ON ASYMPTOTIC ESTIMATES
FOR ARITHMETIC COST FUNCTIONS

CARLOS GUSTAVO T. DE A. MOREIRA
(Communicated by Andreas R. Blass)

Abstract. Let \( \tau(n) \) (the cost of \( n \)) be the minimum number of arithmetic operations needed to obtain \( n \) starting from 1. We prove that \( \tau(n) \geq \frac{\log n}{\log \log n} \) for almost all \( n \in \mathbb{N} \), and, given \( \varepsilon > 0 \), \( \tau(n) \leq \frac{(1+\varepsilon)\log n}{\log \log n} \) for all \( n \) sufficiently large. We prove analogous results for costs of polynomials with integer coefficients.

Introduction

The aim of this work is the study of arithmetic cost functions. The most basic case is the cost of an integer, which is the minimum number of arithmetic operations needed to obtain the integer starting from 1. That is, given \( k \in \mathbb{Z} \), \( \tau(k) \) is defined as the minimum \( m \in \mathbb{N} \) such that there exists a sequence \((s_0, s_1, \ldots, s_m)\) where \( s_0 = 1 \), \( s_m = k \) and for each \( \ell \geq 1 \) there are \( i, j \), \( 0 \leq i, j < \ell \), with \( s_\ell = Op(s_i, s_j) \), \( Op \in \{+, -, \cdot\} \). This function plays an important role in the paper by Shub and Smale ([SS]), in which some questions are proposed about the behavior of the function \( \tau \) (which we shall discuss later) that have consequences in the study of an algebraic version of the famous and fundamental problem in computer science known as “\( NP \neq P \)” (see [SS]). In [SS] and in [MS] the following universal bounds for \( \tau \) are presented:

\[
\log \log \log m + 1 \leq \tau(m) \leq 2 \log_2 m, \quad \forall m \in \mathbb{N}.
\]

The first inequality cannot be improved, since \( \tau(2^{2^k}) = k + 1 \). These inequalities are proved in the paper by de Melo and Svaiter ([MS]), where it is also proved that for any fixed \( \varepsilon > 0 \), we have \( \tau(n) \geq \frac{\log n}{(\log \log n)^{1+\varepsilon}} \) for almost all \( n \in \mathbb{N} \).

Here we improve the above results, by proving the following:

For almost all \( n \in \mathbb{N} \) we have \( \tau(n) \geq \frac{\log n}{\log \log n} \) and, for any given \( \varepsilon > 0 \), we have \( \tau(n) \leq \frac{(1+\varepsilon)\log n}{\log \log n} \), for every \( n \in \mathbb{N} \) sufficiently large. More precisely, given \( \varepsilon > 0 \),

\[
\begin{align*}
\tau(n) &\geq \frac{\log n}{\log \log n} + (1-\varepsilon)\frac{\log n}{\log \log n}, \quad \text{for almost all } n \in \mathbb{N}, \\
\tau(n) &\leq \frac{\log n}{\log \log n} + (3+\varepsilon)\frac{\log n}{\log \log n}, \quad \text{for } n \text{ large enough.}
\end{align*}
\]

Moreover, the first inequality does not depend on the number of binary operations that we can use (provided that this number is finite).

Received by the editors October 25, 1994 and, in revised form, August 15, 1995.

1991 Mathematics Subject Classification. Primary 11Y16; Secondary 68Q25, 68Q15, 11B75.

©1997 American Mathematical Society
Subsequently, we extend our results for more general cost functions, such as cost functions of polynomials, and discuss some open problems.

1. The lower estimates

**Theorem 1.** Suppose that we have $O$ binary or unary operations in the definition of $\tau$. Then $N(k) = \# \{ n \in \mathbb{N} \mid \tau(n) \leq k \}$ satisfies $N(k) \leq A^k \cdot k^k$ for some constant $A > 0$.

**Proof.** Let $\Omega = \{\text{Op}_1, \text{Op}_2, \ldots, \text{Op}_\Omega\}$ be the set of operations. Then $\tau(n) = k \Rightarrow \exists (s_0, s_1, \ldots, s_k); s_0 = 1$, and for each $\ell \geq 1 \exists i, j, 0 \leq i, j < \ell$, such that $s_\ell = \text{Op}(s_{i}, s_{j})$, $\text{Op} \in \Omega$, with $s_k = n$. Let $(r_1, r_2, \ldots, r_{2k}) := (i_1, j_1, i_2, j_2, \ldots, i_k, j_k)$. We must have $\{i_1, j_1, \ldots, i_k, j_k\} = \{0, 1, 2, \ldots, k - 1\}$, since otherwise we would create an unnecessary $s_i$, and so $\tau(n) < k$. Moreover, we may assume that there exists a sequence $1 \leq \ell_1 < \ell_2 < \cdots < \ell_k \leq 2k$ such that $r_{\ell_i} = i - 1$, for $i = 1, 2, \ldots, k$. Indeed, we may assume $i_\ell \leq j_\ell$, for $\ell = 1, 2, \ldots, k$ (perhaps by creating the new operations $\text{Op}_j^p(x, y) = \text{Op}_j(y, x)$), and, if $P(i) := \min \{ j \in \{1, 2, \ldots, 2k\} \mid r_i = j \}$, for $i = 0, 1, \ldots, k - 1$, we shall have, perhaps rearranging the order of the operations, that $P(0) < P(1) < \cdots < P(k - 1)$, because if $P(i) > P(i + 1)$, then we do not use $s_i$ to create $s_{i+1}$, and can therefore create $s_{i+1}$ before $s_i$. Now we can take $\ell_i := P(i)$.

Let $N'(k) = \# \{ n \in \mathbb{N} \mid \tau(n) = k \}$. By the above arguments, $N'(k) \leq O^k \cdot N''(k)$, where $N''(k) = \# \{(r_1, r_2, \ldots, r_{2k}) \mid r_i \in \{0, 1, \ldots, k - 1\} \text{ there are } 1 \leq \ell_1 < \ell_2 < \cdots < \ell_k \leq 2k \text{ with } r_{\ell_i} = i - 1 \text{ for } i = 1, 2, \ldots, k \}$, but then $N''(k) \leq C_{2k}^k \cdot k^k \leq 4^k \cdot k^k$; thus $N'(k) \leq (4^k)^k \cdot k^k$, and so $N(k) \leq \sum_{r=0}^{k} N'(r) \leq A^k \cdot k^k$, for $A = 4^k + 1$.

**Corollary.** Given $\varepsilon > 0$, we have, for almost all $n \in \mathbb{N}$, that

$$
\tau(n) \geq \frac{\log n}{\log \log n} + (1 - \varepsilon) \frac{\log n \log \log \log n}{(\log \log n)^2} =: f(n)
$$

**Proof.** Let us compute $B(n) = \# \{ k \leq n \mid \tau(k) \leq f(n) \}$. If $k \in B(n)$ then $\tau(k) \leq f(k) \leq f(n)$, and we have at most $N(f(n))$ numbers with this property, that is, at most $A^f(n) \cdot f(n)^f(n)$ numbers with this property, by the theorem; but now we have, for $n$ large enough,

$$
A^f(n) f(n)^f(n) = \exp(f(n) \log(Af(n))) \leq \exp \left( f(n) \log \left( \frac{\log n}{(\log \log n)^{1 - \frac{\varepsilon}{2}}} \right) \right) = \exp(\log n \left( 1 + (1 - \varepsilon) \frac{\log \log n}{\log \log n} \right) \left( 1 + \frac{1 - \varepsilon}{2} \frac{\log \log n}{\log \log n} \right)) \leq n \exp \left( -\frac{\varepsilon \log n \log \log n}{2 \log \log n} \right).
$$

Therefore $\lim_{n \to \infty} \frac{B(n)}{n} = 0$.

**Remark.** If we have $p$-ary operations instead of binary operations ($p \geq 2$), we have an analogous result by changing $N(k) \leq A^k \cdot k^k$ into $N(k) \leq A^k k^{(p-1)k}$ in the theorem and $f(n)$ into $\frac{f(n)}{p^k}$ in the corollary.
2. The upper estimates

**Theorem 2.** Given $\varepsilon > 0$, we have, for $n$ large enough,

$$\tau(n) \leq \frac{\log n}{\log \log n} + (3 + \varepsilon) \frac{\log n \log \log n}{(\log \log n)^2} =: g(n).$$

**Proof.** Let $B = \left\lfloor \frac{\log n}{(\log \log n)^3} \right\rfloor$, $C = B^k$, where $k = \left\lfloor \log \log n \right\rfloor$ (in the calculations we shall omit the integer parts). Take

$s_0 = 1, s_1 = 2, s_2 = 3, \ldots, s_{B-2} = B - 1,$
$s_{B-1} = B, s_{B+1} = 3B, \ldots, s_{2B-3} = (B - 1)B, \ldots,$
$s_{(k-1)(B-1)} = B^{k-1}, s_{(k-1)(B-1)+1} = 2B^{k-1}, \ldots, s_{k(B-1)-1} = (B - 1)B^{k-1},$
$s_{k(B-1)} = B^k = C.$

Now take the representation of $n$ in basis $C$

$$n = a_0 + a_1 C + \cdots + a_r C^r, \quad 0 \leq a_i < C, \quad r = \left\lfloor \frac{\log n}{\log C} \right\rfloor \sim \frac{\log n}{(\log \log n)^2},$$

and the representations of the $a_i$ in basis $B$:

$$a_i = b_{i1} + b_{i2} B + \cdots + b_{ik} B^{k-1}, \quad 0 \leq b_{ij} < B.$$

Observe now that we have constructed the $b_{ij} B^{j-1}$, so we can construct an $a_i$ doing $k - 1$ sums. Since we have $r + 1$ coefficients $a_i$ we have a total time $(k - 1)(r + 1)$ to generate all the $a_i$’s. But

$$(k - 1)(r + 1) \sim \left( \frac{\log n}{\log C} \right) \left( \frac{\log C}{\log B} \right) + k - r \sim \frac{\log n}{\log B} + k - r.$$

Having generated the $a_j$, we do the following, in $2r$ steps:

$$a_r \rightarrow a_r \cdot C \rightarrow a_r \cdot C + a_{r-1} \rightarrow (a_r \cdot C + a_{r-1}) \cdot C$$
$$\cdots \rightarrow a_r C^r + a_{r-1} C^{r-1} + \cdots + a_1 C + a_0 = n.$$

The total time is less than

$$k \cdot B + \frac{\log n}{\log B} + r \sim \frac{\log n}{(\log \log n)^3} \cdot \log \log n$$
$$+ \frac{\log n}{\log \log n - 3 \log \log \log n} + \frac{\log n}{(\log \log n)^2}$$
$$= \frac{\log n}{\log \log n} + \frac{3 \log n \log \log n}{(\log \log n)^2} + O\left( \frac{\log n}{(\log \log n)^2} \right)$$
$$\leq g(n).$$

**Remark.** Using essentially the same proof, we can replace $g(n)$ by $\frac{g(n)}{p-1}$ if we have the binary product and the $p$-ary sum $s(x_1, x_2, \ldots, x_p) = x_1 + x_2 + \cdots + x_p.$
3. The cost of polynomials

We shall extend the previous results to polynomials in several variables whose coefficients are integers or natural numbers.

Let $P(X_1, X_2, \ldots, X_k)$ be a polynomial. We define $\tau(P)$ as the minimum $m \in \mathbb{N}$ such that there exists a sequence $(s_{-k}, s_{-k+1}, \ldots, s_{-1}, s_0, s_1, \ldots, s_m)$, where $s_{-i} = X_i, i = 1, 2, \ldots, k$; $s_0 = 1$ and $s_m = P$, such that for each $\ell \geq 1$ there are $i, j < \ell$ with $s_\ell = Op(s_i, s_j)$, $Op \in \{+, -, 0\}$.

**Theorem 3.** Let $P$ be a polynomial whose coefficients are natural numbers.

Let $d_i$ be the degree of $P$ in the variable $X_i$, and let

$$h(P) = (1 + d_1)(1 + d_2) \ldots (1 + d_k) \max_{i} \log(1 + a_i),$$

where $a_i$ are the coefficients of $P$. Then, we have the following results:

i) $\tau(P) \geq \frac{h(P)}{\log h(P)}$ for almost all $P$ (in the sense that the ratio of the number of polynomials that satisfy this property to the number of polynomials with $h(P) \leq M$ tends to 1 as $M$ tends to infinity).

ii) For any given $\varepsilon > 0$, $\tau(P) \leq \frac{(1+\varepsilon)h(P)}{\log h(P)}$, provided $h(P)$ is large enough.

**Remark 1.** If we consider polynomials with integer coefficients we still have the above results, provided we define

$$h(P) = (1 + d_1)(1 + d_2) \ldots (1 + d_k) \max_{i} \log(1 + 2|a_i|).$$

**Remark 2.** In the proof below we work with $\log = \log_2$.

**Proof.** i) Given $d_1, d_2, \ldots, d_k$ and $M \in \mathbb{N}$, there are $N = (1 + M)^{(1+d_1)(1+d_2)\ldots(1+d_k)}$ polynomials with degrees bounded by the $d_i$’s and with $\max_i a_i \leq M$ (the $a_i$’s are natural numbers). Since

$$\frac{\log N}{\log \log N} = \frac{h}{\log h},$$

where $h = (1+d_1) \ldots (1+d_k) \log(M+1)$, the result will follow exactly as in Theorem 1, corollary.

ii) Let $B = \left[\frac{h(P)}{\log h(P)}\right]$, $C = B^{[\log h(P)]}$. At least one of the following two cases must occur:

ii.1) $\max_i a_i \geq C^{[\log h(P)]}$. In this case we proceed as in Theorem 2. We construct $1, 2, \ldots, B - 1, B, 2B, \ldots, (B - 1)B, \ldots, (B - 1)B^{[\log h(P)]}, C$ and then construct all the $a_i$’s using basis $C$, writing the coefficients in basis $B$. After this, we obtain the polynomial term by term. The total cost of this is bounded by

$$B^{\log C} \log B + \left(1 + \frac{\log M}{\log B} + \frac{\log M}{\log C} + \frac{\log C}{\log B}\right) D + 2D,$$

where $M = \max_i (1 + a_i)$ and $D = (1 + d_1)(1 + d_2) \ldots (1 + d_k)$, and so is bounded by

$$(1 + \frac{\varepsilon}{2}) \frac{D \log M}{\log B} < (1 + \varepsilon) \frac{h(P)}{\log h(P)},$$

for $h(P)$ sufficiently large.

ii.2) $2^{(1+d_i)} \geq C^{[\log h(P)]}$ for some $i$, say for $i = 1$. Let $M = \max_i (1 + a_i)$. ...
We have two subcases:

i.2.1) \( M \geq 2\sqrt{\log h(P)} \): In this case we construct all the polynomials \( \sum_{j=0}^{r-1} \delta_j X_i^j \), with \( \delta_j \in \{0,1\}, \forall j, \) and \( r = \lceil \log B \rceil + 1 \), forming the set \( \beta \) and then we construct all the polynomials in the set \( \Gamma = \{X_i^{r_i} \cdot P \mid P \in \beta, 0 \leq j \leq \lceil \log h(P) \rceil \} \). Let \( t = X_i^{r_i\log h(P)} \) and

\[
A = \left\{ \sum_{j=0}^{r_i\log h(P)} \delta_j X_i^j, \delta_j \in \{0,1\} \right\}
\]

Then, we can write

\[
P(x) = \sum \tilde{a}(t) \cdot 2^{i_0} t^{i_1} X_2^{i_2} X_3^{i_3} \ldots X_k^{i_k},
\]

(*)

where \( i_0 < \log(M+1); i_1 \leq \frac{d_1}{r \lceil \log h(P) \rceil} \),

\( i_r \leq d_r \) for \( r \geq 2 \) and \( \tilde{a}(t) \in A \). An element of \( A \) can be obtained as the sum of \( \lceil \log h(P) \rceil \) elements of \( \Gamma \), and, since the total number of coefficients \( \tilde{a}(t) \) is bounded by

\[
(1 + \log(M+1)) \left( 1 + \frac{d_1}{r \lceil \log h(P) \rceil} \right) (1 + d_2) \ldots (1 + d_k),
\]

we can obtain all the \( \tilde{a}(t) \)'s and then obtain \( P \) using (*), and we can estimate the total cost as before, obtaining

\[
\tau(P) \leq \frac{(1 + \epsilon)h(P)}{\log h(P)};
\]

for \( h(P) \) sufficiently large.

i.2.2) \( M < 2\sqrt{\log h(P)} \): In this case we construct all the polynomials \( \sum_{j=0}^{r-1} \delta_j X_i^j \), with \( \delta_j \in \{0,1,2,\ldots,M-1\}, \forall j, \) and \( r = \lceil \log M \rceil \), forming the set \( \beta \), and then we construct all the polynomials in the set \( \Gamma = \{X_i^{r_i} \cdot P \mid P \in \beta, 0 \leq j \leq \lceil \log h(P) \rceil \} \). Let \( t = X_i^{r_i\log h(P)} \) and

\[
A = \left\{ \sum_{j=0}^{r_i\log h(P)} \delta_j X_i^j, \delta_j \in \{0,1,2,\ldots,M-1\} \right\}
\]

Then we can write

(*)

\[
P(x) = \sum \tilde{a}(t) \cdot t^{i_1} X_2^{i_2} X_3^{i_3} \ldots X_k^{i_k}, \text{ where } i_1 \leq \frac{d_1}{r \lceil \log h(P) \rceil},
\]

\( i_r \leq d_r \) for \( r \geq 2 \) and \( \tilde{a}(t) \in A \). An element of \( A \) can be obtained as the sum of \( \lceil \log h(P) \rceil \) elements of \( \Gamma \), and, since the total number of coefficients \( \tilde{a}(t) \) is bounded by

\[
(1 + \frac{d_1}{r \lceil \log h(P) \rceil}) (1 + d_2) \ldots (1 + d_k),
\]

we can obtain all the \( \tilde{a}(t) \)'s and then obtain \( P \) using (*), and we can estimate the total cost as before, obtaining

\[
\tau(P) \leq \frac{(1 + \epsilon)h(P)}{\log h(P)};
\]

for \( h(P) \) sufficiently large.
Remark. The case of integer coefficients with \( \max_I |a(I)| = M \) is essentially equivalent to the case of natural numbers with \( \max_I a(I) = 2M \). We can easily see this by adding or subtracting the polynomial

\[
M(1 + X_1 + \cdots + X_i^{d_1}) \cdots (1 + X_k + \cdots + X_i^{d_k}),
\]

which has very small cost.

4. Remarks and open problems

The most important open question related to \( \tau(n) \) is to decide whether there are natural numbers \( m_k \geq 1 \) and a \( C > 0 \) such that \( \tau(m_k \cdot k!) \leq (\log k)^C, \forall k \in \mathbb{N} \). If such a sequence does not exist then the algebraic version of \( NP \neq P \) is true, as is proved in [SS]. About this question we have the following remarks:

- If there is a \( C > 0 \) such that \( \tau(C_{2^{n-1}}) \leq n^C \), then there exists such a sequence \( m_k \), namely \( m_k = (\log k)^C, \forall k \in \mathbb{N} \).

- If we have the operations \( +, -, \cdot \) and \( 2^n \), then there exists such a sequence \( m_k \). Indeed, we can construct a sequence \( M_k \) with \( \tau(M_k) \leq 4k \) satisfying \( n \leq 2^k \); hence \( n|M_k \). It is enough to take \( M_0 = 1, M_{k+1} = 2M_k (2^2M_k - 1) \), because \( n \leq 2^{k+1}, n \text{ odd } \Rightarrow \frac{\varphi(n)}{2} \leq 2^k \Rightarrow \frac{\varphi(n)}{2} | M_k \Rightarrow n|M_{k+1} \) (\( M_{k+1} \) is divisible by sufficiently large powers of 2). Now it is easy to check that \( (2^n)!|M_0 \cdot M_1^2 \cdot M_2^4 \cdots M_n^{2^n} = P_n, \) which satisfies \( \tau(P_n) \leq \frac{5n(n+1)}{2} \).

There is another open question concerning costs with bounded memory. Let \( g \geq 1 \) and define \( \tau_g(n) \) as \( \tau(n) \) with the restriction that for each \( \ell \geq 1, s_\ell = Op(s_i, s_j) \) with \( s_i, s_j \in \{1, s_{\ell-g}, s_{\ell-g+1}, \ldots, s_{\ell-1}\} \). The question is to determine if there are \( g \geq 1 \) and \( C > 0 \) such that \( \tau_g(n) \leq \tau(n)^C \), for every \( n \in \mathbb{N} \). About this we have the following remarks:

- If \( g = 5 \) and we have the operations \( +, -, \cdot, / \) and \( \lfloor \rceil \) (the integer part function), then \( \tau_g(n) \leq \tau(n)^C \) for some \( C \). Indeed, if \( (s_0, s_1, \ldots, s_k) \), \( s_k = n \) is a computing sequence for \( n \), we may keep \( 2s_0 + 4s_1 + 16s_2 + \cdots + 2^7 s_r \) in the position \( P_1 \) of the memory and use the position \( P_2 \) to generate \( 2^{2^7} \). We obtain \( s_j \) doing

\[
\left\lfloor \left( \frac{X_j}{(2^7)^2} \right) \right\rfloor \cdot 2^{2^7} = s_j
\]

(recall that \( s_j \leq 2^{2^7-1} \)), where \( X_j = 2s_0 + 4s_1 + \cdots + 2^7 s_j \). We put \( s_i \) and \( s_j \) on the positions \( P_3 \) and \( P_4 \) and do \( Op(s_i, s_j) = r_{i+1} \). Then we put \( s_{r+1} \) in the position \( P_3 \) and \( X_{r+1} = X_2 + 2^7 s_{r+1} \) in the position \( P_1 \). We proceed so until we obtain \( s_k \). It is easy to see that the total time is polynomially related to \( \tau(n) = k \).

- If \( g = 1 \) and the operations are \( +, -, \cdot \) and \( / \), then \( \tau_g \) is not polynomially related to \( \tau \). Indeed, we can only transform an \( s_i \) into \( s_{i+1} = s_i \pm 1 \), \( s_{i+1} = 2s_i \) or \( s_{i+1} = s_i^2 \). If \( (s_0, s_1, \ldots, s_k) \) is a computing sequence for \( n = s_k \), and \( r \) is the maximum index such that \( s_r = s_{r-1}^2 \), we will have \( s_k = 2^m s_{r-1}^2 + p \), with \( m \geq 0 \),
\[ |p| \leq 2^{k-r} \leq 2^k; \] hence there is \( \ell \in \mathbb{N} \) such that \( |n - \ell^2| \leq 2^k \) or \( |n - 2\ell^2| \leq 2^k \). If
\[
X_m = \frac{1}{2}((3 + 2\sqrt{2})2^m + (3 - 2\sqrt{2})2^m),
\]
\[
Y_m = \frac{1}{2\sqrt{2}}((3 + 2\sqrt{2})2^m - (3 - 2\sqrt{2})2^m), \quad n = X_m^2 + 2^m,
\]
then \( X_m^2 - 2Y_m^2 = 1, X_{m+1} = 2X_m^2 - 1 \), so \( \tau(X_m) \leq 3m \) and \( \tau(X_m^2 + 2^m) \leq 4m + 1 \), but if \( a \in \mathbb{N} \), then
\[
\left| a^2 - (X_m^2 + 2^m) \right| = \min\{2^{2m}, |(X_m + 1)^2 - (X_m^2 + 2^m)|\} = 2^{2m},
\]
and
\[
\left| 2a^2 - (X_m^2 + 2^m) \right| = \left| 2a^2 - 2Y_m^2 - 1 - 2^{2m} \right|
\]
\[
\geq \min\{2^{2m} + 1, |2(Y_m + 1)^2 - Y_m^2 - 1 - 2^{2m}|\} = 2^{2m} + 1 \Rightarrow 2^k \geq 2^m.
\]
Thus \( k \geq 2^m \), so
\[
\tau_1(X_m^2 + 2^m) \geq 2^m >> \tau(X_m^2 + 2^m)^C,
\]
\( \forall C > 0 \) fixed, if \( m \) is large enough.

Notice, however, that \( \tau_1(2^k) = k + 1 \), and \( \tau_1(n) \leq 2 \log_2 n \), for every \( n \in \mathbb{N} \).

**Acknowledgements**

I would like to thank Steve Smale, Mike Shub, Welington de Melo and Benar Svaiter for many helpful conversations and suggestions of questions which are discussed in this article.

**References**


**Instituto de Matematica Pura e Aplicada, Estrada Dona Castorina, 110, 22460-320, Rio de Janeiro, Brasil**

E-mail address: gugu@impa.br