstrained vs. constrained optimization. In some cases, we do not have any constraints. In such cases, we need to find the values x_i that maximize the given function $f(x_1,\dots,x_n)$.

In other practical situations, we need to optimize the objective function $f(x_1,\dots,x_n)$ under given constraints

$$f_1(x_1,\dots,x_n)=0,\dots,f_m(x_1,\dots,x_n)=0.$$

Lagrange multiplier method reduces this problem to unconditional optimization of the auxiliary function

$$F(x_1,\dots,x_n)\stackrel{\text{def}}{=}f(x_1,\dots,x_n)+\sum_{i=1}^{m}\lambda_i\cdot f_i(x_1,\dots,x_n).$$

Summarizing. Practical problems lead to two types of mathematical problems.

- Some problems lead to solving systems of equations:

$$f_1(x_1,\dots,x_n)=0,\dots,f_m(x_1,\dots,x_n)=0.$$

- Some problems lead to finding the values $x_1,\dots,x$ for which the given function $F(x_1,\dots,x_n)$ attains its largest value.

Which of these two types of problems is, in general, easier to solve? It is known that these two problems can be reduced to each other.

- Optimization $F \to \max$ can be reduced to solving a system of equations obtained by equating all partial derivatives to 0:

$$\frac{\partial F}{\partial x_i} = 0.$$

- Solving a system of equations $f_1(x_1, \ldots, x_n) = 0, \ldots, f_m(x_1, \ldots, x_n) = 0$ is equivalent to minimizing the sum

$$\sum_{i=1}^{m} (f_i(x_1, \ldots, x_n))^2.$$

Because of this possible mutual reduction, one would expect their computational complexity to be comparable.

Surprising empirical fact and the resulting challenge. In spite of this expectation, empirically, in general, optimization problems are faster-to-solve; see, e.g., [1]. Thus, we have the following challenge:

We have two types of problems: solving systems of equations and optimization. Because of the possible mutual reduction, one would expect their computational complexity to be comparable. However, empirically, in general, optimization problems are faster-to-solve.

How can we explain this unexpected empirical fact?

2 Possible explanation

General observation about relative computation time. To provide an explanation, let us recall cases when some class of problem is computationally easier.

In each computation problem, there is one or more inputs and desired outputs. The output of computations is usually uniquely determined by the inputs. In mathematical terms, this means that the output is a function of the inputs. In this sense, every computation is a computation of the value of an appropriate function.

In general:

- functions of two variables take more time to compute than functions of one variable,
- functions of three variables take more time to compute than functions of two variables, etc.

In other words, the more inputs we have, the more computation time the problem requires.

Let us apply this general observation to our problem. For both optimization problem and the problem of solving a system of equation, the inputs are functions. The difference is in how many functions form the input.

- To describe an optimization problem we need to describe only one function $F(x_1,\ldots,x_n)$; this function is to be maximized.
- On the other hand, to describe a system of m equations with n unknowns,
- we need to describe m functions

$$f_1(x_1,\ldots,x_n),\ldots,f_m(x_1,\ldots,x_n).$$

So:

- The input to an optimization problem is a single function.
- The input to a solving-system-of-equations problem consists of several functions.

Thus, solving systems of equations requires more inputs than optimization.

So, not surprisingly, optimization problems are, in general, faster to solve. This explains the above empirical fact.

Acknowledgments

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-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

The authors are thankful to all the participants of the 27th Joint NMSU/UTEP Workshop on Mathematics, Computer Science, and Computational Sciences (Las Cruces, New Mexico, April 2, 2022) for valuable suggestions.

References

1. R. L. Muhanna, R. L. Mullen, and M. V. Rama Rao, “Nonlinear Interval Finite Elements for Beams”, *Proceedingds of the Second International Conference on Vulnerability and Risk Analysis and Management (ICVRAM) and the Sixth International Symposium on Uncertainty, Modeling, and Analysis (ISUMA)*, Liverpool, UK, July 13–16, 2014.