Computational Complexity of Continuous Problems

Columbia University Computer Science Department Report CUCS-025-96

Henryk Woźniakowski
University of Warsaw and Columbia University

May 20, 1996

Abstract

Computational complexity studies the intrinsic difficulty of solving mathematically posed problems. Information-based complexity is a branch of computational complexity that deals with continuous problems defined on spaces of multivariate functions. For such problems only approximate solutions are possible to compute. The complexity is defined as the minimal cost needed to compute an approximation with error at most ε. Error and cost can be defined in different settings such as the worst case, average case, probabilistic or randomized settings.

In this paper, we survey recent results on complexity of linear multivariate problems and on path integration. In particular, we show that multivariate integration and approximation are (strongly) tractable in the average case setting for the class of continuous functions equipped with the Wiener sheet measure. This means that their complexity is a polynomial in ε^{-1}.

We consider path integration for the Wiener measure in the worst case and randomized settings. For the class of r times Frechet differentiable functions, the problem is intractable in the worst case setting, whereas it is tractable in the randomized setting and the classical Monte Carlo algorithm is optimal. On the other hand, for the specific class of entire functions, the problem is tractable in the worst case setting and its complexity is proportional to $\varepsilon^{-2/3}$.

1 Introduction

The goal of this paper is to introduce the reader to computational complexity. This is a relatively new and fast developing area of theoretical computer science. Computational complexity studies the intrinsic difficulty of solving mathematically posed problems.

To study complexity we must first define a model of computation. The model states which operations are allowed, what the cost of each operation is, and how computation is performed. Not surprisingly, complexity results depend on the model of computation, and sometimes an apparently innocent change of a model leads to a completely different complexity result.

In discrete computational complexity, the Turing machine model is usually assumed. Roughly speaking, in this model we operate on bits, the cost depends on the size of numbers, and we count how many bit operations are necessary to solve the problem. The Turing machine model is used for discrete problems, and there is a deep theory culminating in the famous question whether P $\neq$ NP, see e.g., [4].

In continuous computational complexity, we study continuous problems. Many scientific phenomena correspond to continuous problems. They are usually solved using fixed precision floating point arithmetic. The cost of floating point operations is independent of the size of the numbers. Furthermore, all arithmetic operations cost about the same to execute. If we ignore rounding errors, floating point arithmetic corresponds to the *real number* model of computation. That is why, for continuous problems, we usually choose the real number model and study computational complexity in this model. For the precise definition of the real number model the reader is referred to [2, 12].

Continuous computational complexity may be split into two branches. The first branch deals with problems for which the information is *complete*. Informally, information may be complete for problems which are specified by a finite number of inputs. Examples include matrix multiplication, and the solution of linear algebraic systems or systems of polynomial equations.

To illustrate this branch of continuous computational complexity, consider the problem of solving linear systems $Ax = b$ with a given $n \times n$ matrix A and a $n \times 1$ given vector b. If n is not too large and the matrix A is dense then we input $n^2 + n$ data given by all coefficients of A and b. Information is then complete.

What is the complexity of solving systems of linear equations? That is, what is the minimal number of arithmetic operations needed to solve $Ax = b$ for an arbitrary nonsingular $n \times n$ matrix A and an arbitrary $n \times 1$ vector b? We do not know exactly the complexity. We only know bounds on it. The lower bound is given by the total number of data and is proportional to n^2. The upper bound is given by the cost of an algorithm that solves the problem. Classical algorithms compute the solution vector $x = A^{-1}b$ using $\Theta(n^3)$ arithmetic operations[1]. Examples of such algorithms include Gaussian elimination and Householder's

[1] By $\Theta(n^3)$ we mean a function which can be bounded from below and from above by a multiple of n^3.

method. However, we can do better. In 1969, Strassen [17] found an algorithm which computes the solution using $\Theta(n^{\log_2 7})$ arithmetic operations. Since $\log_2 7 = 2.81...$, this yields a better upper bound, at least for large n. Today, the best known upper bound is due to Coppersmith and Winograd [3] and it is $\Theta(n^{2.376})$. The constant in the theta notation of the latter bound is, unfortunately, huge.

We stress that problems with complete information may be very hard in the real number model. The first NP-complete problem over the reals was established in [2]. This is the problem of deciding whether a real polynomial of degree 4 in n variables has a real root. Hence, modulo the conjecture $\mathrm{P} \neq \mathrm{NP}$, but this time over the reals, the complexity of the latter problem is *not* polynomial in n.

The second branch of continuous computational complexity is *information-based complexity*, denoted for brevity as IBC. It deals with problems for which the information is *partial*. Typically, IBC studies problems whose input is an element of an infinite-dimensional space. Examples of such problems include multivariate integration or approximation, solution of ordinary or partial differential equations, integral equations, optimization, and solving non-polynomial equations. The input of such problems is often a multivariate function on the reals. Information is usually supplied by a subroutine which computes function values. Using this subroutine finitely many times, we know only partial information about the function. Typically, this partial information is *contaminated* with errors such as round-off errors or measurement errors. Thus, the available information is partial and/or contaminated. Therefore, the original problem can be solved only *approximately*. The goal of IBC is to compute such an approximation at minimal cost. The error and the cost of approximation can be defined in different settings, including the worst case, average case, probabilistic, randomized and mixed settings. The ε-complexity is then defined as the minimal cost of computing an approximation with error at most ε. The reader who wants to find more about IBC is referred to the books and recent surveys [5, 9, 11, 14, 18, 20, 27].

We believe that the readers of this proceedings are mainly interested in solving scientific problems for which only partial information is available. That is why we restrict ourselves in the rest of this paper to IBC issues. To make this paper self-contained, we present an abstract formulation of IBC in Section 2. This abstract formulation is illustrated by a simple example of scalar integration.

We then briefly survey recent results on complexity of linear multivariate problems in Section 3. Many problems in science, engineering, economics and finance are modeled by multivariate problems involving functions of d variables with large or even huge d. For path integration, we even have $d = +\infty$; the approximation of path integrals yields multivariate integration with huge d.

We are interested in the complexity of linear multivariate problems in various settings.

In particular, the complexity depends on the error parameter ε, and on the number d of variables.

In the worst case setting, it is known that many problems are intractable. More specifically, for many problems the complexity is an exponential function of d. This means that for large d the complexity is so huge that it is impossible to solve the problem. This is sometimes called the *curse of dimension.*

We stress that the exponential dependence on d is a complexity result and it is impossible to get around it by designing efficient algorithms. The only way to break the curse of dimension is to weaken the notion of error and/or cost. This can sometimes be done by switching from the worst case setting to another setting. Hence, we wish to examine how complexity depends on ε and d in other settings. If the dependence is polynomial in d and ε^{-1} then the curse of dimension is broken.

For a given setting, we say that a linear multivariate problem is *tractable* if its complexity depends polynomially on d and ε^{-1}. It is called *strongly tractable* if its complexity is independent of d and depends polynomially on ε^{-1}. There are some general results characterizing which linear multivariate problems are tractable or strongly tractable, see, [30]. In particular, multivariate integration and approximation are strongly tractable in the average case setting for the class of continuous functions equipped with the Wiener sheet measure. Specific complexity bounds are given in Section 3.

The final section deals with path integration, see [24, 25]. Usually Monte Carlo algorithms are used to approximate path integrals. We study deterministic algorithms in the worst case setting. Then path integration is tractable (i.e., its complexity is polynomial in ε^{-1}) if the class of integrands consists of entire functions. Finite smoothness of integrands is not enough if the measure of the path integration problem is supported on an infinite dimensional subspace. In this case, the classical Monte Carlo algorithm is almost optimal in the randomized setting. We conclude with a remark on Feynman-Kac path integrals.

2 Basic Concepts of IBC

In this section[2] we present an abstract formulation of IBC and illustrate it by a simple example. A proof technique which leads to tight complexity bounds for some problems will also be indicated. Let

$$S : F \to G,$$

[2]This section is based on Section 2 of [19].

where F is a subset of a linear space and G is a normed linear space over the real or complex field. We wish to approximate $S(f)$ for all f from F.

Let $U(f)$, where $U: F \to G$, denote a computed approximation to $S(f)$ for $f \in F$. We now explain how the approximation U can be constructed. To do this we first need to discuss the concept of information.

The basic assumption of IBC is that, in general, we do not have full knowledge of an element f since typically f is a multivariate function and it cannot be represented exactly on a digital computer. Instead, it is assumed that we can gather some knowledge about f by computations of the form $L(f)$, where $L: F \to H$ for some set H.

Let Λ denote a class of permissible information operations L. That is, $L \in \Lambda$ iff $L(f)$ can be computed for each f from F. For example, if F is a set of functions then Λ is often taken as a set of L consisting of function evaluations, $L(f) = f(x)$, $\forall f \in F$, for some x from the domain of f. Such Λ is denoted by Λ^{std}. If the class Λ is taken as a set of all linear functionals L then it is denoted by Λ^{all}. Let

$$N(f) = [L_1(f), L_2(f), \ldots, L_n(f)], \qquad L_i \in \Lambda, \quad \forall\, f \in F, \tag{1}$$

be the computed information about f. We stress that the L_i as well as the number n can be chosen adaptively. That is, the choice of L_i may depend on the already computed $L_1(f), L_2(f), \ldots, L_{i-1}(f)$. The number n may also depend on the computed $L_i(f)$. (This permits arbitrary termination criteria.)

$N(f)$ is called the information about f, and N the information operator. In general, N is many-to-one, and thus knowing $y = N(f)$ it is impossible to recover the element f. For this reason, the information N is called *partial*.

The approximation $U(f)$ is constructed by combining the computed information $N(f)$. That is, $U(f) = \phi(N(f))$, where $\phi : N(F) \to G$. A mapping ϕ is called an algorithm. The approximation U can thus be identified with the pair (N, ϕ), where N is an information operator and ϕ an algorithm that uses the information N.

We illustrate these concepts by an example.

Example: Integration

Let F be a class of functions $f : [0,1] \to \mathbb{R}$ that satisfy a Lipschitz condition with constant q,

$$|f(x) - f(y)| \leq q\,|x - y|, \quad \forall\, x, y \in [0,1].$$

Let $G = \mathbb{R}$ and

$$S(f) = \int_0^1 f(t)\, dt.$$

The class Λ is a collection of $L : F \to \mathbb{R}$, such that for some x from $[0,1]$, $L(f) = f(x)$, $\forall f \in F$. The information N is given by

$$N(f) = [f(x_1), f(x_2), \ldots, f(x_n)]$$

with the points x_i and the number n adaptively chosen. The approximation U is now of the form $U(f) = \phi(N(f)) = \phi(f(x_1), f(x_2), \ldots, f(x_n))$. An example of an algorithm ϕ is a quadrature given by $U(f) = \phi(N(f)) = \sum_{i=1}^{n} a_i\, f(x_i)$ for some numbers a_i. $\square$

We now present a model of computation. It is defined by two postulates:

- We are charged for each information operation. That is, for every $L \in \Lambda$ and for every $f \in F$, the computation of $L(f)$ costs $\mathbf{c}$, where $\mathbf{c}$ is positive and fixed, independent of L and f.
- Let Ω denote the set of permissible combinatory operations including the addition of two elements in G, multiplication by a scalar in G, arithmetic operations, comparison of real numbers, and evaluations of certain elementary functions. We assume that each combinatory operation is performed exactly with unit cost.

In particular, this means that we use the real number model, where we can perform operations on real numbers exactly and at unit cost.

We now discuss the cost of the approximations $U(f) = \phi(N(f))$. Let $\mathrm{cost}(N, f)$ denote the cost of computing the information $N(f)$. Note that $\mathrm{cost}(N, f) \geq \mathbf{c}\, n$, and the inequality may occur since adaptive selection of L_i and n may require some combinatory operations. If $N(f)$ cannot be computed by using n information operations and a finite number of operations from Ω, then $\mathrm{cost}(N, f) = +\infty$.

Knowing $y = N(f)$, we compute $U(f) = \phi(y)$ by combining the information $L_i(f)$. Let $\mathrm{cost}(\phi, y)$ denote the number of combinatory operations from Ω needed to compute $\phi(y)$. If $\phi(y)$ cannot be computed by using a finite number of operations from Ω, then $\mathrm{cost}(\phi, y) = +\infty$.

The cost of computing $U(f)$, $\mathrm{cost}(U, f)$, is given by

$$\mathrm{cost}(U, f) \;=\; \mathrm{cost}(N, f) + \mathrm{cost}(\phi, N(f)).$$

We now define the concepts of error and cost of the approximation U. The definitions of error and cost depend on the setting. We first discuss three settings: worst case, average case and probabilistic. Then we turn to a randomized setting.

In the worst case setting, the error and cost of U are defined as

$$\begin{aligned} e(U) &= \sup_{f\in F} \|S(f) - U(f)\|, \\ \mathrm{cost}(U) &= \sup_{f\in F} \mathrm{cost}(U, f). \end{aligned}$$

In the average case and probabilistic settings, we assume that the set F is equipped with a probability measure μ. In the average case setting the error and cost of U are defined as

$$\begin{aligned} e(U) &= \left(\int_F \|S(f) - U(f)\|^2 \mu(df)\right)^{1/2}, \\ \mathrm{cost}(U) &= \int_F \mathrm{cost}(U, f)\, \mu(df). \end{aligned}$$

In the probabilistic setting, we assume that we are given a number $\delta \in [0,1]$, and the error and cost of U are defined as

$$\begin{aligned} e(U) &= \inf\left\{ \sup_{f\in F-A} \|S(f) - U(f)\| \;:\; A \text{ such that } \mu(A) \le \delta \right\}. \\ \mathrm{cost}(U) &= \sup_{f\in F} \mathrm{cost}(U, f). \end{aligned}$$

We now discuss a randomized setting. In this setting the approximation U is defined by a random selection of information and algorithm. More precisely, let ρ be a probability measure on a set T. Then for each $t \in T$ we select information N_t and an algorithm ϕ_t, and compute $U_t(f) = \phi_t(N_t(f))$. Here t is a random variable distributed according to the measure ρ. Random information N_t is of the form (1) with randomly chosen L_i and n. A random algorithm is $\phi_t : N_t(F) \to G$. The approximation U can now be identified as the 4-tuple, $U = (N, \phi, T, \rho)$.

The error of U in the randomized setting is defined as

$$e(U) = \sup_{f\in F} \int_T \|S(f) - U_t(f)\|\, \rho(dt),$$

In the randomized setting, the cost of $U_t(f) = \phi_t(N_t(f))$ is defined as above, and then the cost of U is defined as

$$\mathrm{cost}(U) = \sup_{f\in F} \int_T \mathrm{cost}(U_t, f)\, \rho(dt).$$

We illustrate the randomized setting by continuing the integration example.

Example (continued) Consider the classical Monte Carlo algorithm

$$U_t(f) = \frac{1}{n}\sum_{i=1}^{n} f(t_i),$$

with uniformly distributed points t_i. That is, $t = [t_1, t_2, \ldots, t_n] \in T = [0,1]^n$ and ρ is the uniform distribution over the unit n dimensional cube. In this case,

$$N_t(f) = [f(t_1), f(t_2), \ldots, f(t_n)]$$

is random information with randomly chosen points t_i and deterministically chosen n. The algorithm ϕ_t is deterministic and equal to $\phi_t(y_1, y_2, \ldots, y_n) = \frac{1}{n}\sum_{i=1}^{n} y_i$. The error of U is proportional to $n^{-1/2}$ and the cost of U is proportional to n. $\square$

We are ready to define the computational complexity of IBC problems The basic notion is the ε-complexity which is defined as the minimal cost of *all* U with error at most ε,

$$\mathrm{comp}(\varepsilon) \;=\; \inf\left\{\mathrm{cost}(U) : \; U \text{ such that } e(U) \le \varepsilon\right\}.$$

Here we use the convention that the infimum of the empty set is infinity.
Depending on how $e(U)$ and $\mathrm{cost}(U)$ are specified, this defines ε-complexity in each of the four settings discussed above.

We stress that we take the infimum over *all* possible U for which the error does not exceed ε. In the worst case, average case and probabilistic settings, U can be identified with the pair (N, ϕ), where N is the information and ϕ is the algorithm that uses that information. This means that we take the infimum over *all* information N consisting of information operations from the class Λ, and over *all* algorithms ϕ that use N such that (N, ϕ) computes approximations with error at most ε. In the randomized setting, U can be identified with the 4-tuple (N, ϕ, T, ρ) and we take the infimum over *all* random information N_t and *all* random algorithms ϕ_t, where $t \in T$ is distributed accordingly to an arbitrary probability measure ρ. Sometimes we write

$$\mathrm{comp}^{\mathrm{wor}}(\varepsilon),\ \mathrm{comp}^{\mathrm{avg}}(\varepsilon),\ \mathrm{comp}^{\mathrm{prob}}(\varepsilon, \delta),\ \text{and}\ \mathrm{comp}^{\mathrm{ran}}(\varepsilon)$$

to emphasize the setting and the dependence on the parameter δ in the probabilistic setting. If we want to stress that we use one of the deterministic settings we then say, for example, the worst case deterministic setting or the average case deterministic setting and write $\mathrm{comp}^{\mathrm{wor-det}}(\varepsilon)$ or $\mathrm{comp}^{\mathrm{avg-det}}(\varepsilon)$.

Example (continued) For the integration problem, the model of computation assumes that one function evaluation costs $\mathbf{c}$, and each arithmetic operation, comparisons of real numbers, and evaluations of certain elementary functions can be performed exactly at unit cost. Usually $\mathbf{c} \gg 1$.

The worst case ε-complexity for the class F of Lipschitz functions with constant q is

$$\mathrm{comp}^{\mathrm{wor}}(\varepsilon) = \Theta\left(\mathbf{c}\,\frac{q}{\varepsilon}\right) \quad \text{as } \varepsilon \to 0.$$

For the average case and probabilistic settings, assume that μ is a truncated classical Wiener measure placed on the first derivatives. Then in the average case setting we have

$$\mathrm{comp}^{\mathrm{avg}}(\varepsilon) = \Theta\left(\mathbf{c}\,\left(\frac{1}{\varepsilon}\right)^{1/2}\right) \quad \text{as } \varepsilon \to 0.$$

In the probabilistic setting, for $q \gg \ln(1/\delta)$, we have

$$\mathrm{omp}^{\mathrm{prob}}(\varepsilon, \delta) = \Theta\left(\mathbf{c}\,\left(\frac{\sqrt{\ln(1/\delta)}}{\varepsilon}\right)^{1/2}\right) \quad \text{as } \varepsilon \to 0.$$

Finally, in the randomized setting we have

$$\mathrm{comp}^{\mathrm{ran}}(\varepsilon) = \Theta\left(\mathbf{c}\,\left(\frac{1}{\varepsilon}\right)^{2/3}\right) \quad \text{as } \varepsilon \to 0.$$

The complexity of integration in different settings has been studied for various classes of functions by many researchers, see [9, 18] for a list of references. $\square$

One of the main goals of IBC is to find or estimate the ε-complexity, and to find an ε-complexity optimal U, or equivalently, an ε-complexity optimal pair (N, ϕ). In the randomized setting, we want to find an ε-complexity optimal 4-tuple (N, ϕ, T, ρ). By *ε-complexity optimality* of U we mean that the error of U is at most ε and the cost of U is equal to, or not much greater than, the ε-complexity. For a number of problems this goal has been achieved due to the work of many researchers.

We briefly indicate a proof technique often used to obtain tight bounds on computational complexity of IBC problems. In what follows, we restrict ourselves to the worst case setting although a similar approach can be used in other settings.

As already explained, the approximation $U(f)$ is computed by combining information operations from the class Λ. Let $y = N(f)$ denote this computed information. In general, the operator N is many-to-one, and therefore the set $N^{-1}(y)$ consists of many elements from F which are indistinguishable from f. Then the set $S(N^{-1}(y))$ consists of all elements from G which are indistinguishable from $S(f)$. Since $U(f)$ is the same for any f from the set $N^{-1}(y)$, the element $U(f)$ must serve as an approximation to any element g from the set $SN^{-1}(y)$. It is clear that the quality of the approximation $U(f)$ depends on the "size" of the set $SN^{-1}(y)$.

The intuitive notion of size can be formalized by using the concept of radius. The radius of the set $A = SN^{-1}(y)$ is defined as the smallest radius of the ball which contains A,

$$\mathrm{rad}(A) = \inf_{g\in G} \sup_{a\in A} \|a - g\|.$$

The *radius of information* $r(N)$ is then defined as the maximal radius of the set $SN^{-1}(y)$ for $y \in N(F)$,

$$r(N) = \sup_{y\in N(F)} \mathrm{rad}(SN^{-1}(y)).$$

Clearly, the radius of information $r(N)$ is a sharp lower bound on the worst case error of any U. We can guarantee an ε-approximation iff $r(N)$ does not exceed ε (modulo a technical assumption that the corresponding infimum is attained).

The cost of computing $N(f)$ is at least $\mathbf{c}\, n$, where n, called the *cardinality* of N, denotes the number of information operations in N. By the *ε-cardinality number* $m(\varepsilon)$ we mean the minimal number n of information operations for which the information N has radius $r(N)$ at most ε,

$$m(\varepsilon) = \min\{n : \text{ there exists } N \text{ of cardinality at most } n \text{ such that } r(N) \le \varepsilon\}.$$

From this we obtain a lower bound on the ε-complexity,

$$\mathrm{comp}^{\mathrm{wor}}(\varepsilon) \;\ge\; \mathbf{c}\, m(\varepsilon).$$

It turns out that for many problems it is possible to find an information operator N_ε consisting of $m(\varepsilon)$ information operations, and a mapping ϕ_ε such that the approximation $U(f) = \phi_\varepsilon(N_\varepsilon(f))$ has error at most ε and $U(f)$ can be computed with cost at most $(\mathbf{c} + 2)\, m(\varepsilon)$. (For examples of such problems see [18], Chapter 5 and 7). This yields an upper bound on the ε-complexity,

$$\mathrm{comp}^{\mathrm{wor}}(\varepsilon) \;\le\; (\mathbf{c} + 2)\, m(\varepsilon).$$

Since usually $\mathbf{c} \gg 1$, the last two inequalities yield the almost exact value of the ε-complexity,

$$\mathrm{comp}^{\mathrm{wor}}(\varepsilon) \simeq \mathbf{c}\, m(\varepsilon).$$

This also shows that the pair $(N_\varepsilon, \phi_\varepsilon)$ is almost ε-complexity optimal.

In each setting of IBC one can define a radius of information such that we can guarantee an ε-approximation iff $r(N)$ does not exceed ε. This permits one to sometimes obtain tight complexity bounds in other settings.

The essence of this approach is that the radius of information as well as the ε-cardinality number $m(\varepsilon)$ and the information N_ε do not depend on particular algorithms, and they can often be expressed entirely in terms of well known mathematical concepts. Therefore we can sometimes obtain tight complexity bounds by drawing on powerful mathematical results.

3 Linear Multivariate Problems

In this section[3] we discuss complexity of linear multivariate problems. By a linear multivariate problem we mean an approximation of a linear operator defined on functions f of d variables. More precisely, let F_d be a class of functions $f : [0,1]^d \to \mathbb{R}$, and let

$$S_d : F_d \to G_d,$$

where G_d is a normed linear space.

We wish to approximate $S_d(f)$ for $f \in F_d$. Two primary examples are *multivariate integration*,

$$S_d(f) = \int_{[0,1]^d} f(t)\, dt \qquad \text{with } G_d = \mathbb{R},$$

and *multivariate approximation*,

$$S_d(f) = f \qquad \text{with } G_d = L_2([0,1]^d),$$

with F_d being a class of functions that are continuously r times differentiable.

As in Section 2, the cost of one function evaluation (or one evaluation of $L(f)$) is denoted by $\mathbf{c}$. To stress the dependence on the number d of variables, we write $\mathbf{c} = \mathbf{c}(d)$.

We are particularly interested in the complexity for large d and/or in large ε^{-1}. To stress the dependence on the error parameter ε and on the number of variables d, we denote the complexity by $\mathrm{comp}(\varepsilon, d)$.

[3]This section is based on Section 3 of [32].

Many multivariate problems are *intractable* and their complexity grows exponentially with the number d of variables. This is sometimes called the *curse of dimension*. Typically,

$$\mathrm{comp}(\varepsilon, d) \;=\; \Theta(\mathbf{c}(d)\varepsilon^{-d/r}), \quad \text{as } \varepsilon \to 0,$$

where r stands for the smoothness of the functions in the class F_d.

Problems which suffer the curse of dimension in the worst case setting include integration, approximation, global optimization, integral and partial differential equations for classes of functions whose rth derivatives are uniformly bounded in L_∞, see [1, 6, 8, 9, 13, 18, 27].

In the average case and randomized settings, the curse of dimension is present for approximation over the class of functions with r continuous derivatives which is equipped with the folded isotropic Wiener measure, see [15, 22] for the average case, and [7, 10, 21] for the randomized setting.

For some problems we can break the curse of dimension by switching to a different setting. For example, in the randomized setting, it is well known that the classical Monte Carlo algorithm breaks the curse of dimension for multivariate integration. In the average case setting, the curse of dimension is broken for multivariate integration no matter what probability measure is given on the class of functions. However, in general, the proof is not constructive. For the Wiener sheet measure, the proof is constructive and we know almost optimal algorithms, see [23, 28]. For multivariate approximation, the curse of dimension is broken only for some probability measures. For instance, it is broken for the Wiener sheet measure, see [23, 29], however, as already mentioned, it is not broken for the isotropic Wiener measure, see [15, 22].

It seems natural to characterize which multivariate problems are *tractable* or *strongly tractable* in various settings. More precisely, we say that the multivariate problem is *tractable* if there exist nonnegative numbers K, p and q such that

$$\mathrm{comp}(\varepsilon, d) \;\le\; K\,\mathbf{c}(d)\,d^q\,\varepsilon^{-p}, \quad \forall\, d, \forall\, \varepsilon \le 1. \tag{2}$$

If $q = 0$ then we say that the multivariate problem is *strongly tractable*. For strongly tractable problems, the only dependence of the complexity on d is through the cost $\mathbf{c}(d)$.

Tractability and strong tractability of linear multivariate problems have been studied in [30] for the information classes Λ^{std} and Λ^{all}. In the worst case and randomized settings we assume that the domain F_d and the range of S_d are Hilbert spaces. In the average case and probabilistic settings we assume that F_d is a Banach space equipped with a Gaussian measure μ_d and that the range of S_d is a Hilbert space.

For the class Λ^{all}, necessary and sufficient conditions for tractability and strong tractability can be obtained by using known IBC results on complexity of linear problems. They are expressed in terms of singular values of S_d or in terms of eigenvalues of the covariance operator of the measure $\mu_d S_d^{-1}$. Roughly speaking, tractability and strong tractability hold if the singular values tend to zero sufficiently fast.

Tractability and strong tractability in the randomized setting and the worst case setting are equivalent, and the corresponding complexities differ only by constants. This follows easily from [10]. Similarly, tractability and strong tractability in the probabilistic setting and the average case setting are equivalent due to relations between these two settings for linear problems, see [10].

We stress that for the class $\Lambda^{\rm all}$ the construction of an ε-approximation with minimal cost is easy since we know the optimal choice of linear functionals, and that linear algorithms are optimal.

We now turn to the class $\Lambda^{\rm std}$. Under mild assumptions, we prove in [30] that tractability and strong tractability in the classes $\Lambda^{\rm std}$ and $\Lambda^{\rm all}$ are equivalent. In particular, we prove that the exponents in ε^{-1} may differ by at most two. The proof of this equivalence is, however, *not* constructive.

One may suspect that only trivial problems are strongly tractable. However, even in the worst case setting, this is not true. More precisely, if F_d is a unit ball of a reproducing kernel Hilbert space and the linear problem is suitably normalized, then there exists a constant K such that

$$\mathrm{comp}(\varepsilon, d) \;\le\; K\,\mathbf{c}(d)\,\varepsilon^{-p},$$

where $p = 2$ for the class $\Lambda^{\rm all}$, and $p = 4$ for the class $\Lambda^{\rm std}$, see [30]. It is also known that $p = 2$ for the class $\Lambda^{\rm all}$ is sharp, whereas it is open whether $p = 4$ for the class $\Lambda^{\rm std}$ can be improved.

As before, the proof for the class $\Lambda^{\rm std}$ is not constructive. A construction is known for linear multivariate problems that are defined by tensor products, [23, 31]. For tractable tensor product problems and for the class $\Lambda^{\rm std}$, we construct polynomial-time algorithms, see [23]. This construction is based on Smolyak's algorithm, see [16]. More precisely, in the worst case and average case settings, we present linear algorithms that compute an ε-approximation for the multivariate tensor product problem with cost

$$\mathrm{cost}(\varepsilon, d) \;\le\; (\mathbf{c}(d) + 2)\;\beta_1 \left(\beta_2 \,+\, \beta_3\,\frac{\ln\,1/\varepsilon}{d-1}\right)^{\beta_4(d-1)} \left(\frac{1}{\varepsilon}\right)^{\beta_5}.$$

The coefficients β_i's do not depend on d; they are determined by the properties of the problem for $d = 1$.

Note the intriguing dependence of the cost bound on d. The leading term $\varepsilon^{-\beta_5}$ does not depend on d, whereas $\ln\,1/\varepsilon$ is divided by a multiple of $d-1$ and then raised to a multiple of $d-1$. If the tensor product problem is tractable then the cost bound does not exceed $\mathbf{c}(d)\, K\, \varepsilon^{-p}$ for some numbers K and p, both independent of d.

We illustrate the results for multivariate approximation and integration for the class $\Lambda^{\rm std}$ in the average case setting for the class of continuous functions $f:[0,1]^d \to \mathbb{R}$ equipped with the Wiener sheet measure. For the approximation problem, we know a linear algorithm, see [23], that computes an ε-approximation with cost

$$\mathrm{cost}(\varepsilon,d) \;\le\; \mathbf{c}(d)\, 0.8489\, \left(2.9974 \;+\; 4.3869\, \frac{-0.9189 \;+\; \ln 1/\varepsilon}{d-1}\right)^{2(d-1)} \left(\frac{1}{\varepsilon}\right)^2 .$$

This algorithm has optimal powers of ε^{-1} and $\ln 1/\varepsilon$ since

$$\mathrm{comp}(\varepsilon,d) \;=\; \Theta\left(\varepsilon^{-2}\, (\ln 1/\varepsilon)^{2(d-1)}\right),$$

see [29]. This approximation problem is strongly tractable since

$$\mathrm{cost}(\varepsilon,d) \;\le\; \mathbf{c}(d)\, 2.37632\, \varepsilon^{-5.672}.$$

The exponent 5.672 seems to be too high; however, no smaller exponent has been found so far.

We would like again to add that the choice of the Wiener sheet measure is essential. It is known, see [22], that if we replace the Wiener sheet measure by the isotropic Wiener measure then the approximation problem is intractable since $\mathrm{comp}(\varepsilon,d) = \Theta\left(\mathbf{c}(d)\, \varepsilon^{-2d}\right)$.

Consider now the integration problem $S_d f = \int_{[0,1]^d} f(x)dx$ in the average case setting for the class of continuous functions $f:[0,1]^d \to \mathbb{R}$ equipped with the Wiener sheet measure. Then we know a linear algorithm, see [23], which computes an ε-approximation with $\mathrm{cost}(\varepsilon,d)$ bounded by

$$\mathrm{cost}(\varepsilon,d) \;\le\; \mathbf{c}(d)\, 3.304\, \left(1.77959 \;+\; 2.714\, \frac{-1.12167 \;+\; \ln 1/\varepsilon}{d-1}\right)^{1.5(d-1)} \frac{1}{\varepsilon}.$$

The power of ε^{-1} is optimal and the power of $\ln 1/\varepsilon$ is too large since

$$\mathrm{comp}(\varepsilon,d) \;=\; \Theta\left(\varepsilon^{-1}\, (\ln 1/\varepsilon)^{(d-1)/2}\right),$$

see [28]. This integration problem is strongly tractable since

$$\mathrm{cost}(\varepsilon,d) \;\le\; \mathbf{c}(d)\, 7.26\, \varepsilon^{-2.454}.$$

The exponent 2.454 is too high. There exists an algorithm with an exponent at most 1.4788..., see [26]. The proof of this latter fact is, however, *not* constructive.

This integration problem is related to discrepancy in the L_2-norm, see [28]. Using this relation we obtain an upper bound, which is independent of d, for the number $n(\varepsilon, d)$ of points for which discrepancy (with unequal weights) is at most ε,

$$n(\varepsilon, d) \leq 7.26\, \varepsilon^{-2.454}, \quad \forall\, d, \forall\, \varepsilon \leq 1.$$

4 Path Integration

Path integrals occur in many applied fields including quantum physics and chemistry, differential equations, and financial mathematics, as well as average case complexity. The path integration problem is defined as the approximation of

$$S(f) = \int_X f(x)\, \mu(dx), \quad \forall\, f \in F.$$

Here, X is a separable infinite dimensional Banach space and μ is a zero mean Gaussian measure on X. The class F is a class of (Borel) measurable real functions defined on X.

A typical approach is to approximate the path integral by high dimensional integrals and apply a Monte Carlo (randomized) algorithm. Do we really need to use randomized algorithms for path integrals? Perhaps we can find an effective *deterministic* algorithm that approximates path integrals with small error. To answer this question, we study the *worst case complexity* of path integration in the class Λ^{lstd}. Path integration is considered with respect to different Gaussian measures μ and different classes F of integrands.

Tractability of path integration means that the complexity depends polynomially on ε^{-1}. For the class F of integrands that are r times Frechet differentiable, tractability of path integration holds iff the covariance operator of the Gaussian measure μ has finite rank. Hence, if the Gaussian measure μ is supported on an infinite dimensional space then path integration is intractable. In this case, there exists no effective deterministic algorithm, and the use of randomized algorithms is reasonable. In fact, for this class of integrands, the classical Monte Carlo algorithm is optimal and the complexity in the randomized setting is proportional to ε^{-2}, see [24].

On the other hand, for a particular class F of entire integrands, the worst case complexity of path integration is at most of order ε^{-p} with p depending on the Gaussian measure μ. Hence, path integration is now tractable. Furthermore, for any Gaussian measure μ, the exponent p is less than or equal to 2. For the Wiener measure we have $p = 2/3$. For this class of entire integrands, we provide effective deterministic algorithms that solve the path integration problem with (worst case) cost that is usually much less than the (randomized) cost of the classical Monte Carlo algorithm, see [24].

In [25] we consider a class of functions related to the Feynman-Kac formula. More precisely, this is the class of potential and initial conditions functions that define the heat equation. Although these functions do not need to be very smooth, we prove tractability of path integration, and in many cases, the worst case complexity is substantially smaller than ε^{-2}.

Acknowledgment

We thank L. Plaskota and A. G. Werschulz for useful comments on this paper.

References

[1] N. S. Bakhvalov, On approximate calculation of integrals (in Russian), *Vestnik MGU, Ser. Mat. Mekh. Astron. Fiz. Khim.*, 4, 3-18, 1959.

[2] L. Blum, M. Shub and S. Smale, On a Theory of Computation and Complexity over the Real Numbers: NP-Completeness, Recursive Functions and Universal Machines *Bull. Amer. Math. Soc.*, 21, 1-46, 1989.

[3] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progression, in *Proc. of the Nineteenth ACM Symp. on Theor. of Comp.*, 1-6, New York, 1987.

[4] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, *W. H. Freeman and Company*, New York, 1979.

[5] S. Heinrich, Random Approximation in Numerical Analysis, in Proc. of the Functional Analysis Conference, Essen 1991, *Lecture Notes in Pure and Applied Mathematics*, ed. K. D. Bierstedt et al. , vol. 150, Marcel Dekker, New York, 123-171, 1993.

[6] S. Heinrich, Complexity of integral equations and relations to s-numbers, *J. Complexity*, 9, 141-153, 1993.

[7] P. Mathé, Random approximation of Sobolev embedding, *J. Complexity*, 7, 261-281, 1993.

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

[9] E. Novak, Deterministic and Stochastic Error Bounds in Numerical Analysis, *Lectures Notes in Math.*, Springer Verlag, Berlin, vol. 1349, 1988.

[10] E. Novak, Optimal linear randomization methods for linear operators in Hilbert spaces, *J. Complexity*, 8, 22-36, 1992.

[11] E. Novak, Algorithms and complexity for continuous problems, in Geometry, Analysis, and Mechanics, ed. J. M. Rassias, *World Scientific*, Singapore, 96-128, 1994.

[12] E. Novak, The real number model in numerical analysis, *J. Complexity*, 11, 57-73, 1995.

[13] S. V. Pereverzev, On the complexity of the problem of finding solutions of Fredholm equations of the second kind with differentiable kernels (in Russian), *Ukrain. Mat. Sh.*, 41, 1422-1425, 1989.

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

[15] K. Ritter and G. W. Wasilkowski, Integration and L_2-approximation: average case setting with isotropic Wiener measure for smooth functions, *Rocky Mount. J. Math.*, to appear, 1995.

[16] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, *Dokl. Akad. Nauk SSSR*, p. 240-243, 1963.

[17] V. Strassen, Gaussian elimination is not optimal, *Numer. Math.*, 13, 354-356, 1969.

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

[19] J. F. Traub and H. Woźniakowski, Theory and Applications of Information-Based Complexity, *1990 Lectures in Complex Systems, Santa Fe Institute*, Lect. Vol. III, Eds. L. Nadel and D. Stein, Addison-Wesley, 163-193, 1991.

[20] J. F. Traub and H. Woźniakowski, Recent Progress in Information-Based Complexity, *Bulletin of EATCS*, 51, 141-154, 1993.

[21] G. W. Wasilkowski, Randomization for Continuous Problems, *J. Complexity*, 5, 195-218, 1989.

[22] G. W. Wasilkowski, Integration and approximation of multivariate functions: average case complexity with isotropic Wiener measure, *Bull. Amer. Math. Soc. (N.S)*, 28, 308-314, 1993.

[23] G. W. Wasilkowski, and H. Woźniakowski, Explicit cost bounds of algorithms for multivariate tensor product problems, *J. Complexity*, 11, p. 1-56, 1995.

[24] G. W. Wasilkowski, and H. Woźniakowski, On tractability of path integration, to appear, 1995.

[25] G. W. Wasilkowski, and H. Woźniakowski, Worst case complexity of Feynman-Kac path integration, to appear, 1995.

[26] G. W. Wasilkowski, and H. Woźniakowski, The exponent of discrepancy is at most 1.4778..., to appear, 1995.

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

[28] H. Woźniakowski, Average Case Complexity of Multivariate Integration, *Bull. Amer. Math. Soc. (N.S)*, 24, p. 185-194, 1991.

[29] H. Woźniakowski, Average case complexity of linear multivariate problems: Part I. Theory, Part II. Applications, *J. Complexity*, 8, 337-372, 373-392, 1992.

[30] H. Woźniakowski, Tractability and strong tractability of linear multivariate problems, *J. Complexity*, 10, 96-128, 1994.

[31] H. Woźniakowski, Tractability and strong tractability of multivariate tensor product problems, *J. Computing and Information*, 4, 1-19, 1994.

[32] H. Woźniakowski, Overview of information-based complexity, in *Lectures in Applied Mathematics*, eds. J. Renegar, M. Shub and S. Smale, to apper, 1995.

Author's Address:

Henryk Woźniakowski, Department of Computer Science, Columbia University,
New York, NY 10027, USA, and Institute of Applied Mathematics, University of Warsaw,
ul. Banacha 2, 02-097 Warszawa, Poland, email: henryk@cs.columbia.edu