Information-based complexity and information-based optimization

J. F. Traub
Department of Computer Science
Columbia University, New York, NY 10027

A. G. Werschulz
Department of Computer and Information Sciences
Fordham University, New York, NY 10023
Department of Computer Science
Columbia University, New York, NY 10027

Technical Report CUCS-011-99

August 21, 2000

Abstract

This is an article that will appear in the *Encyclopedia of Optimization* (Kluwer, 2000). It concerns optimization in two senses. The first is that informaion-based complexity (IBC) is the study of the minimal computational resources to solve continuous mathematical problems. The second is that the computational complexity of optimization problems is one of the areas studied in IBC. We discuss IBC and information-based optimization in turn.

This article concerns optimization in two senses. The first is that *information-based complexity* (IBC) is the study of the minimal computational resources to solve *continuous* mathematical problems. (Other types of mathematical problems are also studied; the problems studied by IBC will be characterized later.) `J.F. Traub` and `A.G. Werschulz` [14] provide an expository introduction to the theory and applications of IBC, with over 400 recent papers and books. A general formulation with proofs can be found in `J.F. Traub`, `G.W. Wasilkowski`, and `H. Woźniakowski` [13].

The second is that the *computational complexity of optimization problems* is one of the areas studied in IBC. `S.A. Vavasis` [16, pg. 135] calls this *information-based optimization*. We will discuss information-based complexity and information-based optimization in turn.

1 Information-based complexity

To introduce computational complexity, we first define the *model of computation*. The model of computation states which operations are permitted and how much they cost. The model of computation is based on two assumptions:

1. We can perform arithmetic operations and comparisons on real numbers at unit cost.

2. We can perform an information operation at cost c. Usually, $c \gg 1$.

We comment on these assumptions. The *real number model* (Assumption 1) is used as an abstraction of the floating-point model typically used in scientific computation. Except for the possible effect of roundoff errors and numerical stability, complexity results will be the same in these two models.

The real number model should be contrasted with the *Turing Machine model*, typically used for discrete problems. The cost of an operation in a Turing Machine model depends on the size of the operands, which is not a good assumption for floating point numbers. For a full discussion of the pros and cons of the Turing Machine and real number models see [14, Chapter 8]. Whether the real number or Turing Machine model is used can make an enormous difference. For example, `L.G. Khachiyan` [3] shows that *linear programming* is polynomial in the Turing Machine model. In 1982, `J.F. Traub` and `H. Woźniakowski` [15] showed that Khachiyan's algorithm is not polynomial in the real number model and conjectured that linear programming is not polynomial in this model. This conjecture is still open.

The purpose of information operations (Assumption 2) is to replace the input by a finite set of numbers. For integration, the information operations are typically function evaluations.

1.1 Computational complexity of high-dimensional integration

We illustrate some of the important ideas of IBC with the example of high-dimensional integration.

We wish to compute the integral of a real-valued function f of d variables over the unit cube in d dimensions. Typically, we have to settle for computing a numerical approximation with an error ε. To guarantee an ε-approximation we have to know some global information about the integrand. We assume that the class F of integrands has smoothness r. One such class is F_r, which consists of those functions having continuous derivatives of order through r, these derivatives satisfying a uniform bound.

A real function of a real variable cannot be entered into a digital computer. We evaluate f at a finite number of points and we call the set of values of f the local information, for brevity *information*, about f. An algorithm combines the function values into a number that approximates the integral.

In the *worst case setting* we want to guarantee an error at most ε for every $f \in F$. The computational complexity, for brevity complexity, is the least cost of computing the integral to within ε for every f. We want to stress that the complexity depends on the problem and ε, but not on the algorithm. Every possible algorithm, whether or not it is known, and all possible points at which the integrand is evaluated are permitted to compete when we consider least possible cost.

It can be shown that if $F = F_r$, then the complexity of our integration problem is of order $\varepsilon^{-d/r}$. If $r = 0$, e.g., if our set of integrands consists of uniformly bounded continuous functions, the complexity is infinite. That is, it is impossible to solve the problem to within ε. Let r be positive and in particular let $r = 1$. Then the complexity is of order ε^{-d}. Because of the exponential dependence on d, we say the problem is computationally intractable. This is sometimes called the *curse of dimensionality.*

We'll compare this d-dimensional integration problem with the well-known *Traveling Salesman Problem* (TSP), an example of a discrete combinatorial problem. The input is the location of the n cities and the desired output is the minimal route; the city locations are usually represented by a finite number of bits. Therefore the input can be exactly entered into a digital computer. The complexity of this problem is unknown but *conjectured* to be exponential in the number of cities. That is, the problem is conjectured to be computationally intractable and many other combinatorial problems are conjectured to be intractable.

Most problems in scientific computation which involve multivariate functions belonging to F_r have been proven computationally intractable in the number of variables in the worst case setting. These include nonlinear equations [10], partial differential equations [19], function approximation [7], integral equations [19], and optimization [5]. Material on the computational complexity of optimization will be presented in the second half of this article.

Very high dimensional integrals occur in many disciplines. For example, problems with dimension ranging from the hundreds to the thousands occur in mathematical finance. Path integrals, which are of great importance in physics, are infinite-dimensional, and therefore invite high dimensional approximations. This motivates our interest in breaking the curse of dimensionality. Since this is a complexity result, we cannot get around it by a clever algorithm. We can try to break the curse by settling for a stochastic assurance rather than a worst case deterministic assurance. Examples of stochastic assurance are provided by the randomized and average case settings which we will consider below. We can also try to break the curse by changing the class of inputs. A good example of this occurs in mathematical finance.

1.2 Mathematical finance mathematical finance

The valuation of financial instruments often requires the calculation of very high dimensional integrals. Dimensions of 360 and higher are not unusual. Furthermore, since the integrals can be very complicated requiring between 10^5 and 10^6

floating point operations per integrand evaluation, it is important to minimize the number of evaluations. Extensive numerical testing shows that these problems do not suffer from the curse of dimensionality. A possible explanation is given by `I. Sloan` and `H. Woźniakowski` [11], who show that the curse can be broken by changing the class of integrands to capture the essence of the mathematical finance problem. See [14, Chapter 4] for a survey of high dimensional integration and mathematical finance.

1.3 General theory

In general, IBC is defined by the assumptions that the *information* concerning the mathematical model is

- partial,
- contaminated, and
- priced.

Referring to the integration example, the mathematical input is the integrand and the information is a finite set of function values. It is *partial* because the integral cannot be recovered from function values. For a partial differential equation the mathematical input consists of the functions specifying the initial value and/or boundary conditions. Generally, the mathematical input is replaced using a finite number of information operations. These operations may be functionals on the mathematical input or physical measurements that are fed into a mathematical model.

In addition to being partial the information is often *contaminated* by, for example, round-off or measurement error (`L. Plaskota` [8]). If the information is partial or contaminated it is impossible to solve the problem exactly. Finally, the information is *priced*. As examples, functions can be costly to evaluate or information needed for oil exploration models can be obtained by setting off shocks. With the exception of certain finite-dimensional problems, such as roots of systems of polynomial equations and problems in numerical linear algebra, the problems typically encountered in scientific computation have information that is partial and/or contaminated and priced.

As part of our study of complexity we investigate *optimal algorithms*, that is, algorithms whose cost is equal or close to the complexity of the problem. This has sometimes led to new solution methods. The reason that we can often obtain the complexity and an optimal algorithm for IBC problems is that partial and/or contaminated information permits arguments at the information level. This level does not exist for combinatorial problems where we usually have to settle for trying to establish a complexity hierarchy and trying to prove conjectures such as $\mathrm{P} \neq \mathrm{NP}$.

A powerful tool at the information level is the notion of the *radius of information*, R. The radius of information measures the intrinsic uncertainty of solving a problem using given information. We can compute an ε-approximation if and only if $R \le \varepsilon$. The radius depends only on the problem being solved and the available information; it is independent of the algorithm. The radius of information is defined in all IBC settings.

2 Information-based optimization

We turn to the application of IBC concepts to information-based optimization.

In their seminal book, `A.S. Nemirovsky` and `D.B. Yudin` [5] study a *constrained optimization* problem. They wish to minimize a nonlinear function subject to nonlinear constraints. Let $f = [f_0, f_1, \ldots, f_m]$, where f_0 denotes the objective function and $f_1, \ldots, f_m$ denote constraints. Let F be the product of $m+1$ copies of F_r. Then

$$\mathrm{comp}(\varepsilon) = \Theta\left(\left(\frac{1}{\varepsilon}\right)^{d/r}\right).$$

Thus this problem suffers from the curse of dimensionality.

Vavasis [16, Chapter 6] reports on the worst-case complexity of minimizing an objective function with box constraints. He assumes objective functions defined on the unit cube in d dimensions and takes F as the class of continuous functions with uniform Lipschitz constant L. For *global minimization*,

$$\mathrm{comp}(\varepsilon) = \Theta\left(\left(\frac{L}{2\varepsilon}\right)^{d}\right).$$

Thus global minimization is intractable.

In contrast to global minimization, the problem of computing a *local minimum* is tractable with suitable conditions on F. Let F consist of continuously differentiable real functions on $[0,1]^d$ whose gradients satisfy a uniform Lipschitz condition with constant M. Then $4d(M/\varepsilon)^2$ function and gradient evaluations are sufficient.

As discussed above, there are two ways one can attempt to break the curse of dimensionality: by settling for a stochastic assurance, or by changing the class of inputs. For the constrained optimization problem, we first describe changing the class of functions, and then turn to weakening the assurance.

Nemirovsky and Yudin [5] take $F = F_{\mathrm{conv}}$ to be the class of *convex* functions that satisfy a Lipschitz condition with a uniform constant on a bounded convex set D. Then

$$\mathrm{comp}(\varepsilon) = \Theta\left(\log\frac{1}{\varepsilon}\right),$$

where the constant in the Θ-notation depends polynomially on the dimension d of D and m, the number of constraints. Thus, convexity breaks the curse of dimensionality.

The worst case deterministic assurance may be weakened to a stochastic assurance; we report on the randomized and average case settings.

Nemirovsky and Yudin [5] show that *randomization* does not break the curse of dimensionality for computing the minimum value of the nonlinear constrained problem. `G.W. Wasilkowski` [17] establishes an even more negative result if an ε-approximation to the value of x that minimizes f_0 is sought. He permits randomization and shows that for all $\varepsilon < \frac{1}{2}$, this problem is unsolvable even if $d = 1$.

The results considered so far use a sequential model of computation. One could also ask about the complexity under a *parallel* model of computation. If we have k processors running in parallel, how much can the computation of the minimum be sped up? Clearly, the best possible speedup is k. `A. Nemirovsky` [6] considers this problem for the case $F = F_{\text{conv}}$, showing that

$$\text{comp}^{\text{par}}(\varepsilon, k) = \Omega\left(\left(\frac{d}{\ln(2kd)}\right)^{1/3} \ln\left(\frac{1}{\varepsilon}\right)\right),$$

where the Ω-constant is independent of k and ε. Hence we find that

$$\frac{\text{comp}(\varepsilon)}{\text{comp}^{\text{par}}(\varepsilon, k)} = O\left(\left(\frac{\ln(2kd)}{d}\right)^{1/3}\right),$$

which is much less than k. Thus parallel computation is not very attractive for this problem.

The average case setting looks more promising than the randomized setting, but since it is technically very difficult, the results to date are quite limited. In the average case setting we want to guarantee that the expected error is at most ε and we minimize the expected cost.

In the *average case setting*, an a priori measure must be placed on F. Typically, this measure is *Gaussian*; in particular, *Wiener measures* are used. Since the distribution of the random variable $\min_x f(x)$ is difficult to obtain, the average case analysis of the global optimization problem is very difficult. Only partial results have been obtained.

Let $d = 1$ and $F \subset C^r[0, 1]$. Assume that F is endowed with the r-fold Wiener measure. `G.W. Wasilkowski` [18] shows that approximately $(\varepsilon^{-1}\sqrt{\ln \varepsilon^{-1}})^{1/(r+1/2)}$ function evaluations suffice. This is better than the worst case, where some $\varepsilon^{-1/r}$ function values are needed.

Stronger results have been obtained for the case of $d = 1$ and $r = 0$, i.e., optimization for continuous scalar functions, equipped with the Wiener measure. `K. Ritter` [9] considers the case of *nonadaptive methods*, showing that

$$\text{comp}^{\text{non}}(\varepsilon) = \Theta\left(\left(\frac{1}{\varepsilon}\right)^2\right).$$

Moreover, the optimal evaluation points are equidistant knots. More recently, `J.M. Calvin` [1] investigates *adaptive methods* for this problem, showing that for any $\delta \in (0, 1)$,

$$\text{comp}^{\text{ad}}(\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(1-\delta)}\right).$$

The study of optimization in the average case setting is a very promising area for future research. Important open problems include:

- obtaining multivariate results,
- obtaining lower bounds,
- obtaining better upper bounds.

We now restrict our attention to the special optimization problem of linear programming (LP), which we discuss in the worst case setting.

In 1979, Khachiyan [3] studied an *ellipsoid algorithm* and proved that LP is polynomial in the Turing Machine model. In 1982, Traub and Woźniakowski [15] showed that the cost of this ellipsoid algorithm is not polynomial in the real-number model, and conjectured that the LP problem is not polynomial in the real-number model. This nicely illustrates the difference between the cost of an algorithm and the complexity of a problem, since the result concerning the cost of the ellipsoid algorithm leaves open the question of problem complexity. The Traub-Woźniakowski conjecture remains open.

A related open question is whether LP can be solved in *strongly polynomial time*. (Note that the underlying models of computation are different: the real-number model versus the Turing Machine model.) This question is also still open, with results known only for special cases. In 1984, `N. Megiddo` [4] showed that LP can be solved in linear time if the number of variables is fixed, while in 1986, `E. Tardos` [12] showed that many LP problems that arise from combinatorial applications can be solved in strongly polynomial time.

We now discuss the computation of *fixed points*, which we include here because the result involves ellipsoid methods. The problem is to compute the fixed point of $f(x)$; that is, to solve the nonlinear equation $x = f(x)$ for any $f \in F$, where F is the class of functions on $[0,1]^d$ having a Lipschitz constant of q, with $q \in (0,1)$.

The simple iteration algorithm $x_{i+1} = f(x_i)$, with $x_0 = 0$, can compute an ε-approximation with at most

$$n_{\mathrm{si}}(\varepsilon, q) = \left\lceil \frac{\ln 1/\varepsilon}{\ln 1/q} \right\rceil$$

evaluations of f. Thus the simple iteration algorithm behaves poorly if q is close to one.

`Z. Huang`, `L. Khachiyan`, and `K. Sikorski` [2] show that an inscribed ellipsoid algorithm computes an ε-approximation with

$$n_{\mathrm{e}}(\varepsilon, q) = O\left(d\left(\ln\frac{1}{\varepsilon} + \ln\frac{1}{1-q}\right)\right)$$

function evaluations. Thus their algorithm is excellent for computing fixed points of functions with q close to unity; that is, almost non-contracting functions.

References

[1] Calvin, J.M.: 'Average performance of adaptive algorithms for global optimization', *Ann. Appl. Prob.* **34** (1997), 711–730.

[2] Huang, Z., Khachiyan, L., and Sikorski, K.: 'Approximating fixed points of weakly contracting mappings', *J. Complexity* **15** (1999), 200–213.

[3] Khachiyan, L.G.: 'A polynomial algorithm in linear programming', *Soviet Math. Dokl.* **20** (1979), 191–194.

[4] Megiddo, N.: 'Linear programming in linear time when the dimension is fixed', *J. ACM* **31** (1984), 114–127.

[5] Nemirovsky, A.S., and Yudin, D.B.: *Problem Complexity and Method Efficiency in Optimization*, Wiley-Interscience, New York, 1983.

[6] Nemirovsky, A. S.: 'On parallel complexity of nonsmooth convex optimization', *J. Complexity* **10** (1994), 451–463.

[7] Novak, E.: *Deterministic and Stochastic Error Bounds in Numerical Analysis*, Vol. 1349 of *Lecture Notes in Mathematics*, Springer-Verlag, New York, 1988.

[8] Plaskota, L.: *Noisy Information and Computational Complexity*, Cambridge University Press, Cambridge, 1996.

[9] Ritter, K.: 'Approximation and optimization on the Wiener space', *J. Complexity* (1990), 337–364.

[10] Sikorski, K.: *Optimal Solution of Nonlinear Equations*, Oxford University Press, Oxford, 2000.

[11] Sloan, I. H., and Woźniakowski, H.: 'When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?', *J. Complexity* **14** (1998), 1–33.

[12] Tardos, E.: 'A strongly polynomial algorithm to solve combinatorial linear programs', *Oper. Res.* **34**, no. 2 (1986), 250–256.

[13] Traub, J.F., Wasilkowski, G.W., and Woźniakowski, H.: *Information-Based Complexity*, Academic Press, New York, 1988.

[14] Traub, J.F., and Werschulz, A.G.: *Complexity and Information*, Cambridge University Press, Cambridge, 1998.

[15] Traub, J. F., and Woźniakowski, H.: 'Complexity of linear programming', *Oper. Res. Lett.* **1** (1982), 59–62.

[16] Vavasis, S.A.: *Nonlinear Optimization: Complexity Issues*, Oxford University Press, New York, 1991.

[17] WASILKOWSKI, G.W.: 'Randomization for continuous problems', *J. Complexity* **5** (1989), 195–218.

[18] WASILKOWSKI, G.W.: 'On average complexity of global optimization problems', *Math. Programming* **57**, no. 2 (Ser. B) (1992), 313–324.

[19] WERSCHULZ, A.G.: *The Computational Complexity of Differential and Integral Equations: An Information-Based Approach*, Oxford University Press, New York, 1991.