# RANDOMNESS, PROVABILITY, AND THE SEPARATION OF MONTE CARLO TIME AND SPACE

MAREK KARPINSKI
RUTGER VERBEEK

UNIVERSITY OF BONN

**Abstract.**

Separation theorems are essential in complexity theory: looking for strict lower and upper bounds makes sense only in the context of a hierarchy theorem. For probabilistic complexity classes with deterministically constructible bounds the standard diagonalization techniques can be applied and yield at least as dense hierarchies as in the deterministic case. For Monte Carlo (i.e. bounded error probability) classes the situation is quite different. On one hand we can construct arbitrarily slowly growing Monte Carlo space constructible functions (even far below log log $n$) [KV 86], on the other hand the existence of different – even deterministically constructible – bounds is not sufficient for a proof of separation. Up to now there is no way to separate, e.g., Monte Carlo Time ($n$) from Monte Carlo Time ($n^{\log n}$). We are able, however, to display a method of separating Monte Carlo Time ($n^{\log n}$) from $\bigcap_{\varepsilon}$ Monte Carlo Time ($2^{n^{\varepsilon}}$).

Note that the Monte Carlo property of probabilistic algorithms is $\Pi_1$-complete. For diagonalization, however, an enumerable set of machines is required. For practical purposes the only interesting Monte Carlo algorithms are those which are provable in some reasonable theory (e.g. within Peano arithmetic or Zermelo-Fraenkel set theory). For such provable complexity classes dense space and time hierarchies are established, space hierarchies even below log log $n$ or $log^* n$.

## 0. Introduction.

It is known from [LSH 65] that any meaningful padding-control requires at least log log $n$ space. Therefore there is 'no life' for deterministic machines between $O(1)$ and $O(\log \log n)$.

In contrast to this Freivalds [Fr 81] displayed exponential (and therefore arbitrary elementary recursive) padding doable in $O(1)$ Monte Carlo space. We prove that even

arbitrary recursive paddings are achievable within $O(1)$ Monte Carlo space (Theorem 1). This enables the proofs of separation results for complexity classes with arbitrarily small bounds.

For the definition of *probabilistic Turing machines (PTMs)* see [Gi 77]. If $\mathcal{M}$ is a PTM, then $\phi_{\mathcal{M}}$ is the function computed by $\mathcal{M}$.

$f$ is in *probabilistic time* T(n) ($PrTIME(T(n))$) if, for some PTM $\mathcal{M}$, $\phi_{\mathcal{M}} = f$ and for all $x$

$$Pr\{\mathcal{M} \text{ stops after at most } t(|x|) \text{ steps and outputs } \phi_{\mathcal{M}}(x)\} > \frac{1}{2}.$$

If $\frac{1}{2}$ can be replaced by $\frac{3}{4}$, we call the corresponding machine a *Monte Carlo Turing machine* (MTM). The corresponding time complexity classes are denoted by $MTIME(T(n))$. Sets are recognized by PTMs or MTMs computing their characteristic functions.

$RTIME(T(n))$ (for sets only) are defined in the same way as Monte Carlo classes, with the restriction that the error probability is 0 on the complement of the recognized set. $f$ is in *probabilistic space* $(S(n))(PrSPACE(S(n)))$ if there is a PTM $\mathcal{M}$ such that $\phi_{\mathcal{M}} = f$ and for all $x$

$$Pr\{\mathcal{M} \text{ uses on input } x \text{ at most } S(|x|) \text{ space and outputs } \phi_{\mathcal{M}}(x)\} > \frac{1}{2}.$$

If in addition $\mathcal{M}$ always stops with probability 1, then $f \in Pr^{T}SPACE(S(n))$ (cf.[We 83]). $MSPACE(S(n))$, $M^{T}SPACE(S(n))$, $RSPACE(S(n))$, and $R^{T}SPACE(S(n))$ are defined in the same way as for the time complexity classes. A function $f : I\!N \to I\!N$ will be called *Monte Carlo (MC-)constructible* if there is a MTM $\mathcal{M}$ with space bound $f(n)$ for which for all $n$ there is some $x, |x| = n$, such that $\phi_{\mathcal{M}} = f(n)$. If $\mathcal{M}$ satisfies the above for $x = 0^n$, then $f$ is called *fully MC-constructible.*

## 1. Small Monte Carlo Space Constructible Functions.

We use two machine models

(i) off line two counter Turing machines (2CTs) [HU 79, p.171] and
(ii) classical (unary input) three counter (Minsky) machines (3CMs) [Mi 61].

A configuration of a 3CM $\mathcal{M}$ is to be encoded in the form $0^{q+1}1^{z_1}2^{z_2}3^{z_3}$, where $q$ is the state of $\mathcal{M}$, $z_i$ is the content of the $i$-th counter for $i = 1, 2, 3$. The code of a computation is a sequence of encoded $\mathcal{M}$-configurations according to its transition table. 3CMs are able to compute all partial recursive functions (with unary input/output) [Mi 61 ], whereas 2CMs are not [Ba 62]. In the case of 2CTs, configurations are encoded

by $0^{q+1}1^{z_1}2^{z_2}$ (note that we do not mind the input). 2CTs are able to compute all p.r. functions (input/output binary) [HU 79].

Given 3CM $M$ and an accepted input $n$, $comp_M(n)$ will denote the code of the accepting computation on $n$. If $n$ is not accepted, $comp_M(n)$ is undefined. In the same way $comp_T(x)$ is defined for 2CT $T$.

**Lemma 1.** For every 3CM $M$, $\{comp_M(n) \mid n \in I\!N\} \in MSPACE(O(1))$.

PROOF. The recognition of the set $\{comp_M(n)\}$ is based on the idea of Freivalds' example $\left\{0^{2^0}10^{2^1}1\cdots0^{2^k}1 \mid k \in I\!N\right\}$ [Fr 81] (cf. also Lemmas 1 and 2 of [Fr 81]) used for the exponential padding. A deterministic finite automaton can check whether the sequence of states is correct (the next state depends only on the zero-tests and the current state). What remains is to compare the (non-zero) contents of the counters in succeeding configuration by a sequence of tests of roughly the form "Is $n = m$ in $1^n0^+1^m$?" (the differences +1 or -1 can be handled in the finite control). These tests are performed by tossing $8n$ coins on $1^n$ and $8m$ coins on $1^m$. This procedure is repeated until two times the outcomes of all the $8n$ or $8m$ tosses were 'heads'. If this happens both times on the same substring, decide '$n \neq m$'; otherwise decide '$n = m$'. If $n = m$, $Pr\{\text{outcome is '}n = m\text{'}\} = \frac{1}{2}$; if $n \neq m$, $Pr\{\text{outcome is '}n = m\text{'}\} \leq 1 - \left(\frac{1}{1+2^{-8}}\right)^2 \leq 2^{-7}$.

Thus, in a sequence of $\ell$ tests the probability that all tests give the result

$$\text{'}n_i = m_i\text{' is } \begin{cases} 2^{-\ell} & \text{if } n_i = m_i \text{ for all } i, 1 \leq i \leq \ell \\ \leq 2^{-\ell-6} & \text{otherwise.} \end{cases}$$

Thus we must "compare" $Pr\{\text{all } \ell \text{ tests have outcome '}n = m\text{'}\}$ with $2^{-\ell}$ or, better, with $2^{-\ell-3}$. This is done in a way similar to the single comparisons:

```
begin repeat
            t1 := true; t2 := true;
            for i := 1 to ℓ do
            begin  compare n_i with m_i;
                   if 'n_i ≠ m_i' then  t1 := false;
                   toss a coin;
                   if  the outcome is 'tail' then t2 := false;
            end;
            toss 3 coins;
            if  one of them is 'tail' then  t2 := false;
      until  t1 or  t2;
      if  t1 then  write ('(∀i)n_i = m_i')
           else  write ('(∃i)n_i ≠ m_i')
```

**end**

Observe that this algorithm requires finite storage only (using a two way input tape, $i$ and $\ell$ need not be stored).

The probability analysis is quite simple:

If $(\forall i) n_i = m_i$, then $Pr\{\text{answer is } \neq\} \leq \frac{2^{-(\ell+3)}}{2^{-\ell}} < \frac{1}{4}$,
if $(\exists i) n_i \neq m_i$, then $Pr\{\text{answer is } =\} \leq \frac{2^{-\ell-6}}{2^{-(\ell+3)}} < \frac{1}{4}$. □

**Lemma 2.** For every recursive function $f : I\!N \to I\!N$ there is a 3CM $\mathcal{M}$ such that for all $n$ , $f(n) \leq |comp_{\mathcal{M}}(n)| \leq |comp_{\mathcal{M}}(n+1)|$.

PROOF Given recursive $f$, there is a 3CM $\mathcal{M}_1$ computing $f$. We construct a 3CM $\mathcal{M}$, computing $f' : n \mapsto f(0)\,f(1) \cdots f(n)$ in a canonical way. Then for all $n$, $f(n) \leq f'(n) \leq |comp_{\mathcal{M}}(n)| \leq |comp_{\mathcal{M}}(n+1)|$. □

**Theorem 1.** For every unbounded nondecreasing (*u.nd.*) recursive function $f$ there is an u.nd. MC-constructible minorant $g$ with $g(n) \leq f(n)$ for all $n$.

PROOF. Given $f : I\!N \to I\!N$, $F(n) := \max\{m \mid f(m) \leq n\}$. Take 3CM $\mathcal{M}$ of Lemma 2 for the function $F$, i.e. satisfying $F(n) \leq |comp_{\mathcal{M}}(n)| \leq |comp_{\mathcal{M}}(n+1)|$.

Define $g(n) = \min\{m \mid |comp_{\mathcal{M}}(m)| \geq n\}$. Construct by Lemma 1 an MTM $\mathcal{T}$ with space $O(1)$ that recognizes all strings having a prefix of the form $comp_{\mathcal{M}}(m)$ and outputs $m$ for such a word and 0 otherwise. Obviously on input $w$ the output is at most $g(|w|)$. For the input of the form $comp_{\mathcal{M}}(g(n))0^{n-|comp_{\mathcal{M}}(g(n))|}$ $\mathcal{T}$ computes exactly $g(n)$. □

The function $g$ constructed above has an important predictability property: for all $n$ there is an $x$ with $|x| = n$, such that the MTM constructing $g$ either outputs $g(n)$ (with probability $> \frac{3}{4}$) or outputs 0. We call such a function *predictably MC-constructible.*

## 2. Monte Carlo Constructibility and Diagonalization.

For most complexity measures constructibility of the bounds and closure under complement provides everything one needs for a separation (via diagonalization) of complexity classes with different bounds.

In the case of probabilistic classes with MC-constructible bounds the argument fails: $\forall x\ Pr\{\mathcal{M} \text{ uses at most } S(n) \text{ space}\} \geq \frac{3}{4}$ and $\forall x\ Pr\{\mathcal{M} \text{ outputs } \phi_{\mathcal{M}}(x)\} > \frac{1}{2}$ does *not* imply that $\forall x\ Pr\{\mathcal{M} \text{ outputs } \phi_{\mathcal{M}}(x) \text{ within space } S(n)\} > \frac{1}{2}$, and hence $\phi_{\mathcal{M}}$ is in general not in $PrSPACE(S(n))$.

For Monte Carlo classes, $\forall x\ Pr\{\mathcal{M} \text{ outputs } \phi_{\mathcal{M}}(x)\} > \frac{7}{8}$ and $\forall x\ Pr\{\mathcal{M} \text{ uses not more than } S(n) \text{ space}\} > \frac{7}{8}$ implies $\forall x\ Pr\{\mathcal{M} \text{ outputs } \phi_{\mathcal{M}}(x) \text{ within space } S(n)\} > \frac{3}{4}$, and

hence $\phi_M \in SPACE(S(n))$. If $M$ is Monte Carlo and terminating with probability 1, the error probability can be reduced to an arbitrarily small constant (e.g. $\frac{1}{8}$) without changing the space and time bound by more than a constant factor.

**Remark.** For PTMs the well-known speed-up theorem for deterministic or non-deterministic TMs (speed-up by a constant factor by enlarging the alphabet) seems not to be valid, since the degree of branching in random choices is exactly 2 for this model. □

Thus it seems possible to show a proper hierarchy for arbitrarily small bounds at least for the $M^TSPACE$ classes. For non-terminating MTMs it can be shown that $MSPACE(f(n)) \subseteq M^TSPACE(2^{2^{f(n)}})$ (see [KV 86]).

But for Monte Carlo complexity classes some other difficulty arises: beside the constructibility and closure under complement another property is required for diagonalization, namely recursive enumerability of the machines to be diagonalized.

The class of Monte Carlo machines is $\Pi_1$-complete. Of course there might be an enumerable subset such that for any Monte Carlo machine there is an equivalent one in that subset working in the same space (or time) bound, as it is the case for deterministic or probabilistic space classes with deterministically constructible bounds (in this case take the recursive set of explicitly bounded machines). But such a situation seems improbable in the Monte Carlo (or R) case.

A similar situation arises also in the case of some deterministic time complexity classes, where an explicit clock apparently requires additional time (e.g. for one-tape Turing machines); this is the reason why the time hierarchy is not dense for such classes (to our knowledge).

If we restrict ourselves to classes of "provable" machines (i.e. to enumerable subsets of the complexity classes), the hierarchy becomes dense.

**Example.** We consider deterministic one-tape Turing machines.

$L_t := \{M | M$ is a one-tape TM and there is $K$ with $t = \phi_K$ and there is a proof in Peano arithmetic for the fact that $M$ is $t(n)$-space bounded $\}$

one-tape-$TIME^{provable}(f) := \{\phi_M | M \in L_f\}$.

It is easy to show that for time-constructible $f, g$ with $g \notin O(f)$

$$\text{one-tape-}TIME^{provable}(f(n)) \not\supseteq \text{one-tape-}TIME^{provable}(g(n)).$$

We do not know, however, if under these assumptions one-tape-$TIME(f(n)) \not\supseteq$ one-tape-$TIME(g(n))$ or if one-tape-$TIME(f(n))$ = one-tape-$TIME^{provable}(f(n))$. □

We shall return to provable classes in section 4.

## 3. Separation by Padding.

We have seen that we have basically no tools for proving

$$MSPACE(f(n)) \not\supseteq MSPACE(g(n)) \quad \text{or}$$
$$MTIME(f(n)) \not\supseteq MTIME(g(n)) \quad \text{for } g \notin O(f)$$

unless $MSPACE(f(n))$ (or $MTIME(f(n))$, respectively) is a provable class. The lowest provable class containing an arbitrary MC-complexity class is (to our knowledge) the next deterministic complexity class, i.e.

$$MSPACE(f(n)) \subseteq DSPACE(f(n)^2), \text{ if } f(n) \geq \log n \text{ [BCP 83]}$$
$$MSPACE(f(n)) \subseteq DSPACE(\log^2 n), \text{ if } f(n) \leq \log n$$
$$MTIME(f(n)) \subseteq DTIME(2^{2f(n)}).$$

This implies

(1) $MSPACE(f(n)) \not\supseteq MSPACE(g(n))$, if $f$, $g$ are MC-space constructible and $g(n) \notin O(\max(\log^2 n, f(n)^2))$

(2) $MTIME(f(n)) \not\supseteq MTIME(g(n))$, if $f$, $g$ are time constructible and $g(n) \notin O(2^{2f(n)})$.

Especially for $TIME$-classes we have exponential gaps in the hierarchy; and we cannot prove a strict space hierarchy below $\log n$.

Some gaps can be closed by the padding technique introduced by Ruby and Fischer [RF 65] and Cook [Co 73], since the translation lemma is valid also for Monte Carlo classes.

**Lemma 3.**

(1) If $f(n), g(n), gh(n)$ are fully space constructible, $g(n) \geq \log n, h(n) \geq n$, then

$$MSPACE(f(n)) \subseteq MSPACE(g(n)) \quad \text{implies}$$
$$MSPACE(fh(n)) \subseteq MSPACE(gh(n)).$$

(2) If $f(n), g(n), h(n)$ are fully time constructible, $g(n), h(n) \geq n$, then

$$MTIME(f(n)) \subseteq MTIME(g(n)) \quad \text{implies}$$
$$MTIME(fh(n)) \subseteq MTIME(gh(n)).$$

PROOF. Exactly the same proof as for the deterministic or non-deterministic case (see e.g.[HK 79]). □

Thus we can use, e.g., the technique of Ibarra [Ib 72] to show rather dense space hierarchies above log $n$ (e.g. $MSPACE(\log^p n) \subsetneq MSPACE(\log^q n)$, if $1 \leq p < q$).

For non-deterministic time and space hierarchies there are several refinements of this argument (see [Co 73], [SFM 73], [Se 77]), but they all require a universal simulator in the greater bound, which is seemingly not available in the Monte Carlo case.

**Theorem 2.** Suppose $f$, $g$ are fully time constructible, $g(n) \geq 2^{n^\epsilon}$ for some $\epsilon > 0$, $\forall k\ f \circ (f(n))^k \leq g(n)$. Then $MTIME(f(n)) \subsetneq MTIME(g(n))$

PROOF. Assume $MTIME(f(n)) \supseteq MTIME(g(n))$, and w.l.o.g. $f(n) \geq n^2$. Then for $k \geq \lceil 2/\epsilon \rceil$

$$\begin{aligned} MTIME(2^{(f(n))^2}) &\subseteq MTIME(g((f(n))^k)) \\ &\subseteq MTIME(f((f(n))^k)) \text{ (by Lemma 3)} \\ &\subseteq MTIME(g(n)) \\ &\subseteq MTIME(f(n)) \text{ (a contradiction).} \end{aligned}$$

□

Theorem 2 entails separation of $n^{\log n}$ and $2^{n^\epsilon}$ Monte Carlo time:

$$MTIME(poly) \subseteq MTIME(n^{\log n}) \subsetneq \bigcap_\epsilon MTIME(2^{n^\epsilon})$$

but the separations

$$MTIME(n) \stackrel{?}{\subsetneq} MTIME(n^2)$$

or even

$$MTIME(n) \stackrel{?}{\subsetneq} MTIME(n^{\log n})$$

are still open.

There are even faster growing functions than $n^{\log n}$ for which this inclusion may be improper,

$$\begin{aligned} exp(n) &:= 2^n,\ \log(n) := \lceil \log_2(n) \rceil, \\ f(n) &:= n,\ f^{(i+1)}(n) := f \circ f^{(i)}(n), \\ f_i(n) &:= exp^{(i)}(1 + \log^{(i)}(n)). \end{aligned}$$

Then

$$f_0(n) = n+1,\ f_1(n) \approx 2n,\ f_2(n) \approx n^2,\ f_3(n) \approx n^{\log n},\ f_4(n) \approx n^{(\log n)^{\log \log n}} \text{ etc.,}$$

and for all $i$, $k$, $\varepsilon$ and almost all $n$

$f_i \circ (f_i(n))^k = exp^{(i)}(1 + \log^{(i)}(exp^{(i)}(1 + \log^{(i)}(n)))^k) \leq exp^{(i)}(1 + k \cdot \log^{(i)}(n)) < f_{i+2}(n) \leq 2^{n^\varepsilon}$.

Hence

$$MTIME(f_i(n)) \subsetneqq \bigcap_{\varepsilon} MTIME(2^{n^\varepsilon}),$$

but

$$MTIME(n) \stackrel{?}{\subsetneqq} MTIME(f_i(n))$$

is open.

## 4. Provability and the Monte Carlo Dense Time and Space Hierarchies.

Unlike the cases of deterministic and non-deterministic computations the existence of distinct constructible complexity bounds does not automatically guarantee the separation of the corresponding Monte Carlo complexity classes. This is because the definition of Monte Carlo algorithms and the complexity classes is not effective: the Monte Carlo property is not decidable (in fact it is $\Pi_1$-complete). This leads to the situation (discussed in section 3) that no one knows whether $MSPACE(\log n) \neq MSPACE(\log n \log\log n)$ (more generally, whether $MSPACE(f) \neq MSPACE(g)$ for $f = o(g)$) or $MTIME(n^{r_1}) \neq MTIME(n^{r_2})$ or, even much more embarrassingly, whether $MTIME(n) \neq MTIME(n^{\log n})$ ??). It is also clear that for practical purposes the only interesting class of probabilistic algorithms is the class provable within some reasonable theory, e.g. Peano arithmetic or Zermelo-Fraenkel set theory. In order to prove the correctness of an algorithm design, so to speak, we must provide a guarantee for being Monte Carlo within some theory.

We are now going to formulate results on the provable Monte Carlo and the randomised complexity classes, both for randomised space and time.

For a fixed sound enumerable theory $\tau$ (e.g., Peano arithmetic) let $\underline{\mathcal{M}}^\tau$ be the set of all PTMs $\mathcal{M}$ for which "$\forall x\, Pr\{M \text{ outputs } \phi_{\mathcal{M}}(x)\} > \frac{7}{8}$" is a theorem of $\tau$. Let us denote by $M^\tau SPACE(S(n))$ and $M^\tau TIME(T(n))$ the corresponding complexity classes:

$$M^\tau SPACE(S(n)) = \{\phi_{\mathcal{M}} | \mathcal{M} \in \underline{\mathcal{M}}^\tau \text{ and } `\forall x\, Pr\{\mathcal{M} \text{ uses at most } S(|x|) \text{ space}\} > \frac{7}{8}'\}$$

$$M^\tau TIME(T(n)) = \{\phi_{\mathcal{M}} | \mathcal{M} \in \underline{\mathcal{M}}^\tau \text{ and } `\forall x\, Pr\{\mathcal{M} \text{ works in } T(|x|) \text{ time}\} > \frac{7}{8}'\}.$$

**Remark.** The bound on the error probability of $\frac{1}{8}$ may seem a bit arbitrary. For terminating (with probability 1) Monte Carlo machines $\mathcal{M}$, we can decrease the error

probability and the probability that $\mathcal{M}$ uses more than $S(n)$ space effectively to an arbitrarily small constant without using more space. For Monte Carlo machines with very small space bounds which diverge with a probability greater than 0, this may be false.

It is *not* known if the Monte Carlo complexity classes contain complete problems. In case it is indeed so, we have the identities $M^{\tau}SPACE(S(n)) = MSPACE(S(n))$ and $M^{\tau}TIME(T(n)) = MTIME(T(n))$, accordingly.

Any provable Monte Carlo complexity class $M^{\tau}SPACE(S(n))$ (or $M^{\tau}TIME(T(n))$, respectively) contains a canonical complete problem (a universal language for $M^{\tau}SPACE(S(n))$ restricted to machines with fixed alphabet or for $M^{\tau}TIME(T(n))$ restricted to a fixed alphabet). Thus $MSPACE(S(n))$ contains a complete problem $C$ if and only if $MSPACE(S(n)) = M^{\tau}SPACE(S(n))$ for any reasonable sound (axiomatisable) theory $\tau$ containing the theorem '$C$ is complete for $M^{\tau}SPACE(S(n))$'. A similar statement is true for $MTIME$.

The above definition can be applied to the polynomial time complexity classes, e.g. $BPP^{\tau}$, the class of all sets recognized by polynomial time bounded error PTMs which are provably Monte Carlo; or $R^{\tau}$ the class of all sets recognized by polynomial time PTMs which have provably one-sided error (randomised). While it is a well-known open problem whether $BPP$ and $R$ do contain $\leq_{pol}$-complete problems, $BPP^{\tau}$ and $R^{\tau}$ do possess complete problems.

By the construction of Hartmanis and Hemachandra [HH 86] a complete set for $BPP$ (or $R$) exists if and only if there is a complete set of the form $L \cap MAJ$, where $L$ is in $P$ (or even in $DSPACE(\log n)$, by more careful inspection of the proof) and $MAJ$ is the set of boolean formulas that are satisfied by more than half of the truth assignments to the variables, which is complete for $PP = PrTIME(poly)$ [Gi 77]. Sipser [Si 82] shows that the existence (or non-existence) of complete sets for $R$ is not relativizable: there are oracles $A, B$ such that $R^A$ contains a complete set and $R^B$ does not. Thus the known proof methods for complete sets (which all relativize) must fail in a randomised case.

Let $\tau_1, \tau_2$ be enumerable theories, $\tau_1 \subseteq \tau_2$, then for all bounds $S(n)$, $T(n)$

$$M^{\tau_1}SPACE(S(n)) \subseteq M^{\tau_2}SPACE(S(n)) \subseteq MSPACE(S(n))$$

and

$$M^{\tau_1}TIME(T(n)) \subseteq M^{\tau_2}TIME(T(n)) \subseteq MTIME(T(n)).$$

**Theorem 3.** ([KV 86]) If $\tau$ is an enumerable theory, $g$ is MC-constructible and $f = o(g)$, then $M^{\tau}SPACE(f) \subsetneq M^{\tau}SPACE(2^{2^g})$.

PROOF. By diagonalization over all machines which are provably (in given $\tau$) Monte Carlo and working in $f(n)$-space. Bring the diagonalizing $\mathcal{M}$ machine to the "halting

with probability $1^n$ form ([KV 86], Theorem 2). The resulting machine $\mathcal{M}'$ works provably in space $2^{2^{g(n)}}$. Application of majority vote provably reduces the error probability to $\frac{1}{4}$. We have

$$\phi_{\mathcal{M}} = \phi_{\mathcal{M}'} \in M^{\tau}SPACE(2^{2^g}) \setminus M^{\tau}SPACE(f).$$

□

**Theorem 4.** If $\tau$ is an enumerable theory, $g$ is MC-constructible, $g(n) \geq \log n$, and $f = o(g)$, then $M^{\tau}SPACE(f) \subsetneq M^{\tau}SPACE(g)$.

PROOF. Same diagonalization procedure as in Theorem 3. By [Si 81], for every Monte Carlo machine $\mathcal{M}$ working in space $g(n)$ such that $g(n) \geq \log n$, there exists an equivalent Monte Carlo machine halting with probability 1 and working in the same space $g(n)$. □

The summary of Monte Carlo space separation results is given in Fig. 1.
We are now going to prove the hierarchy result on Monte Carlo time classes.

**Theorem 5.** If $\tau$ is an enumerable theory, $g$ is MC-time constructible, and $f(n) \cdot \log f(n) = O(g(n))$, then $M^{\tau}TIME(f) \subsetneq M^{\tau}TIME(g)$.

PROOF. By diagonalization similar to Theorem 4. □

At the end we formulate dense hierarchy theorems (both for space and time) for the randomised classes, $R^{\tau}$SPACE and $R^{\tau}$TIME.

**Theorem 6.** There exists a dense hierarchy in $\{RSPACE(S(n))\}, S(n) \geq \log n$.

PROOF. $RSPACE(S(n)) = NSPACE(S(n))$ ([Gi 77]). The rest follows from [SFM 73] and [Se 77]. □

**Theorem 7.** If $\tau$ is an enumerable theory, then there exists a dense hierarchy in $\{R^{\tau}TIME(T(n))\}$.

PROOF. The notion of a "universal simulator" of [SFM 73] generalizes to the provably randomised case. This enables us to prove a dense hierarchy for provably randomized time classes $R^{\tau}TIME(T(n))$ (almost as dense as those of non-deterministic time, cf. [SFM 73]). □

The summary of Monte Carlo time separation results is given in Fig. 2.

**Remark** Unlike the non-deterministic case (where two tapes are as powerful as $k$ [Se 77]), $k+1$ tapes seem to be more powerful than $k$ in the Monte Carlo and randomised case. Therefore a strict inclusion in Theorem 5 and 7 seemingly requires that $f(n) \cdot$

$\log f(n) = o(g(n))$ (for $M^rTIME$) or $f(n+1) \cdot \log f(n) = o(g(n))$ (for $RTIME$). Using padding techniques (i.e. Lemma 3, which is also true for the provable classes for any reasonable theory), this can be relaxed to $f(n) \cdot \log^\varepsilon f(n) = O(g(n))$ (or $f(n+1) \cdot \log^\varepsilon f(n) = O(g(n))$) for any $\varepsilon > 0$. The restriction to $k$-tape machines ($k \geq 2$) yields a proper inclusion if $f(n) = o(g(n))$ (or $f(n+1) = o(g(n))$) (using the construction of Fürer [Fü 82]).

In the probabilistic (unbounded error) case, a tape reduction as in the non-deterministic case is available *and* the complexity classes are closed under complement (which yields a standard diagonalization). Thus the probabilistic time hierarchy (without restriction on the number of tapes) is even more dense than the deterministic or non-deterministic hierarchies.

## 5. Randomised Circuits.

A parallel model of Monte Carlo computation (***randomised uniform boolean circuits***) was introduced in [Co 85]. The mode of computation are randomised uniform boolean circuits. $RNC^k$ is the class of Monte Carlo circuits $\{C_n\}$, $n \in \mathbb{N}$, in $\log^k$-depth (depth$(C_n) = O(\log^k n)$) and $n^{O(1)}$size (size $(C_n) = n^{O(1)}$). Uniformity means that the mapping $n \mapsto \bar{C}_n$ ($\bar{C}_n$ is a binary string coding of the circuit $C_n$) is computable in deterministic log(size $(C_n)$)-space (cf.[Co 85]). The circuits $C_n$ are Monte Carlo circuits if their bitwise error probability is bounded away by $\frac{1}{4}$ (cf[Co 85]). We define $RNC = \bigcup_k RNC^k$ (by analogy to the *deterministic* $NC = \bigcup_k NC^k$). [KUW 85] suggest that the randomised (Monte Carlo circuits) would be exponentially more powerful than the deterministic ones, i.e. $NC \neq RNC$.

It is clear that for arbitrary uniform probabilistic circuits families the property of being Monte Carlo is $\Pi_1$-complete (the sequential randomised machines alike).

Denote by $MC_n$ the set of all probabilistic circuits $C$ of $n$ inputs which satisfy the *Monte Carlo property (MCP)*, i.e. fulfilling the formula '$\forall x \in \{0,1\}^n \left[Pr\{C \text{ outputs } 1\} \notin [\frac{1}{4}, \frac{3}{4}]\right]$'. The 'brute force' algorithm to check the circuit for the Monte Carlo property costs $2^{n^{O(1)}}$ time. Therefore, for every probabilistic circuit $C$ on $n$ inputs can give $2^{n^{O(1)}}$ proof (guarantee), say in the Propositional Calculus ($P_0$). The problem, though, is: can we find for all Monte Carlo circuits $C$ another equivalent Monte Carlo circuit $C'$ with the *small* boolean (propositional) proof $g$ for the MCP. Using the techniques of [Ad 78],[BG 81] and [AB-O 84] we can prove the following statement: for arbitrary Monte Carlo circuits $C$ there exists an equivalent Monte Carlo circuit $C'$ with *small* (polynomial) propositional proof for the MCP. Adleman [Ad 78] notices that the circuit $C'$ can be chosen to be deterministic.

In designing $RNC$-algorithms $R = \{C_n\}$, a ('*fabrication*') guarantee $g_n$ for all circuits $C_n$ of being Monte Carlo, $C_n \in \underline{MC}_n$, must be provided by their designers along

with the correctness proof for $R$. It is always done in some obvious proof system. Therefore, for practical purposes a much stronger notion of the randomised uniformity is needed (for detailed discussion see [KV 87]). Let us fix some sound axiomatizable proof system $\tau$ in which we are ready to formulate and check up our proofs (guarantees). Call it our *circuit guarantee-system (g-system) according to* $\tau$.

We call a class of Monte Carlo circuits $\{(n)\}$ ***strongly uniform*** provided the mapping $n \mapsto (\bar{C}_n, g_n)$, $g_n$ is a proof for '$C_n \in \underline{MC}_n$', is computable in the log (size$(C_n)$ + size$(g_n)$)- deterministic space.

Denote now by $R^qNC^k$ the class of all problems solvable by strongly uniform Monte Carlo $\log^k n$-depth, poly-size circuits. $R^qNC = \bigcup_k R^qNC^k$.

The *Circuit Value Problem (CV)* ([La 75], [Go 77] is known to be log space complete ($NC^1$-complete [Co 85]) for $P$ (even when restricted to the monotone boolean circuits). Let us now define the Provably Randomised Circuit Value Problem,
$R^qCV = \{(C, x; g) \,|\, C(x) = 1,\ g$ is a proof for '$C \in \underline{MC}_{|x|}$'$\}$.
Define also
$R^qCV^k = \{(C, x; g) \,|\, C(x) = 1,\ g$ is a proof for '$C \in \underline{MC}_{|x|}$' and depth $(C) \leq \log |x|\}$.

The theorems below on the completeness of the Randomised Circuits Value Problem were proven in the accompanying paper [KV 87] on parallel computation and were motivated by the completeness results of section 3.

**Theorem 8.** For every sound axiomatizable proof system $\tau$, $R^qCV$ is complete for $BPP^\tau$, and $R^qCV^k$ is $NC^i$-complete for $R^qNC^k$.

Since, if $BPP$ has a complete problem, then $BPP = BPP^\tau$ for all sound systems $\tau$, and also $BPP^\tau \leq NC^iR^qCV$, we have:

**Theorem 9.** If $BPP$ has a complete problem, then the Randomised Circuit Value Problem $R^qCV$ is complete for $BPP$, and $R^qCV^k$ is $NC^i$-complete for $BPP^k$ for all $k$, for certain sound axiomatizable systems $\tau$.

Theorems 8 and 9 once more raise the question of provability of the $BPP$-class and connect it directly with the problem of the existence of a natural complete circuit value problem in $BPP$ (and $RNC^k$). An interesting outcome of our Theorem 9 is that if $BPP$ has a complete problem, then for every Monte Carlo circuit there exists (within the same depth and size) an equivalent Monte Carlo circuit with the small and easy to compute MCP certificate, and vice versa. The nonexistence of complete sets in $BPP$ entails impossibility of ***easy*** MCP circuit certificates.

**Acknowledgements.**

The authors thank Steve Cook, Charlie Rackoff, and Ruediger Reischuk for a number of interesting conversations during the last Oberwolfach Complexity Theory conference in November 1986. It was Steve Cook's suggestion to test limits of classical "padding" techniques in separating randomised classes.

**References**


[Ad 78] Adleman, L.,
*Two Theorems on Random Polynomial Time*
Proc. $19^{th}$ IEEE FOCS (1978), pp. 75-83

[AMa 77] Adleman, L., and Manders, K.,
*Reducibility, Randomness and Intractability*
Proc. $9^{th}$ ACM STOC (1977), pp. 151-163

[AB-O 84] Ajtai, M., and Ben-Or, M.,
*A Theorem on Probabilistic Constant Depth Computations*
Proc. $16^{th}$ ACM STOC (1984), pp. 471-474

[AW 85] Ajtai, M., and Wigderson, A.,
*Deterministic Simulation of Probabilistic Constant Depth Circuits*
Proc. $26^{th}$ IEEE FOCS (1985), pp. 11-19

[Ba 62] Barzdin, Ya.M.,
*On One Class of Turing Machines (Minsky Machines)*
Algebra and Logic Seminar, Novosibirsk **6**, (1962), pp. 42-51 (Russian)

[BCP 83] Borodin, A., Cook, S., and Pippenger, N.,
*Parallel Computation for Well-Endowed Rings and Space-Bounded Probabilistic Machines*
Information and Control **58** (1983), pp. 113-136

[BG 81] Bennet, C., and Gill, J.,
*Relative to a Random Oracle A, $P^A \neq NP^A \neq co\text{-}NP^A$ with Probability 1*
SIAM J. Comput. **10** (1981), pp. 96-114

[BGM 82] Babai, L., Grigoryev, D.Yu., and Mound, D.M.,
*Isomorphism of Graphs with Bounded Eigenvalue Multiplicity*
Proc. $14^{th}$ ACM STOC (1982), pp. 310-324

[BGS 75] Baker, T., Gill, J., and Solovay, R.,
*Relativizations of the $P = NP$? question*
SIAM J. Comput. **4** (1975), pp. 431-442

[BS 83] Berman, P., and Simon, J.,
*Lower Bounds on Graph Threading by Probabilistic Machines*
Proc. $24^{th}$ IEEE FOCS (1983), pp. 304-311

[Co 71] Cook, S.A.,
*The Complexity of Theorem-Proving Procedures*
Proc. $3^{rd}$ ACM STOC (1971), pp. 151-158

[Co 73] Cook, S.A.,
*A Hierarchy for Non-deterministic Time Complexity*
J. Comput. System Sci. **7** (1973), pp. **343-353**

[Co 85] Cook, S.A.,
*A Taxonomy of Problems with Fast Parallel Algorithms*
Information and Control **64** (1985), pp. 1-22

[Fe 57] Feller, W.,
*An Introduction to Probability Theory and its Applications*
John Wiley, New York 1957

[Fr 81] Freivalds, R.,
*Probabilistic Two-Way Machines, MFCS '81*
Springer LNCS **118** (1981), pp. 33-45

[Fü 82] Fürer, M.,
*The Tight Deterministic Time Hierarchy*
Proc. $14^{th}$ ACM STOC (1982), pp. 8-16

[Gi 77] Gill, J.,
*Computational Complexity of Probabilistic Turing Machines*
SIAM J. Comput. **6** (1977), pp. 675-694

[GJ 79] Garey, M.R., and Johnson, D.S.,
*Computers and Intractability: A Guide to the Theory of NP-Completeness*
W.H. Freeman, San Francisco (1979)

[Go 77] Goldschlager, L.M.,
*The Monotone and Planar Circuit Value Problema are Log Space Complete for P*
SIGACT News **9** (1977), pp. 25-29

[HH 86] Hartmanis, J. and Hemachandra, M.,
*Complexity Classes without Machines: On Complete Languages for UP*
Proc. $13^{th}$ ICALP '86, Springer, LNCS **226** (1986), pp. 121-135

[HU 67] Hopcroft, J.E., and Ullman, J.D.,
*An Approach to a Unified Theory of Automata*
The Bell System Technical J., vol. 46, no. 8, (1967), pp. 1793-1829

[HU 79] Hopcroft, J.E., and Ullman, J.D.,
*Introduction to Automata Theory, Languages, and Computation*
Addison-Wesley, Reading, Ma., (1979)

[Ib 72] Ibarra, O.H.,
*A Note Concerning Non-deterministic Tape Complexities*
J. ACM **19** (1972), pp. 608-612

[Ju 84] Jung, H.,
*On Probabilistic Tape Complexity and Fast Circuits for Matrix Inversion Problems*
Proc. $11^{th}$ ICALP '84, Springer LNCS **172** (1984), pp. 281-291

[KUW 85] Karp, R.M., Upfal, E., and Wigderson, A.,
*Are Search and Decision Problems Computationally Equivalent*
Proc. $17^{th}$ ACM STOC (1985), pp. 464-475

[KV 86] Karpinski, M., and Verbeek, R.,
*On the Monte Carlo Space Constructible Functions and Separation Results for Probabilistic Complexity Classes*
Research Report #854-CS, University of Bonn (1986), submitted to Information and Control

[KV 87] Karpinski, M., and Verbeek, R.,
*Randomised NC-Classes and the Provably Randomised Circuit Value Problem*
Research Report # 8511-CS, University of Bonn (1987), to be submitted

[La 75] Ladner, R.E.,
*The Circuit Value Problem is Log Space Complete for P*
SIGACT News **7** (1975), pp. 18-20

[LSH 65] Lewis, P.M., Stearns, R.E., and Hartmanis, J.,
*Memory Bounds for Recognition of Context-Free and Context-Sensitive Languages*
Proc. $6^{th}$ IEEE Symp. on Switching Circuit Theory and Logical Design (1965), pp. 191-202

[Mi 61] Minsky, M.L.,
*Recursive Unsolvability of Post's Problem of 'Tag' and Other Topics in the Theory of Turing Machines*
Annals of Math. **74** (1961), pp. 437-455

[Ra 82] Rackoff, C.,
*Relativized Questions Involving Probabilistic Algorithms*
J. ACM **29** (1982), pp. 261-268

[Ro 67] Rogers, H.,
*The Theory of Recursive Functions and Effective Computability*
McGraw-Hill, New York (1967), pp. 1-482

[RF 65] Ruby, S., and Fischer, P.C.,
*Translational Methods in Computational Complexity*
IEEE Conference Record on Switching Circuit Theory and Logical Design, Ann Arbor (1965), pp. 173-178

[Se 77] Seiferas, J.I.,
*Techniques for Separating Space Complexity Classes*
J. Comput. System Sci. **14** (1977), pp. 73-99

[Si 82] Sipser, M.,
*On Relativization and the Existence of Complete Sets*
Proc. $9^{th}$ ICALP '82, Springer LNCS **140** (1982), pp. 521-531

[Si 83] Sipser, M.,
*Borel Sets and Circuit Complexity*
Proc. $15^{th}$ ACM STOC (1983), pp. 61-69

[SS 77] Solovay, R., and Strassen, V.,
*A Fast Monte Carlo Test for Primality*
SIAM J. Comput. **6** (1977), pp. 84-85

[We 83] Welsh, D.J.A.,
*Randomised Algorithms*
Discrete Applied Mathematics **5** (1983), pp. 133-145