On the power of restricted Monte Carlo algorithms

Stefan Heinrich
Department of Computer Science
University of Kaiserslautern
D-67653 Kaiserslautern, Germany

March 10, 2019

Abstract

We introduce a general notion of restricted Monte Carlo algorithms that generalizes previous notions in two ways: it includes full adaptivity and general (i.e. not only bit) restrictions. We show that for each such restricted setting there is a computational problem that can be solved in the general randomized setting but not under the restriction.

1 Introduction

Restricted Monte Carlo algorithms were considered in [11, 12, 16, 9, 13, 2, 17, 3, 4] (in the present paper the terms 'randomized' and 'Monte Carlo' will be used synonymously). Restriction usually means that the algorithm has access only to random bits or to random variables with finite range. In a number of numerical problems the admission of randomized algorithms brings considerable gains in terms of the convergence rate, e.g., in high dimensional integration, see [3, 5, 6, 12, 15] for this and other problems. So it is certainly of theoretical interest to understand how much randomness is really needed. Looking at algorithms that use random bits is an obvious way to quantify randomness.

Most of the papers on restricted randomized algorithms consider the non-adaptive case. Only [4] includes adaptivity, but considers a class of algorithms where each information call is tied to one random bit call. Since for a number of important problems essentially fewer random elements are needed than function values [9, 13, 2, 17], a general notion of adaptive restricted randomized algorithms is of interest.

Here we give such a general definition, which extends the previous notions in two ways: Firstly, we include full adaptivity, that is, the question whether to call a function value or a random element is decided solely on the basis of the outcome of the computation carried out so far. Secondly, we do not restrict the consideration to random bits, but include models in which the algorithms have access to an arbitrary, but fixed set of random variables, for example, uniform distributions on $[0, 1]$. This general definition was inspired by the approach to stochastic problems from [7, 8].

A first (simple, but technically somewhat involved) step to fit this notion into the existing IBC framework is to represent each restricted randomized algorithm as a general randomized algorithm of suitable cardinality. Secondly, and this is the main topic of the present paper, we study the question of the power of restricted as compared to general randomized algorithms. It became clear from the previous work that there are problems which can be solved by general

Monte Carlo but not by Monte Carlo algorithms that use only random bits or random variables with finite range (see [16], Th. 3.1, or example 14 in [4], the latter credited to E. Novak).

The question arises whether there are such problems for randomized algorithms with access to uniform distributions on $[0,1]$, or with access to an even more general, but still restricted set of random variables. We settle this question by showing that for each restriction model there is a problem which cannot be solved in this restricted model, but can be solved (in one step) by a general randomized algorithm. Our examples are based on cardinal number considerations and on probability measures on suitably large sets. This can be viewed as a generalization of the above-mentioned examples.

Let us finally mention that recently a very practical aspect came up, which motivates the consideration of restricted Monte Carlo algorithms. The usual sequences of 'random numbers' on a computer are generated in a deterministic way, using number theoretic methods, thus they are not random at all, just have appropriate statistical properties. With the appearance of quantum random generators the use of truly random bits became realistic, see, e.g., [14] and references therein. Of course, now the minimal number of random bits is also of practical interest.

2 Restricted randomized algorithms in a general setting

We work in the framework of information-based complexity theory (IBC) [12, 15]. First we recall the notions of deterministic and randomized algorithms in the IBC approach from [5, 6], see also [7, 8]. An abstract numerical problem $\mathscr{P}$ is given as

$$\mathscr{P} = (F, G, S, K, \Lambda). \tag{1}$$

Here F is a non-empty set, G a Banach space and S is a mapping $F \to G$. The operator S is called the solution operator, it sends the input $f \in F$ of our problem to the exact solution $S(f)$. Moreover, Λ is a nonempty set of mappings from F to K, the set of information functionals, where K is any nonempty set - the set of values of information functionals.

For illustration, let us consider an example (which will be used later on). Let $(Q, \mathscr{F}, \mu)$ be a probability space and let $\mathscr{L}_2(Q, \mathscr{F}, \mu)$ denote the set of all $\mathscr{F}$-to-Borel measurable, μ-square integrable functions $f : Q \to \mathbb{R}$. We set

$$F = B_{\mathscr{L}_2(Q,\mathscr{F},\mu)} = \{f \in \mathscr{L}_2(Q, \mathscr{F}, \mu) : \|f\|_{\mathscr{L}_2(Q,\mathscr{F},\mu)} \le 1\}, \quad G = \mathbb{R}, \tag{2}$$

$$S = I_{Q,\mu} : F \to \mathbb{R}, \quad I_{Q,\mu}(f) = \int_Q f(x)d\mu(x) \quad (f \in F), \tag{3}$$

$$K = \mathbb{R}, \quad \Lambda = \{\delta_x : x \in Q\}, \quad \delta_x(f) := f(x) \quad (f \in F). \tag{4}$$

Thus, we want to compute (approximately) the integral of functions from the unit ball of $\mathscr{L}_2(Q, \mathscr{F}, \mu)$. The set of available information functionals consists of function evaluations at arbitrary points of Q.

The basic IBC approach to a general notion of an algorithm is the following. The algorithm starts with evaluating an information functional $L_1 \in \Lambda$ at input $f \in F$, that is $L_1(f) \in K$. Next a termination function $\tau_1(L_1(f))$ is evaluated. If its value is 1, we stop the process of gathering information. If the value is 0, we go on and choose, depending on $L_1(f)$, another functional $L_2 \in \Lambda$, and $L_2(f)$ is evaluated. The termination function $\tau_2(L_1(f), L_2(f))$ decides if to stop or to continue. In the latter case, the choice of the next functional $L_3 \in \Lambda$ may depend on $L_1(f)$ and $L_2(f)$, and so on. The procedure goes on until $\tau_n(L_1(f),\dots,L_n(f))=1$ for some n, thus n values $L_j(f)$ $(j=1,\dots,n)$ are obtained, the 'information' about f. On the basis of this information a final mapping $\varphi_n : K^n \to G$ is applied, representing the computations on the information leading to the approximation $A(f)$ to $S(f)$ in G.

This is formalized as follows (including also the case of choosing no information functionals at all, which we omitted in the above informal description). An (adaptive) deterministic algorithm for $\mathscr{P}$ is a tuple $A=((L_i)_{i=1}^\infty,(\tau_i)_{i=0}^\infty,(\varphi_i)_{i=0}^\infty)$ such that $L_1\in\Lambda$, $\tau_0\in\{0,1\}$, $\varphi_0\in G$, and for $i\in\mathbb{N}$

$$L_{i+1}: K^i \to \Lambda, \quad \tau_i : K^i \to \{0,1\}, \quad \varphi_i : K^i \to G \tag{5}$$

are arbitrary mappings, where K^i denotes the i-th Cartesian power of K. Given an input $f\in F$, we define $(\lambda_i)_{i=1}^\infty$ with $\lambda_i\in\Lambda$ as follows:

$$\lambda_1 = L_1, \quad \lambda_i = L_i(\lambda_1(f),\dots,\lambda_{i-1}(f)) \quad (i\ge 2). \tag{6}$$

Define $\operatorname{card}(A,f)$, the cardinality of A at input f, to be 0 if $\tau_0=1$. If $\tau_0=0$, let $\operatorname{card}(A,f)$ be the first integer $n\ge 1$ with $\tau_n(\lambda_1(f),\dots,\lambda_n(f))=1$ if there is such an n. If $\tau_0=0$ and no such $n\in\mathbb{N}$ exists, put $\operatorname{card}(A,f)=+\infty$. We define the output $A(f)$ of algorithm A at input f as

$$A(f)=\begin{cases}\varphi_0 & \text{if} \quad \operatorname{card}(A,f)\in\{0,\infty\}\\ \varphi_n(\lambda_1(f),\dots,\lambda_n(f)) & \text{if} \quad 1\le \operatorname{card}(A,f)=n<\infty.\end{cases} \tag{7}$$

The cardinality of A is defined as

$$\operatorname{card}(A,F)=\sup_{f\in F}\operatorname{card}(A,f).$$

Given $n\in\mathbb{N}_0$, we define $\mathscr{A}_n^{\rm det}(\mathscr{P})$ as the set of deterministic algorithms A for $\mathscr{P}$ with $\operatorname{card}(A)\le n$, the error of A in approximating S as

$$e(S,A,F,G)=\sup_{f\in F}\|S(f)-A(f)\|_G,$$

and for $n\in\mathbb{N}_0$ the deterministic n-th minimal error of S as

$$e_n^{\rm det}(S,F,G)=\inf_{A\in\mathscr{A}_n^{\rm det}(\mathscr{P})} e(S,A,F,G). \tag{8}$$

In the case of the example (2)–(4) above a deterministic algorithm calls function values $\lambda_1(f)=f(x_1)$, $\dots$, $\lambda_n(f)=f(x_n)$, where the sample points $x_i\in Q$ can be chosen adaptively, depending on the so far computed information, and also the termination number n may be adaptive in this sense. The cardinality $\operatorname{card}(A,f)$ is the total number of function values n called at input f. Finally, the mapping φ_n is applied to produce the output of the algorithm, the approximation $\varphi_n(f(x_1),\dots,f(x_n))$ to the integral $I_{Q,\mu}(f)$.

An (unrestricted) randomized algorithm for $\mathscr{P}$ is a tuple $A=((\Omega,\Sigma,\mathbb{P}),\ (A_\omega)_{\omega\in\Omega})$, where $(\Omega,\Sigma,\mathbb{P})$ is a probability space and for each $\omega\in\Omega$, A_ω is a deterministic algorithm for $\mathscr{P}$. Let $n\in\mathbb{N}_0$. Then $\mathscr{A}_n^{\rm ran}(\mathscr{P})$ stands for the class of randomized algorithms A for $\mathscr{P}$ with the following properties: For each $f\in F$ the mapping $(\omega)\to\operatorname{card}(A_\omega,f)$ is Σ-measurable,

$$\mathbb{E}\operatorname{card}(A_\omega,f)\le n,$$

and the mapping $\omega \to A_\omega(f)$ is Σ-to-Borel measurable and $\mathbb{P}$-almost surely separably valued, i.e., there is a separable subspace G_f of G such that $\mathbb{P}\{\omega : A_\omega(f) \in G_f\} = 1$. We define the cardinality of $A \in \mathscr{A}_n^{\mathrm{ran}}(\mathscr{P})$ as

$$\operatorname{card}(A, F) = \sup_{f \in F} \mathbb{E} \operatorname{card}(A_\omega, f),$$

the error as

$$e(S, A, F, G) = \sup_{f \in F} \mathbb{E} \, \|S(f) - A_\omega(f)\|_G,$$

and the randomized n-th minimal error of S as

$$e_n^{\mathrm{ran}}(S, F, G) = \inf_{A \in \mathscr{A}_n^{\mathrm{ran}}(\mathscr{P})} e(S, A, F, G).$$

Considering trivial one-point probability spaces $\Omega = \{\omega\}$ immediately yields

$$e_n^{\mathrm{ran}}(S, F, G) \le e_n^{\mathrm{det}}(S, F, G). \tag{9}$$

A classical example of an unrestricted randomized algorithm is the standard Monte Carlo method for integration (2)–(4) with $n \in \mathbb{N}$ samples. Here we take a sufficiently large probability space, e.g.,

$$(\Omega, \Sigma, \mathbb{P}) = (Q, \mathscr{F}, \mu)^n,$$

a sequence $(\xi_i)_{i=1}^n$ of independent, uniformly distributed on Q random variables over $(\Omega, \Sigma, \mathbb{P})$, and put $A_n = ((\Omega, \Sigma, \mathbb{P}), (A_{n,\omega})_{\omega \in \Omega})$, where

$$A_{n,\omega}(f) = \frac{1}{n} \sum_{i=1}^n f(\xi_i(\omega)). \tag{10}$$

To view this algorithm in the formal context of (5), fix $n \in \mathbb{N}$ and $\omega \in \Omega$. Then we have $A_{n,\omega} = ((L_i)_{i=1}^\infty, (\tau_i)_{i=0}^\infty, (\varphi_i)_{i=0}^\infty)$ with

$$\begin{aligned} L_i &\equiv \delta_{\xi_i(\omega)} \quad (i = 1, \dots, n) && (11)\\ \tau_i &\equiv 0 \quad (0 \le i < n), \quad \tau_n \equiv 1 && (12)\\ \varphi_n(a_1, \dots, a_n) &= \frac{1}{n} \sum_{i=1}^n a_i \quad (a_i \in \mathbb{R}), && (13) \end{aligned}$$

while all other algorithm components can be chosen arbitrarily – they do not contribute to the output $A_{n,\omega}(f)$. It is well-known (see, e.g., [12], 2.1.3) that

$$e(I_{Q,\mu}, A_n, B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}, \mathbb{R}) = \sup_{f \in B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}} \mathbb{E}|I_{Q,\mu}(f) - A_{n,\omega}(f)| \le n^{-1/2}. \tag{14}$$

Now we introduce the new notion of a restricted randomized algorithm for $\mathscr{P}$. A probability space with access restriction is a tuple

$$\mathscr{R} = \big((\Omega, \Sigma, \mathbb{P}), K', \Lambda'\big), \tag{15}$$

where $(\Omega, \Sigma, \mathbb{P})$ is a probability space, K' a non-empty set, and Λ' a non-empty set of mappings from Ω to K'. With $\mathscr{P} = (F, G, S, K, \Lambda)$ as above we set

$$\bar{K} = K \dot{\cup} K', \quad \bar{\Lambda} = \Lambda \dot{\cup} \Lambda',$$

where $\dot{\cup}$ denotes the disjoint union. For $\lambda \in \bar{\Lambda}$ we define

$$\lambda(f,\omega) = \begin{cases} \lambda(f) & \text{if} \quad \lambda \in \Lambda \\ \lambda(\omega) & \text{if} \quad \lambda \in \Lambda'. \end{cases}$$

An $\mathscr{R}$-restricted randomized algorithm for problem $\mathscr{P}$ is a tuple $A = ((L_i)_{i=1}^\infty, (\tau_i)_{i=0}^\infty, (\varphi_i)_{i=0}^\infty)$ such that $L_1 \in \bar{\Lambda}$, $\tau_0 \in \{0,1\}$, $\varphi_0 \in G$, and for $i \in \mathbb{N}$

$$L_{i+1} : \bar{K}^i \to \bar{\Lambda}, \quad \tau_i : \bar{K}^i \to \{0,1\}, \quad \varphi_i : \bar{K}^i \to G \tag{16}$$

are any mappings. Given $f \in F$ and $\omega \in \Omega$, we define $(\lambda_i)_{i=1}^\infty$ with $\lambda_i \in \bar{\Lambda}$ as follows:

$$\lambda_1 = L_1, \quad \lambda_i = L_i(\lambda_1(f,\omega), \dots, \lambda_{i-1}(f,\omega)) \quad (i \geq 2). \tag{17}$$

Define $\operatorname{card}_{\bar{\Lambda}}(A,f,\omega)$, $\operatorname{card}_{\Lambda}(A,f,\omega)$, and $\operatorname{card}_{\Lambda'}(A,f,\omega)$ all to be 0 if $\tau_0 = 1$. If $\tau_0 = 0$, let $\operatorname{card}_{\bar{\Lambda}}(A,f,\omega)$ be the first integer $n \geq 1$ with $\tau_n(\lambda_1(f,\omega), \dots, \lambda_n(f,\omega)) = 1$ if there is such an n. If $\tau_0 = 0$ and no such $n \in \mathbb{N}$ exists, put $\operatorname{card}_{\bar{\Lambda}}(A,f,\omega) = +\infty$. Let

$$\begin{aligned} \operatorname{card}_{\Lambda}(A,f,\omega) &= |\{k \leq \operatorname{card}_{\bar{\Lambda}}(A,f,\omega) : \lambda_k \in \Lambda\}| \\ \operatorname{card}_{\Lambda'}(A,f,\omega) &= |\{k \leq \operatorname{card}_{\bar{\Lambda}}(A,f,\omega) : \lambda_k \in \Lambda'\}|. \end{aligned}$$

Clearly, $\operatorname{card}_{\bar{\Lambda}}(A,f,\omega) = \operatorname{card}_{\Lambda}(A,f,\omega) + \operatorname{card}_{\Lambda'}(A,f,\omega)$. We define the output $A(f,\omega)$ of algorithm A at input (f,ω) as

$$A(f,\omega) = \begin{cases} \varphi_0 & \text{if} \quad \operatorname{card}_{\bar{\Lambda}}(A,f,\omega) \in \{0,\infty\} \\ \varphi_n(\lambda_1(f,\omega), \dots, \lambda_n(f,\omega)) & \text{if} \quad 1 \leq \operatorname{card}_{\bar{\Lambda}}(A,f,\omega) = n < \infty. \end{cases} \tag{18}$$

Thus, a restricted randomized algorithm depends on randomness of $(\Omega, \Sigma, \mathbb{P})$, but in a special way. Namely, $\omega \in \Omega$ can only be accessed through the functionals $\lambda(\omega)$ for $\lambda \in \Lambda'$. Intuitively, it seems to be clear that a restricted randomized algorithm is a special case of a randomized algorithm. Formally, though, this has to be checked on the basis of the respective definitions. Corollary 2.2 states that this is indeed the case.

Also note the similarities of the definition of a restricted randomized algorithm with the notion of a deterministic algorithm for a stochastic problem from [7, 8].

Given $n, k \in \mathbb{N}_0$, we define $\mathscr{A}_{n,k}^{\text{ran}}(\mathscr{P}, \mathscr{R})$ as the set of those $\mathscr{R}$-restricted randomized algorithms for problem $\mathscr{P}$ with the following properties: For each $f \in F$ the mappings

$$\omega \to \operatorname{card}_{\bar{\Lambda}}(A,f,\omega), \quad \omega \to \operatorname{card}_{\Lambda}(A,f,\omega), \quad \omega \to \operatorname{card}_{\Lambda'}(A,f,\omega)$$

are Σ-measurable,

$$\mathbb{E} \operatorname{card}_{\Lambda}(A,f,\omega) \leq n, \quad \mathbb{E} \operatorname{card}_{\Lambda'}(A,f,\omega) \leq k,$$

and the mapping

$$\omega \to A(f,\omega) \in G$$

is Σ-to-Borel measurable and $\mathbb{P}$-almost surely separably valued. The error of $A \in \mathscr{A}_{n,k}^{\text{ran}}(\mathscr{P}, \mathscr{R})$ is defined as

$$e(S,A,F,G) = \sup_{f \in F} \mathbb{E} \, \|S(f) - A(f,\omega)\|_G.$$

The (n,k)-th minimal $\mathscr{R}$-restricted randomized error of S is defined as

$$e_{n,k}^{\text{ran}}(S,F,G) = \inf_{A \in \mathscr{A}_{n,k}^{\text{ran}}(\mathscr{P},\mathscr{R})} e(S,A,F,G). \tag{19}$$

For example, bit Monte Carlo algorithms fit the above definition with $K' = \{0,1\}$, $\Lambda' = \{\xi_i : 1 \le i < \infty\}$, with (ξ_i) being independent random variables on $(\Omega, \Sigma, \mathbb{P})$ with $\mathbb{P}(\{\xi_i = 0\}) = \mathbb{P}(\{\xi_i = 1\}) = 1/2$. The restricted Monte Carlo algorithms considered by Novak in [11, 12] correspond to arbitrary K' and Λ' consisting of random variables on $(\Omega, \Sigma, \mathbb{P})$ with finite range and rational distribution probabilities. Of particular interest, because most frequently used, is the case where $K' = [0,1]$ and $\Lambda' = \{\eta_i : 1 \le i < \infty\}$, with (η_i) being independent uniformly distributed on $[0,1]$ random variables over $(\Omega, \Sigma, \mathbb{P})$.

Concerning example (2)–(4), one might ask if the same rate as in (14) could be obtained by the help of a finite number of uniformly distributed on $[0,1]$ random variables (and maybe suitable transformations). If Q is too large, this may not be the case. In fact, it can happen that no rate whatsoever is possible. This statement is a special case of Theorem 3.1 below.

To a given $\mathscr{R}$-restricted randomized algorithm A for $\mathscr{P}$ and $\omega \in \Omega$ we can associate a deterministic algorithm A_ω for $\mathscr{P}$. The following proposition is related to Lemma 3 in [7], with a refined statement about the cardinality of the resulting algorithm A_ω.

Proposition 2.1. *Let A be an $\mathscr{R}$-restricted randomized algorithm for $\mathscr{P}$. Then for each $\omega \in \Omega$ there is a deterministic algorithm A_ω for $\mathscr{P}$ such that for all $f \in F$*

$$\begin{aligned} \operatorname{card}(A_\omega, f) &= \operatorname{card}_\Lambda(A, f, \omega) && (20)\\ A_\omega(f) &= A(f, \omega). && (21)\end{aligned}$$

Proof. Let $\nu_0 \in \Lambda$ be any element, let $A = ((L_i)_{i=1}^\infty, (\tau_i)_{i=0}^\infty, (\varphi_i)_{i=0}^\infty)$, and fix $\omega \in \Omega$. Our goal is to define a suitable algorithm $A_\omega = ((L_{i,\omega})_{i=1}^\infty, (\tau_{i,\omega})_{i=0}^\infty, (\varphi_{i,\omega})_{i=0}^\infty)$.

We start with the following construction. Given an arbitrary sequence $(y_l)_{l=1}^\infty \in K^{\mathbb{N}}$, we define two sequences $(\lambda_i)_{i=1}^\infty \in \Lambda^{\mathbb{N}}$ and $(z_i)_{i=1}^\infty \in K^{\mathbb{N}}$ inductively as follows. Let

$$\begin{aligned} \lambda_1 &= L_1 && (22)\\ z_1 &= \begin{cases} y_1 & \text{if} \quad \lambda_1 \in \Lambda \\ \lambda_1(\omega) & \text{if} \quad \lambda_1 \in \Lambda'. \end{cases} && (23)\end{aligned}$$

Now let $i \ge 1$, assume that $(\lambda_j)_{j \le i}$ and $(z_j)_{j \le i}$ have been defined, let

$$l = |\{j \le i : \lambda_j \in \Lambda\}|, \qquad (24)$$

and set

$$\begin{aligned} \lambda_{i+1} &= L_{i+1}(z_1, \dots, z_i) && (25)\\ z_{i+1} &= \begin{cases} y_{l+1} & \text{if} \quad \lambda_{i+1} \in \Lambda \\ \lambda_{i+1}(\omega) & \text{if} \quad \lambda_{i+1} \in \Lambda'. \end{cases} && (26)\end{aligned}$$

Observe that, roughly speaking, $(\lambda_i)_{i=1}^\infty$ is something like the sequence (17), just with 'input' $(y_l)_{l=1}^\infty$ instead of f. It is convenient for us to set $\lambda_\infty = \nu_0$. Let $k_0 = 0$ and define for $l \in \mathbb{N}$

$$k_l = \min\{i \in \mathbb{N} : i > k_{l-1}, \lambda_i \in \Lambda\}, \qquad (27)$$

with the understanding that $\min \emptyset = \infty$. This defines the function

$$\Psi : K^{\mathbb{N}} \to \Lambda^{\mathbb{N}} \times K^{\mathbb{N}} \times (\mathbb{N}_0 \cup \{\infty\})^{\mathbb{N}_0}, \quad \Psi\big((y_l)_{l=1}^\infty\big) = \big((\lambda_i)_{i=1}^\infty, (z_i)_{i=1}^\infty, (k_l)_{l=0}^\infty\big).$$

We note that for each $l \in \mathbb{N}_0$ the following holds. Let $(\tilde{y}_j)_{j=1}^\infty \in K^{\mathbb{N}}$ be such that $(y_j)_{j\le l} = (\tilde{y})_{j\le l}$ and let

$$\Psi\big((\tilde{y}_l)_{l=1}^\infty\big) = \big((\tilde{\lambda}_i)_{i=1}^\infty, (\tilde{z}_i)_{i=1}^\infty, (\tilde{k}_l)_{l=0}^\infty\big).$$

Then

$$(\lambda_j)_{j\le k_{l+1}} = (\tilde{\lambda}_j)_{j\le k_{l+1}}, \quad (z_j)_{j<k_{l+1}} = (\tilde{z}_j)_{j<k_{l+1}}, \quad (k_p)_{p\le l+1} = (\tilde{k}_p)_{p\le l+1}. \tag{28}$$

Next we define $((L_{l,\omega})_{l=1}^\infty, (\tau_{l,\omega})_{l=0}^\infty, (\varphi_{l,\omega})_{l=0}^\infty)$ for finite strings $(y_1,\dots,y_l)$ of the given sequence $(y_l)_{l=1}^\infty$. Let $l \in \mathbb{N}_0$ and set

$$L_{l+1,\omega}(y_1,\dots,y_l) = \begin{cases} \lambda_{k_{l+1}} & \text{if} \quad k_{l+1} < \infty \\ \nu_0 & \text{if} \quad k_{l+1} = \infty \end{cases} \tag{29}$$

$$\tau_{l,\omega}(y_1,\dots,y_l) = \begin{cases} 0 & \text{if} \quad k_{l+1} < \infty \quad \text{and} \quad \tau_i(z_1,\dots,z_i) = 0 \\ & \quad\text{for all } i \text{ with } k_l \le i < k_{l+1} \\ 1 & \text{if} \quad k_{l+1} < \infty \quad \text{and} \quad \tau_i(z_1,\dots,z_i) = 1 \\ & \quad\text{for some } i \text{ with } k_l \le i < k_{l+1} \\ 1 & \text{if} \quad k_{l+1} = \infty \end{cases} \tag{30}$$

$$\varphi_{l,\omega}(y_1,\dots,y_l) = \begin{cases} \varphi_{k_l}(z_1,\dots,z_{k_l}) & \text{if} \quad k_{l+1} < \infty \quad \text{and} \quad \tau_i(z_1,\dots,z_i) = 0 \\ & \quad\text{for all } i \text{ with } k_l \le i < k_{l+1} \\ \varphi_i(z_1,\dots,z_i) & \text{if} \quad i \text{ is the smallest idex with } k_l \le i < k_{l+1} \\ & \quad\text{and } \tau_i(z_1,\dots,z_i) = 1 \\ \varphi_0 & \text{if} \quad k_{l+1} = \infty \quad \text{and} \quad \tau_i(z_1,\dots,z_i) = 0 \\ & \quad\text{for all } i \text{ with } k_l \le i < \infty. \end{cases} \tag{31}$$

Since we defined these functions of finite strings by the help of an infinite string, correctness has to be checked in the sense that for each $l \in \mathbb{N}$ and each sequence $(\tilde{y}_j)_{j=1}^\infty \subset K$ with $y_j = \tilde{y}_j$ for all $j \le l$ the respective values of $L_{l+1,\omega}(y_1,\dots,y_l)$, $\tau_{l,\omega}(y_1,\dots,y_l)$, and $\varphi_{l,\omega}(y_1,\dots,y_l)$ coincide. But this follows readily from (28). This completes the definition of algorithm A_ω.

Now we show (20) and (21). Let $f \in F$ and define according to (17)

$$\lambda_1^* = L_1, \quad \lambda_i^* = L_i(\lambda_1^*(f,\omega),\dots,\lambda_{i-1}^*(f,\omega)) \quad (i \ge 2). \tag{32}$$

Furthermore, put $k_0^* = 0$ and set for $l \in \mathbb{N}$

$$k_l^* = \min\{i \in \mathbb{N} : \, i > k_{l-1}^*, \lambda_i^* \in \Lambda\} \tag{33}$$

$$y_l = \begin{cases} \lambda_{k_l^*}^*(f) & \text{if} \quad k_l^* < \infty \\ \nu_0(f) & \text{if} \quad k_l^* = \infty. \end{cases} \tag{34}$$

Define according to (22)–(27)

$$\Psi\big((y_l)_{l=1}^\infty\big) = \big((\lambda_i)_{i=1}^\infty, (z_i)_{i=1}^\infty, (k_l)_{l=0}^\infty\big).$$

We claim that

$$\big((\lambda_i)_{i=1}^\infty, (z_i)_{i=1}^\infty, (k_l)_{l=0}^\infty\big) = \big((\lambda_i^*)_{i=1}^\infty, (\lambda_i^*(f,\omega))_{i=1}^\infty, (k_l^*)_{l=0}^\infty\big). \tag{35}$$

To prove the claim, we show by induction that for all $i \in \mathbb{N}$

$$\lambda_j = \lambda_j^*, \quad z_j = \lambda_j^*(f,\omega) \quad (j \le i), \tag{36}$$

$$k_p = k_p^* \quad \text{for all} \quad p \le |\{j \le i : \, \lambda_j \in \Lambda\}|. \tag{37}$$

For $i = 1$ we have $k_0 = k_0^*$ and by (22) and (32),

$$\lambda_1 = L_1 = \lambda_1^*.$$

If $\lambda_1 \in \Lambda$, then by (27) and (33), $k_1 = 1 = k_1^*$, $|\{j \le 1 : \lambda_j \in \Lambda\}| = 1$, and by (23) and (34)

$$z_1 = y_1 = \lambda_1^*(f) = \lambda_1^*(f, \omega).$$

If $\lambda_1 \in \Lambda'$, then $|\{j \le 1 : \lambda_j \in \Lambda\}| = 0$ and

$$z_1 = \lambda_1(\omega) = \lambda_1^*(\omega) = \lambda_1^*(f, \omega).$$

This shows (36) and (37) for $i = 1$.

Now let $i \ge 1$ and assume (36) and (37) hold for i. From (25) and (32) we obtain

$$\lambda_{i+1} = L_{i+1}(z_1, \dots, z_i) = L_{i+1}(\lambda_1^*(f, \omega), \dots, \lambda_i^*(f, \omega)) = \lambda_{i+1}^*.$$

Define $l = |\{j \le i : \lambda_j \in \Lambda\}|$. Then $k_l = k_l^*$ and $k_{l+1} \ge i + 1$. If $\lambda_{i+1} \in \Lambda$, we have by (27) and (33), $k_{l+1} = i + 1 = k_{l+1}^*$, $|\{j \le i + 1 : \lambda_j \in \Lambda\}| = l + 1$, and by (26),

$$z_{i+1} = y_{l+1} = \lambda_{k_{l+1}^*}^*(f) = \lambda_{i+1}^*(f) = \lambda_{i+1}^*(f, \omega).$$

If $\lambda_{i+1} \in \Lambda'$, then $|\{j \le i + 1 : \lambda_j \in \Lambda\}| = l$ and

$$z_{i+1} = \lambda_{i+1}(\omega) = \lambda_{i+1}^*(\omega) = \lambda_{i+1}^*(f, \omega).$$

This completes the induction step, showing (36) and (37) for all i, which, in turn, implies (35).

Next we observe that for $l \in \mathbb{N}_0$

$$\big(L_{l+1,\omega}(y_1, \dots, y_l)\big)(f) = y_{l+1}. \tag{38}$$

Indeed, if $k_{l+1} < \infty$, then by (29) and (34)

$$\big(L_{l+1,\omega}(y_1, \dots, y_l)\big)(f) = \lambda_{k_{l+1}}(f) = y_{l+1}.$$

Similarly, if $k_{l+1} = \infty$, we get

$$\big(L_{l+1,\omega}(y_1, \dots, y_l)\big)(f) = \nu_0(f) = y_{l+1}.$$

To complete the proof of the proposition, we consider three cases. First we assume that $\mathrm{card}_{\bar{\Lambda}}(A, f, \omega) = \infty$. This means

$$\tau_i(z_1, \dots, z_i) = 0 \quad (i \in \mathbb{N}_0), \quad A(f, \omega) = \varphi_0.$$

Assume furthermore $k_l < \infty$ for all $l \in \mathbb{N}$, thus $\mathrm{card}_\Lambda(A, f, \omega) = \infty$. It follows from (30) and (31) that

$$\tau_{l,\omega}(y_1, \dots, y_l) = 0 \quad (l \in \mathbb{N}_0), \quad \varphi_{0,\omega} = \varphi_0,$$

hence $\mathrm{card}(A_\omega, f) = \infty$ and

$$A_\omega(f) = \varphi_{0,\omega} = \varphi_0 = A(f, \omega).$$

Next assume that there is an $l_1 \in \mathbb{N}_0$ such that $k_{l_1} < \infty$ and $k_{l_1+1} = \infty$, hence $\mathrm{card}_\Lambda(A, f, \omega) = l_1$, so (30) and (31) give

$$\tau_{l,\omega}(y_1, \dots, y_l) = 0 \quad (l < l_1), \quad \tau_{l_1,\omega}(y_1, \dots, y_{l_1}) = 1, \quad \varphi_{l_1,\omega}(y_1, \dots, y_{l_1}) = \varphi_0,$$

thus $\operatorname{card}(A_\omega, f) = l_1$ and

$$A_\omega(f) = \varphi_{l_1,\omega}(y_1, \dots, y_{l_1}) = \varphi_0 = A(f, \omega).$$

Finally, we assume $\operatorname{card}_{\bar\Lambda}(A, f, \omega) = n < \infty$. This means

$$\tau_i(z_1, \dots, z_i) = 0 \quad (i \in \mathbb{N}_0, i < n), \quad \tau_n(z_1, \dots, z_n) = 1, \quad A(f, \omega) = \varphi_n(z_1, \dots, z_n).$$

There is a unique $l_1 \in \mathbb{N}_0$ such that $k_{l_1} \le n < k_{l_1+1}$, thus $\operatorname{card}_\Lambda(A, f, \omega) = l_1$. By (30) and (31)

$$\tau_{l,\omega}(y_1, \dots, y_l) = 0 \quad (l < l_1), \quad \tau_{l_1,\omega}(y_1, \dots, y_{l_1}) = 1, \quad \varphi_{l_1,\omega}(y_1, \dots, y_{l_1}) = \varphi_n(z_1, \dots, z_n),$$

consequently, $\operatorname{card}(A_\omega, f) = l_1$ and

$$A_\omega(f) = \varphi_{l_1,\omega}(y_1, \dots, y_{l_1}) = \varphi_n(z_1, \dots, z_n) = A(f, \omega).$$

This proves (20) and (21).

□

Corollary 2.2. *For each $\mathscr{R}$-restricted randomized algorithm A for $\mathscr{P}$ there exists a randomized algorithm $\tilde A = (A_\omega)_{\omega\in\Omega}$ for $\mathscr{P}$ such that* (20) *and* (21) *hold. Moreover, if $k, n \in \mathbb{N}_0$ and $A \in \mathscr{A}^{\mathrm{ran}}_{n,k}(\mathscr{P}, \mathscr{R})$, then $\tilde A \in \mathscr{A}^{\mathrm{ran}}_n(\mathscr{P})$. Hence*

$$e^{\mathrm{ran}}_n(S, F, G) \le e^{\mathrm{ran}}_{n,k}(S, F, G). \tag{39}$$

Proof. This is a direct consequence of Proposition 2.1. If $A \in \mathscr{A}^{\mathrm{ran}}_{n,k}(\mathscr{P}, \mathscr{R})$, then the required measurability properties of $\tilde A$ follow from those of A and (20)–(21). Furthermore,

$$\operatorname{card}(\tilde A) = \sup_{f\in F} \mathbb{E} \operatorname{card}(A_\omega, f) = \sup_{f\in F} \mathbb{E} \operatorname{card}_\Lambda(A, f, \omega) \le n.$$

□

3 The power of restricted randomized algorithms

In this section we show the following

Theorem 3.1. *For each probability space with access restriction*

$$\mathscr{R} = \big((\Omega, \Sigma, \mathbb{P}), K', \Lambda'\big), \tag{40}$$

see (15)*, there is a problem $\mathscr{P} = (F, G, S, K, \Lambda)$ such that*

$$e^{\mathrm{ran}}_1(S, F, G) = 0, \tag{41}$$

while

$$e^{\mathrm{ran}}_{n,k}(S, F, G) = 1 \quad \textit{for all} \quad n, k \in \mathbb{N}_0. \tag{42}$$

For Monte Carlo algorithms that use only random bits or random variables with finite range this result is due to Traub and Woźniakowski [16], Th. 3.1, and Novak, see [3], Example 14.

Proof. Let $|\Omega|$ be the cardinality of Ω, let $\aleph = \max(|\Omega|, |\mathbb{N}|)$ and let $\aleph_1$ be any cardinal number $\aleph_1 > \aleph$. By Cantor's theorem, one could take, e.g., $\aleph_1 = 2^{\aleph}$, see [10], Th. 6. We construct a probability space $(Q, \mathscr{F}, \mu)$ as follows. (For the case $\aleph = |\mathbb{N}|$ of this construction, see, e.g., [1], p. 29–30, exercise 2.12 (d).) Let Q be any set with $|Q| = \aleph_1$. Define

$$\mathscr{F}_0 = \{B \subseteq Q : |B| \le \aleph\}, \quad \mathscr{F}_1 = \{B \subseteq Q : |Q \setminus B| \le \aleph\}, \quad \mathscr{F} = \mathscr{F}_0 \cup \mathscr{F}_1,$$

and put for $B \in \mathscr{F}$

$$\mu(B) = \begin{cases} 0 & \text{if} \quad B \in \mathscr{F}_0 \\ 1 & \text{if} \quad B \in \mathscr{F}_1. \end{cases}$$

Since the union of countably many sets $B_i \subseteq Q$ with $|B_i| \le \aleph$ satisfies $| \cup_{i\in\mathbb{N}} B_i| \le \aleph$ ([10], section 6, relations 6.1 and 6.4), it follows that $\mathscr{F}$ is a σ-algebra and μ is a (countably additive) probability measure on $(Q, \mathscr{F})$. The structure of the space $\mathscr{L}_2(Q, \mathscr{F}, \mu)$ is simple: Let $f : Q \to \mathbb{R}$ be $\mathscr{F}$-to-Borel measurable, thus

$$Q(f, a) := \{x \in Q : f(x) \le a\} \in \mathscr{F} \quad (a \in \mathbb{R}).$$

Observe that since

$$\bigcap_{a\in\mathbb{Q}} Q(f, a) = \emptyset, \quad \bigcup_{a\in\mathbb{Q}} Q(f, a) = Q,$$

with $\mathbb{Q}$ the set of rationals, it follows that

$$a_1 = \inf\{a \in \mathbb{R} : Q(f, a) \in \mathscr{F}_1\} \in \mathbb{R}$$

and

$$Q(f, a_1) = \bigcap_{a\in\mathbb{Q},\, a>a_1} Q(f, a) \in \mathscr{F}_1.$$

For all $a > a_1$ we have $Q(f, a) \setminus Q(f, a_1) \in \mathscr{F}_0$, thus

$$\{x \in Q : f(x) \ne a\} = \bigcup_{a\in\mathbb{Q},a<a_1} Q(f, a) \cup \bigcup_{a\in\mathbb{Q},a>a_1} (Q(f, a) \setminus Q(f, a_1)) \in \mathscr{F}_0.$$

Thus, f is constant except for a set of cardinality $\le \aleph$, that is, of μ-measure zero. Since each such function is obviously μ-square integrable, $\mathscr{L}_2(Q, \mathscr{F}, \mu)$ consists of all these functions. Let us mention in passing that the respective space $L_2(Q, \mathscr{F}, \mu)$ of equivalence classes of functions equal up to a set of μ-measure zero is one-dimensional and consists of equivalence classes of constant functions.

We let $\mathscr{P} = (B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}, \mathbb{R}, I_{Q,\mu}, \mathbb{R}, \Lambda)$ be defined by (2)–(4), with $(Q, \mathscr{F}, \mu)$ as above. Let $A_1 = ((Q, \mathscr{F}, \mu), (A_{1,x})_{x\in Q})$ be the classical Monte Carlo method (10) with one sample, which is an unrestricted randomized algorithm, see (11)–(13). Obviously, $\operatorname{card}(A_1, B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}) = 1$ and

$$A_{1,x}(f) = f(x) \quad (x \in Q, f \in F),$$

hence for each $f \in F$ the mapping $x \to A_{1,x}(f) = f(x)$ is $\mathscr{F}$-to-Borel measurable. Moreover,

$$\mu\Big\{x \in Q : A_{1,x}(f) = \int_Q f(y)d\mu(y)\Big\} = 1,$$

which means

$$e(I_{Q,\mu}, A, B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}, \mathbb{R}) = \sup_{f \in B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}} \int_Q |I_{Q,\mu}(f) - A_{1,x}(f)| d\mu(x) = 0,$$

proving (41).

Now let $n, k \in \mathbb{N}_0$ and let $A \in \mathscr{A}_{n,k}^{\mathrm{ran}}(\mathscr{P}, \mathscr{R})$. By Proposition 2.1 for each $\omega \in \Omega$ there is a deterministic algorithm $A_\omega = ((L_{i,\omega})_{i=1}^\infty, (\tau_{i,\omega})_{i=0}^\infty, (\varphi_{i,\omega})_{i=0}^\infty)$ for $\mathscr{P}$ so that

$$A_\omega(f) = A(f, \omega). \tag{43}$$

Consider the zero function $f_0(x) = 0$ $(x \in Q)$. For $\omega \in \Omega$ let, according to (6) and (7),

$$\delta_{t_{1,\omega}} = L_{1,\omega}, \quad \delta_{t_{i,\omega}} = L_{i,\omega}(f_0(t_{1,\omega}), \dots, f_0(t_{i-1,\omega})) = L_{i,\omega}(0, \dots, 0) \quad (i \geq 2), \tag{44}$$

thus

$$\begin{aligned} \operatorname{card}(A_\omega, f_0) &= \min\{i \in \mathbb{N}_0 : \tau_{i,\omega}(0, \dots, 0) = 1\} &(45)\\ A_\omega(f_0) &= \begin{cases} \varphi_0 & \text{if} \quad \operatorname{card}(A_\omega, f_0) \in \{0, \infty\} \\ \varphi_n(0, \dots, 0) & \text{if} \quad 1 \leq \operatorname{card}(A_\omega, f_0) = n < \infty. \end{cases} &(46)\end{aligned}$$

Let

$$B_0 = \{t_{i,\omega} : i \in \mathbb{N}, \omega \in \Omega\}. \tag{47}$$

Then

$$|B_0| \leq |\mathbb{N}| \times |\Omega| \leq |\mathbb{N}| \times \aleph = \aleph,$$

hence $B_0 \in \mathscr{F}_0$. Define $f_j \in B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}$ for $j \in \{1, 2\}$ by

$$f_j(x) = \begin{cases} 0 & \text{if} \quad x \in B_0 \\ (-1)^j & \text{if} \quad x \in Q \setminus B_0. \end{cases} \tag{48}$$

With (6) and (7) we obtain

$$\delta_{t_{1,\omega,j}} = L_{1,\omega}, \quad \delta_{t_{i,\omega,j}} = L_{i,\omega}(f_j(t_{1,\omega,j}), \dots, f_j(t_{i-1,\omega,j})) \quad (i \geq 2), \tag{49}$$

and

$$\begin{aligned} \operatorname{card}(A_\omega, f_j) &= \min\{i \in \mathbb{N}_0 : \tau_{i,\omega}(f_j(t_{1,\omega,j}), \dots, f_j(t_{i,\omega,j})) = 1\} &(50)\\ A_\omega(f_j) &= \begin{cases} \varphi_0 & \text{if} \quad \operatorname{card}(A_\omega, f_j) \in \{0, \infty\} \\ \varphi_n(f_j(t_{1,\omega,j}), \dots, f_j(t_{n,\omega,j})) & \text{if} \quad 1 \leq \operatorname{card}(A_\omega, f_j) = n < \infty. \end{cases} &(51)\end{aligned}$$

Using (45)–(51), it is readily checked by induction that for all $i \in \mathbb{N}$, $\omega \in \Omega$, and $j \in \{1, 2\}$ we have $t_{i,\omega} = t_{i,\omega,j}$, therefore

$$f_j(t_{i,\omega,j}) = 0, \quad \operatorname{card}(A_\omega, f_j) = \operatorname{card}(A_\omega, f_0), \quad A_\omega(f_j) = A_\omega(f_0).$$

Consequently,

$$
\begin{aligned}
&e(I_{Q,\mu}, A, B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}, \mathbb{R}) \\
&= \sup_{f\in B_{\mathscr{L}_2(Q,\mathscr{F},\mu)}} \int_\Omega |I_{Q,\mu}(f) - A_\omega(f)| d\mathbb{P}(\omega) = \max_{j=1,2} \int_\Omega |I_{Q,\mu}(f_j) - A_\omega(f_j)| d\mathbb{P}(\omega) \\
&\ge \frac{1}{2} \sum_{j=1,2} \int_\Omega |I_{Q,\mu}(f_j) - A_\omega(f_j)| d\mathbb{P}(\omega) = \frac{1}{2} \sum_{j=1,2} \int_\Omega |I_{Q,\mu}(f_j) - A_\omega(f_0)| d\mathbb{P}(\omega) \\
&\ge \frac{1}{2} \int_\Omega |I_{Q,\mu}(f_1) - I_{Q,\mu}(f_2)| d\mathbb{P}(\omega) = 1.
\end{aligned}
$$

This shows (42).

□

Acknowledgment: This paper is partly based on research carried out while the author was guest of the International Mathematical Research Institute MATRIX, Melbourne, during the program 'On the frontiers of high dimensional computation'. The author is also grateful to Thomas Müller-Gronbach and Klaus Ritter for a discussion on the topic of Theorem 3.1, to Pawel Przybyłowicz for pointing out reference [1], and to the referees, whose suggestions helped to improve the presentation.

References

[1] P. Billingsley, Probability and Measure, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Inc., New York, 1986.

[2] W. Gao, P. Ye, and H. Wang, Optimal error bound of restricted Monte Carlo integration on anisotropic Sobolev classes, Progr. Natur. Sci. (English Ed.) 16 (2006), 588–593.

[3] M. B. Giles, M. Hefter, L. Mayer, and K. Ritter, Random bit quadrature and approximation of distributions on Hilbert spaces, Found. Comput. Math., 2018. doi: 10.1007/s10208-018-9382-3.

[4] M. B. Giles, M. Hefter, L. Mayer, and K. Ritter, Random bit multilevel algorithms for stochastic differential equations, J. Complexity, in press, https://doi.org/10.1016/j.jco.2019.01.002, see also arXiv:1808.10623.

[5] S. Heinrich, Monte Carlo approximation of weakly singular integral operators, J. Complexity 22 (2006), 192–219.

[6] S. Heinrich, The randomized information complexity of elliptic PDE, J. Complexity 22 (2006), 220–249.

[7] S. Heinrich, Lower complexity bounds for parametric stochastic Itô integration, in: Monte Carlo and Quasi-Monte Carlo Methods 2016 (A. B. Owen, P. W. Glynn, eds.), Springer Proceedings in Mathematics & Statistics 241, Berlin, 2018, pp. 295–312.

[8] S. Heinrich, Complexity of stochastic integration in Sobolev classes, J. Math. Anal. Appl. (2019), in press, https://doi.org/10.1016/j.jmaa.2018.12.077.

[9] S. Heinrich, E. Novak, and H. Pfeiffer. How many random bits do we need for Monte Carlo integration? In: Monte Carlo and Quasi-Monte Carlo Methods 2002 (H. Niederreiter, ed.), Springer-Verlag, Berlin, 2004, pp. 27–49.

[10] T. Jech, Set Theory, Academic Press, New York, 1977.

[11] E. Novak, Eingeschränkte Monte Carlo-Verfahren zur numerischen Integration, Proc. 4th Pannonian Symp. on Math. Statist., Bad Tatzmannsdorf, Austria 1983, W. Grossmann et al. eds., Reidel, 1985, pp. 269-282.

[12] E. Novak, Deterministic and Stochastic Error Bounds in Numerical Analysis, Lecture Notes in Mathematics 1349, Springer-Verlag, 1988.

[13] E. Novak and H. Pfeiffer, Coin tossing algorithms for integral equations and tractability, Monte Carlo Methods Appl. 10 (2004), 491–498.

[14] T. Symul, S. M. Assad, P. K. Lam, Real time demonstration of high bitrate quantum random number generation with coherent laser light, Appl. Phys. Lett. 98, 231103 (2011).

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

[16] J. F. Traub and H. Woźniakowski, The Monte Carlo algorithm with a pseudorandom generator, Math. Comp. 58 (1992), 323–339.

[17] P. Ye and X. Hu, Optimal integration error on anisotropic classes for restricted Monte Carlo and quantum algorithms, J. Approx. Theory 150 (2008), 24–47.