BABY-STEP GIANT-STEP ALGORITHMS FOR NON-UNIFORM DISTRIBUTIONS

SIMON R. BLACKBURN AND EDLYN TESKE

Abstract. The baby-step giant-step algorithm, due to Shanks, may be used to solve the discrete logarithm problem in arbitrary groups. The paper explores a generalisation of this algorithm, where extra baby steps may be computed after carrying out giant steps (thus increasing the giant step size). The paper explores the problem of deciding how many, and when, extra baby steps should be computed so that the expected cost of the generalised algorithm is minimised. When the logarithms are uniformly distributed over an interval of length n, the expected cost of the generalised algorithm is 6% lower than that of Shanks (achieved at the expense of a slightly larger worst case cost). In some situations where logarithms are far from uniformly distributed, any baby-step giant-step algorithm that computes all its baby steps before taking a giant step must have infinite expected cost, but the generalised algorithm has finite expected cost. The results are heuristic, but are supported by evidence from simulations.

1. Introduction

The classic baby-step giant-step algorithm due to Shanks (see, for example, Cohen [1]) makes use of a time-memory trade-off to search an interval of length n for a discrete logarithm using only $O(\sqrt{n})$ operations. In the standard application, we assume that the distribution of answers (logarithms) is uniform over the interval.

This paper considers the situation where the answer need not be taken from a uniform distribution. Our motivation is from recent papers of Stein and Teske [5, 6], that give an algorithm to find the divisor class number of a hyperelliptic function field. Their approach is a essentially a baby-step giant-step algorithm that searches an interval for a discrete logarithm, where the answer is more likely to lie in the centre of the interval than the extremes. Our goal is to design a baby-step giant-step algorithm that will return an integer picked from a given distribution using the smallest expected number of operations. We will consider a one sided distribution, where small integer values are more likely to occur than larger values. This can be easily adapted to two sided distributions such as those considered by Stein and Teske — we will comment briefly on this in the penultimate section of the paper.

The theoretical and experimental data both suggest that the best way of computing baby steps depends on the hazard function of the distribution (see the next section for a definition of this concept). For non-increasing distributions on the non-negative integers, we conclude the following. If the distribution has expected value E and increasing hazard rate, then

Date: January 4, 2000.

1991 *Mathematics Subject Classification.* 11Y16.

Research of the first author supported by an E.P.S.R.C. Advanced Fellowship.

$\sqrt{E}$ baby steps should be computed, and then all the giant steps should be carried out. If, however, the distribution has decreasing hazard rate, extra baby steps should be computed after some giant steps have been taken — see the next section for details.

The arguments in this paper are not rigorous. In particular, we are very cavalier with error terms. Instead, we support our arguments with experimental data.

The remainder of the paper is organised as follows. Section 2 presents the generalisation of the baby-step giant-step algorithm we will consider. Arguments are given as to why this algorithm is optimal. Section 3 lists several natural distributions, and calculates the theoretically optimal algorithm for each of them. Finally, Section 4 presents the simulation results that support the arguments in Section 2.

2. A Baby-Step Giant-Step Algorithm and its Optimisation

This section analyses a baby-step giant-step algorithm that finds an integer s taken from a distribution on the non-negative integers. Rather than computing all the baby steps at the beginning, the algorithm computes extra baby steps (thus increasing the giant step size) after each unsuccessful giant step. The number of baby steps that are computed is controlled by a function b. The question this paper addresses is: For a given distribution, what choice of b minimises the expected number of operations of the algorithm?

Let b be a monotonic increasing positive-integer valued function on the non-negative integers. In the algorithm below, the variable x holds the smallest integer not checked at any stage; the variable y holds the number of baby steps that have been computed.

1. Set $x = 0$ and $y = 0$.
2. Compute $b(x) - y$ extra baby steps, so that a total of $b(x)$ baby steps have been computed. Set $y = b(x)$.
3. Perform a giant step (of size $b(x)$) to scan the interval $[x, x + b(x) - 1]$. If s lies in this interval, output s and stop.
4. Set $x = x + b(x)$. Return to step 2.

If it is known that $s \leq n$ for some integer n, and if b is the constant function at $\lceil \sqrt{n}\, \rceil$, we have the original algorithm of Shanks. If b is any function such that $b(2(i+1)+\frac{1}{2}i(i-1)) = i+2$ for all $i \in \{0, 1, 2, \dots\}$, we have the algorithm recently proposed by Terr [7].

In order to choose a sensible function b for a particular application, we restrict the distributions we consider to those that are non-increasing. More precisely (and to fix notation), let $p_0, p_1, \dots$ be non-negative real numbers such that $\sum_{i=0}^{\infty} p_i = 1$. Assume that $p_i \geq p_{i+1}$ for all i. We consider the distribution associated with the probabilities p_i. Let X be a random variable taking value i with probability p_i, and let E be the expected value of X.

We now define a function b that comes close to minimising the expected number of operations required by the algorithm above. (We note that most of the information encoded in the function b is not used. For example, none of the values $b(1), b(2), \dots, b(b(0) - 1)$ affect the algorithm. We assume that these unused values of b are chosen so as to make b a 'natural' monotonically increasing function that interpolates the points that matter.) Define a function m, from the non-negative integers to the reals by

$$m(k) = \sqrt{\frac{\sum_{i=k+1}^{\infty} p_i}{p_k}}\ . \tag{1}$$

If m is monotonic increasing, let b be a function such that $b(k)$ is an integer as close as possible to $m(k)$, subject to $b(k) \geq 1$. If m is monotonic decreasing, let b be a constant function equal to an integer which is approximately $\sqrt{E}$. If m is not monotonic, then a good choice for b consists of segments that approximate m (since b is monotonic increasing, m must be monotonic increasing on these segments), together with horizontal lines that join these segments. (In this last circumstance, there does not seem to be a simple rule to determine b in general. However, given a distribution, it is not difficult to search for an optimal value of b from functions of this form.)

This choice of the function b is partly justified by simulations; see the final section of the paper. The rest of this section attempts to give a conceptual justification for this choice of b, by analysing a second algorithm. This second algorithm would not be used in practice. However, its expected cost is easier to analyse, and simulations show that the two algorithms have comparable costs. Before carrying out this analysis, we digress to discuss the relationship between the choice of b given above and a concept in statistics known as the hazard rate.

If the distribution does not decay too rapidly, the ratio inside the square root in (1) is well approximated by $(\sum_{i=k}^{\infty} p_i)/p_k$. This quantity is known as the Mills' ratio, and its reciprocal is known as the hazard rate (or failure rate) at k; see Johnson and Kotz [2, p.111]. If X is the random variable corresponding to the probabilities p_i, the hazard rate at k measures the likelihood of X lying in a small interval beyond k, given that $X \geq k$. (It is used in contexts where the random variable models the likelihood of some machine component failing within a time interval.) We then have an intuitive explanation for the form of the function b above. For if the hazard ratio is high at a point x, and we know that $s \notin [0, x-1]$, then it is likely that s will occur soon, and so few baby steps should be computed. If the hazard ratio drops as x increases, more baby steps should therefore by computed at each stage, and so b should be an increasing function. Distributions of this type are known as Decreasing Hazard Rate (DHR) distributions; they tend to be distributions with large tails. If the hazard rate increases (so the distribution is an Increasing Hazard Rate (IHR) distribution), the optimal number of baby steps at a point drops. But in this case, since baby step computations cannot be taken back once they are computed, the best we can do is to compute no extra baby steps, and so b is a constant function. This argument gives some reasons why we might expect b to depend on the hazard rate, but does not reveal the precise form of b. We now give an argument that indicates this more precise form.

We briefly describe our second algorithm. Let h be a positive integer. Divide the non-negative integers into intervals $I_0, I_1, \ldots$ of length h, where $I_j = [jh, (j+1)h-1]$. Define $P_j = \sum_{i=jh}^{(j+1)h-1} p_i$; so P_j is the probability that s lies in I_j. To scan each interval I_j, the algorithm computes some baby steps, increasing the giant step size to $b(jh)$ steps. The algorithm then uses $\lceil h/b(jh) \rceil$ giant steps to test whether $s \in I_j$.

We may regard this second algorithm as an approximation to the first, where the number of baby steps is only updated when h integers have been scanned since the last update. For all the approximations we will make to be reasonable, h should be of a moderate size, slightly larger than the square root of the expected value of a random variable taking the value k with probability p_k, say.

The expected number of operations needed to find s using this second algorithm is approximately

$$\sum_{j=0}^{\infty} P_j \left(b(jh) + \left\lceil \frac{h}{b(0)} \right\rceil + \left\lceil \frac{h}{b(h)} \right\rceil + \cdots + \left\lceil \frac{h}{b((j-1)h)} \right\rceil \right) .$$

This is a lower bound for the expected number of operations. An upper bound is the same expression with a term $\lceil h/b(jh) \rceil$ added to the sum. A more precise estimate would replace this term with $(E(X|X \in I_j) - jh)/b(jh)$, where the random variable X is such that $\Pr(X = i) = p_i$.

Rearranging this sum, and ignoring the rounding errors caused by requiring the number of giant steps on each interval to be integers, we find that the total expected cost of the algorithm is

$$\sum_{k=0}^{\infty} \left(P_k b(kh) + \frac{h \sum_{j=k+1}^{\infty} P_j}{b(kh)} \right) . \tag{2}$$

If we wish to minimise the expected cost of the algorithm, we need to choose the function b so as to minimise the expression (2). Define $m(kh)$ to be that value of $b(kh)$ that minimises the k-th term of (2). So m is a function from the set of non-negative multiples of h to the non-negative integers, and

$$m(kh) = \sqrt{\frac{h}{P_k} \left(\sum_{j=k+1}^{\infty} P_j \right)} , \tag{3}$$

if this expression is an integer, or is one of the two integers closest to this value otherwise. (Note that if $P_k = 0$ then this value is not well defined; but in this case the value of $b(kh)$ does not affect the running time of the algorithm. So when $P_k = 0$, we set $m(kh) = m((k-1)h)$ without loss of generality.)

If the function m defined by (3) is monotonic increasing, setting $b(kh) = m(kh)$ for all k minimises the expected cost of the algorithm. But for many distributions, m is not monotonic increasing and so we cannot use $m(kh)$ to define $b(kh)$ at all values of kh. However, if b is a function that minimises the cost of the algorithm it is not difficult to see that either $b(kh) = m(kh)$, or $b(kh) > m(kh)$ and $b(kh) = b((k-1)h)$, or $b(kh) < m(kh)$ and $b(kh) = b((k+1)h)$. Thus a minimal function b is made up of horizontal line segments, together with segments of the function m defined by (3). In particular, when m is monotonic decreasing (which is the case for a great many sensible probability distributions), the minimising function b is a constant function. Interpreted in algorithmic terms, this means that all the baby steps should be computed at the beginning, before any giant steps are carried out — this is the approach of the original algorithm of Shanks. When b is constant, the value of b giving minimal expected cost is about $\sqrt{E(X)}$, where X is the random variable such that $\Pr(X=i)=p_i$ for all i. To see this, when $b(k)=b$ for some integer b we may write the cost function (2) as

$$\sum_{k=0}^{\infty}\left(P_k b+\frac{h}{b}\left(P_k+P_{k+1}+\cdots\right)\right)=b+\sum_{k=0}^{\infty}\frac{hk}{b}P_k\ .$$

This last sum is approximately $E(X)/b$ provided that h is small enough, since the event that s lies in I_k occurs with probability P_k and this event occurring means that s is approximately hk. Since the minimum value of $b+E(X)/b$ occurs when $b=\sqrt{E(X)}$, choosing b to be this value should minimise the expected cost.

3. Example Distributions

This section considers various distributions, and determines a suitable strategy for computing baby steps in each case.

3.1. **The uniform distribution.** Suppose s is uniformly distributed throughout the interval $[0,n-1]$, so

$$p_i=\begin{cases}\frac{1}{n} & \text{if } i<n\ ,\\ 0 & \text{otherwise}\ .\end{cases}$$

This is the situation most often considered. In this case, $m(k)$ is approximately $\sqrt{n-k}$ when $k<n$, and $m(k)=0$ when $k\geq n$. Since m is monotonic decreasing, the optimal choice for b is a constant function approximately equal to $\sqrt{n/2}$. So the original approach of Shanks, computing all the baby steps first, is the best possible here. (However, Shanks' version of computing approximately $\sqrt{n}$ baby steps yields a worse average-case running time but a better worst-case running time.)

3.2. **The one sided normal distribution.** Suppose that s follows a discrete approximation to one side of a normal distribution of mean 0 and standard deviation σ. So

$$p_i\simeq\frac{2}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}i^2}\ .$$

The normal distribution is an IHR distribution (see, for example, Kececioglu [3, p.362], and so the optimal value for b should be a constant function at $\sqrt{\sqrt{2}\sigma/\sqrt{\pi}}$.

3.3. **The discrete Pareto distribution.** Let $p_i=c/(i+1)^d$, where $d>1$ and where c is chosen so that $\sum_{i=0}^{\infty}p_i=1$, i.e. $c=1/\zeta(d)$, where $\zeta(\cdot)$ denotes the Riemann zeta function. (This distribution is also referred to as the Zipf distribution, or the Riemann zeta distribution.)

Then the function $m(k)$ defined by (1) is approximately $\sqrt{k/(d-1)}$. Since this function is monotonic increasing, we set $b(k)=\max\{1,[m(k)]\}$ in this case. For this choice of $b(k)$, our algorithm becomes similar to an algorithm recently proposed by Terr [7]. Note that when $d\leq 2$, an algorithm that computes all its baby steps at the beginning has an infinite expected cost, while for the algorithm with $b(k) = m(k)$ has infinite expected cost only when $1 < d < 3/2$.

3.4. **The Weibull distribution.** The discrete Weibull distribution is approximately given by

$$p_i = \frac{\beta}{\eta}\left(\frac{i}{\eta}\right)^{\beta-1} e^{-(i/\eta)^\beta} ,$$

where the positive real numbers β and η are respectively the shape parameter and the scale parameter. We only consider the case when $0 < \beta \leq 1$ — the case when $\beta = 1$ is the geometric distribution (see the next subsection) and when $\beta > 1$, the distribution is not decreasing (the p_i rapidly climb to a maximum value, before falling gently towards zero).

The hazard function of the distribution is approximately $\lambda(k) = \beta/\eta \cdot (k/\eta)^{\beta-1}$, which is strictly decreasing for $0 < \beta < 1$. The expected value $E(X)$ is approximately $\eta\,\Gamma(1/\beta + 1)$, where Γ denotes the gamma function. Now, $m(k)$ is approximately $1/\sqrt{\lambda(k)}$ so we set $b(k) = \max\{1, m(k)\}$ for $k \geq 1$, and $b(0) = b(1)$.

3.5. **The geometric distribution.** Let $p_i = qp^i$, where $p + q = 1$. The function m defined by (1) is approximately $\sqrt{p/q}$, a constant function. Note that $E(X) = p/q$ in this case, and so the value for b obtained by using the function m is equal to the value of b that is obtained by minimising the expected cost over all constant functions.

3.6. **The split uniform distribution.** Suppose that the interval $[0, n-1]$ is divided into two parts $[0, \ell - 1]$ and $[\ell, n-1]$ in each of which s is uniformly distributed, i.e.

$$p_i = \begin{cases} p_A & \text{if } i < \ell\ , \\ p_B & \text{if } \ell \leq i < n\ , \\ 0 & \text{otherwise}\ , \end{cases}$$

where p_A, p_B and ℓ are such that $p_A \geq p_B$ and $\ell p_A + (n-\ell)p_B = 1$, i.e., $p_B = (1-\ell p_A)/(n-\ell)$. Then $m(k) = \sqrt{1/p_A - (k+1)}$ if $k < \ell$ and $m(k) = \sqrt{n-(k+1)}$ if $\ell \leq k < n$, so that m is neither decreasing nor increasing. Since m is decreasing on the intervals $[0, \ell - 1]$ and $[\ell, n-1]$, the function b should be constant on these two intervals. Thus b is determined by $b(0)$ and $b(\ell)$. So a good choice of b is computed by minimising the expected cost over all choices of $b(0)$ and $b(\ell)$; in other words, by minimising

$$\left(b(0) + \frac{\ell}{2b(0)}\right)\ell p_A + \left(b(\ell) + \frac{\ell}{b(0)} + \frac{n-\ell}{2b(\ell)}\right)(n-\ell)p_B\ .$$

Hence, the best values for $b(0)$ and $b(\ell)$ should be approximately

$$b(0) = \sqrt{\frac{1}{p_A}\left(\frac{1}{2}\ell p_A + (n-\ell)p_B\right)} \text{ and } b(\ell) = \sqrt{\frac{1}{2}(n-\ell)}\ ,$$

which yields an increasing function if $p_A > 2/n$. If $p_A \leq 2/n$, the cost function is minimised by a constant function, at $\sqrt{E(X)}$, where

$$E(X) = \frac{\ell^2}{2} p_A + \frac{n^2 - \ell^2}{2} p_B \ .$$

3.7. **Two sided distributions.** The distributions we have considered have all been non-increasing, starting at 0. We may easily modify our treatment to deal with two sided distributions such as discrete approximations to the normal distribution. In this case, our algorithm should start at the peak value of the distribution, making giant steps in both directions away from the mean. In the case of a symmetrical distribution about 0, the cost function may be approximated by

$$\sum_{k=0}^{\infty} P_k b(kh) + \frac{2h \sum_{j=k+1}^{\infty} P_j}{b(kh)} \ . \tag{4}$$

where here P_k is the probability that $s \in I_k \cup -I_k$. This changed cost function leads to an altered function $m(k)$, that differs by a factor of $\sqrt{2}$ from the original. The optimal value for a constant function b is equal to $\sqrt{2E(|X|)}$.

To take a specific example, if s is approximately normally distributed with mean 0 and standard deviation σ, we find that the optimal choice of b is a constant function at $\sqrt{2\sqrt{2}\sigma/\sqrt{\pi}}$.

4. Experimental Results

We presented two algorithms in Section 2. The first algorithm, the 'practical' algorithm, has a running time that is difficult to analyse precisely. The second algorithm, the 'theoretical' algorithm, was used to derive an optimal choice for the baby-step function b. It is important to verify experimentally that the running times of the two algorithms are comparable when a theoretically optimal choice of b is used, and that the optimal value for b predicted by the theory accords well with practice. This section contains the experimental results we have obtained.

Subsection 4.1 gives the mean cost of the practical and theoretical algorithms, when s is taken from some of the distributions considered in Section 3. Their costs are found to be comparable. Baby step functions corresponding to the algorithms of Shanks and Terr are also included for the purpose of comparison.

Subsection 4.2 attempts to find the best choice of the function b experimentally. The optimal choice for b via experiment is shown to agree well with the choice predicted by the theory.

For all our experiments, we used the computer algebra system LiDIA [4].

4.1. **Testing the theoretically optimal baby-step functions.** This subsection compares the mean costs of the practical and theoretical algorithms on a range of distributions, where the baby step function b is taken to be the theoretically best choice. Costs of the algorithms of Shanks and of Terr are also included, by considering appropriate baby step functions b.

To this end, we conducted the following experiment. We considered approximately N values of s, taken from various distributions $p_0, p_1, \dots, p_{n-1}$ on $[0, n-1]$. In our experiment, we took $n = 100000$ and $N = 2 \cdot 10^9$. For each distribution, we carried out the following steps:

1. *Simulate the probability distribution on* $[0, n-1]$*:*
   For $0 \le i < n$, let s_i be the nearest integer to $p_i N$. Let $\hat{N} = \sum_{i=0}^{n-1} s_i$, and let $\hat{p}_i = s_i/\hat{N}$. Compute the expected value $E(s) = \sum_{i=0}^{n-1} i\hat{p}_i$.
2. *Compute the theoretically optimal baby-step function:*
   If the distribution is IHR, let $b_{\mathrm{opt}}(0)$ be the closest integer to $\sqrt{E(s)}$, and let $b_{\mathrm{opt}}(k) = b_{\mathrm{opt}}(k-1)$ for $1 \le k < n$.
   If the distribution is DHR, use (1) with p_i, p_k replaced by $\hat{p}_i, \hat{p}_k$ to find $m(k)$ $(0 \le k < n)$ and let $b_{\mathrm{opt}}(k)$ be the integer closest to $m(k)$ subject to the conditions that $b_{\mathrm{opt}}(k) \ge 1$ and that b_{opt} is not decreasing. In the case of the split uniform distribution, choose the function b as given in Subsection 3.6. For the explicit baby-step functions see further below.
3. *Compute Shanks-type baby-step functions:*
   Define $b_{\mathrm{S,n}}(k) = \lceil \sqrt{n} \rceil$ for $0 \le k < n$. This is the baby-step function which is canonically used in baby-step giant-step algorithms. This function does not depend on the probability distribution.
   If the distribution is not IHR, we also define the function $b_{\mathrm{S,E}}$ as a constant function at $\sqrt{E(s)}$. This represents the best choice of baby-step function, if one is restricted to computing all the baby steps before a giant step is carried out; if the distribution is IHR, this function is identical with b_{opt}.
4. *Compute Terr's baby-step function:*
   Let $b_{\mathrm{T}}(k) = 2$ for $0 \le k < 4$ and, for $k \ge 4$, let $b_{\mathrm{T}}(k) = j + 2$ where $j = j(k)$ is the uniquely determined integer such that $2(j+1) + \frac{1}{2}j(j-1) \le k < 2(j+2) + \frac{1}{2}j(j+1))$.The function $b_{\mathrm{T}}(k)$ essentially grows as $\sqrt{2k}$. For example, we have the following values:

k	0	4	7	11	16	22	...	106	121	...	1036	1082	...	10012	10154	...	99682
$b_{\mathrm{T}}(k)$	2	3	4	5	6	7	...	15	16	...	46	47	...	142	143	...	447

   The practical algorithm with this choice of baby-step function is essentially identical to the algorithm of Terr.
5. *For each baby-step function above, determine the mean cost of the practical algorithm:*
   For each $i \in [0, n-1]$ with $s_i \neq 0$ we count the numbers of baby steps and giant steps needed to scan the interval $[0, i]$ using the practical algorithm, and multiply the respective results by s_i. We add these results up for all i and divide the respective sums by $\hat{N}$. This gives the average numbers of baby steps and giant steps. The mean cost is just the sum of these two numbers.
6. *For each baby-step function above, determine the mean cost of the theoretical algorithm:*
   Let h be the integer nearest to $\sqrt{E(s)}$. Then we proceed analogously to the practical algorithm. Notice that, however, we do this only for Terr's baby-step function and, if the distribution is not IHR, for the theoretically optimal function; for all Shanks-type

functions, the theoretical and the practical algorithms are identical.

The explicit probability distributions and their theoretically optimal baby-step functions.

Recall that $n = 100000$ in all our experiments.

The *uniform distribution* on $[0, n-1]$ has expected value $n/2$, so the theoretically optimal baby-step function is given by $b_{\mathrm{opt}}(k) = 224$ for all k.

When considering the *one-sided normal distribution*, we work with $\sigma = 25000$. On $[0, n-1]$ we find that $E(s) = 19941.4$, so the theoretically optimal baby-step function is given by $b_{\mathrm{opt}}(k) = 141$ for all k.

For the *Pareto distribution*, we choose $d = 1.2$, which yields $E(s) = 2453.9$. The theoretically optimal function $b_{\mathrm{opt}}(k)$ looks as follows:

k	0	2	3	5	8	...	107	119	...	1006	1048	...	10036	10263	...	19546	...	39172
$b(k)$	3	4	5	6	7	...	21	22	...	56	57	...	137	138	...	166	...	184

Here, b_{opt} is a step function, and we always tabulate the least value k for which b_{opt} assumes the indicated value. In particular, for all $39172 \le k < n$ we have $b_{\mathrm{opt}}(k) = 184$.

For the *Weibull distribution*, we choose $\beta = 0.5$ and $\eta = 10000$. We find that $E(s) = 12882.4$, so the theoretically optimal function $b_{\mathrm{opt}}(k)$ is a step function that takes the following values (tabulating as before):

k	0	2	3	4	5	6	...	104	113	...	1045	1100	...	10340	10702	...	35884
$b(k)$	14	17	19	20	21	22	...	45	46	...	79	80	...	135	136	...	166

.

(In particular, $b_{\mathrm{opt}}(k) = 166$ when $35884 \le k < n$.)

For the *geometric distribution*, we work with $p = 0.9999$, and we find that $E(s) = 9994.5$. We calculate that the theoretically optimal baby-step function is defined by $b_{\mathrm{opt}}(k) = 100$ for all k.

Finally, in the case of the split uniform distribution, we work with $\ell = 20000$, and $p_A = 1/25000$. Then $E(s) = 19999.5$ and the optimal baby-step function is such that $b_{\mathrm{opt}}(k) = 122$ for $0 \le k < 20000$ and $b_{\mathrm{opt}}(k) = 200$ for $20000 \le k < n$.

Notice that in all distributions above, the parameters have been chosen so that $\max\{i\,;\, s_i \neq 0\} = n - 1$.

The results for this experiment are shown in Table 1.

We see that in all cases, we get the best performance for both the practical and the theoretical algorithm if we use b_{opt}. In particular, the differences between the data for b_{opt} and for $b_{S,317}$ are quite impressive, especially in the case of the Pareto, the Weibull, the geometric and the one-sided normal distributions. While $b_{S,E}$ still yields an acceptable or good performance, this is not the case for the Pareto distribution, where the use of a non-constant function b seems to be particularly practical.

Although the theoretical and practical algorithms have comparable costs for a theoretically optimal choice of b, we notice a discrepancy between the performances of other baby-step functions. This phenomenon is particularly striking in the case of Terr's function, and is

	b	aver. # bs	aver. # gs	aver. # (bs+gs)
Uniform distribution: $E(s) = 49999.5$				
Pr. = Th.	optimal	224	224.712	448.712
Pr. = Th.	Shanks, 317	317	159.225	476.225
Pract.	Terr	299.13	299.138	598.269
Theor. ($h = 224$)	Terr	297.908	396.789	694.698
One sided normal distribution: $\sigma = 25000$, $E(s) = 19941.4$				
Pr. = Th.	optimal	141	142.92	283.92
Pr. = Th.	Shanks, 317	317	64.395	381.395
Pract.	Terr	184.805	184.812	369.617
Theor. ($h = 141$)	Terr	183.511	244.451	427.962
Pareto distribution: $d = 1.2$, $E(s) = 2453.9$				
Pract.	optimal	27.363	25.675	53.038
Theor. ($h = 50$)	optimal	25.441	31.739	57.18
Pr. = Th.	Shanks, 50	50	50.164	100.164
Pr. = Th.	Shanks, 317	317	8.777	325.777
Pract.	Terr	29.016	28.858	57.875
Theor. ($h = 50$)	Terr	27.371	36.198	63.569
Weibull distribution: $\beta = 0.5$, $\eta = 10000$, $E(s) = 12882.4$				
Pract.	optimal	107.772	104.297	212.069
Theor. ($h = 114$)	optimal	105.769	111.365	217.135
Pr. = Th.	Shanks, 114	114	114.427	228.427
Pr. = Th.	Shanks, 317	317	42.009	359.009
Pract.	Terr	123.536	123.56	247.096
Theor. ($h = 114$)	Terr	121.468	166.963	288.432
Geometric distribution: $\sigma = 0.9999$, $E(s) = 9994.5$				
Pr. = Th.	optimal	100	101.435	201.435
Pr. = Th.	Shanks, 317	317	32.999	349.999
Pract.	Terr	126.283	126.295	252.579
Theor. ($h = 100$)	Terr	124.918	166.92	291.838
Split uniform distribution: $\ell = 20000$, $p_A = 1/25000$; $E(s) = 19999.5$				
Pract.	optimal	137.952	139.861	277.813
Theor. ($h = 141$)	optimal	137.598	139.939	277.537
Pr. = Th.	Shanks, 141	141	143.338	284.338
Pr. = Th.	Shanks, 317	317	64.579	381.579
Pract.	Terr	175.519	175.528	351.047
Theor. ($h = 141$)	Terr	174.181	235.003	409.184

TABLE 1. Testing the b's. $n = 100000$.

can be explained as follows: To search the interval $[0, h-1]$, we use giant steps of length $b_T(0) = 2$ in the theoretical algorithm. Thus, to find x in $[0, h-1]$, we use two baby steps and $x/2$ giant steps. To search the whole interval $[0, h-1]$, we use two baby steps and $(h-1)/2$ giant steps of length two. Only after that, the giant step size is increased to $b_T(h)$, etcetc. This leads to a considerably increased number of giant steps, compared with the practical algorithm where the baby step size is adjusted after each giant step. The effect of this can be reduced if we decrease the size of h, as the results in Table 2 show.

The shortcut. If equality checks are cheap, what is the case in the majority of applications, the baby-step giant-step method usually is applied with a "shortcut": Instead of computing all $b(0)$ baby steps first and then using the first giant step to check whether $i \in [0, b(0)-1]$, one checks for each newly computed baby step whether i has been found already. This means that one finds all $i < b(0)$ by performing just $i+1$ baby steps and equality checks, and no giant step. This technique is particularly favourable for all Shanks-type functions. To check whether our theoretically optimal baby-step functions still give the best performances, we have implemented the shortcut in both the theoretical and practical algorithms. The results are shown in Table 3, where we see that indeed, the performance of $b_{S,E}$ and $b_{S,317}$ improves, where the most dramatic improvement we observe in the case of the Pareto distribution. Nevertheless, the theoretically optimal baby-step functions still yield the best performances, both for the theoretical and the practical algorithms.

4.2. **Experimental finding of best baby-step functions.** In this section, we try to find best baby-step functions b experimentally. For this, we work with the same probability distributions as above, and we restrict ourselves to monotonically increasing step functions b that are constant on the five intervals $[j \cdot 20000, (j+1) \cdot 20000[$, $0 \le j \le 4$. We use the practical algorithm with the shortcut; by doing this, we risk some discrepancy with our theoretically optimal function, but this is the algorithm that would be used in practice, after all.

For each distribution, we proceed in several rounds, where with each round we converge on a good step function. In each round, we allow seven different values for the steps, which gives 462 distinct increasing step functions with five steps, and we keep track of those 20 combinations of values which yield the 20 lowest average costs. (The costs are computed in the same way as in the previous section.) In the beginning the seven values are uniformly spread over the interval $[0, \max\{\sqrt{n}, 2\sqrt{E(s)}\}]$, and based upon the best values of the previous round, we choose the seven values for the next round, so that eventually we end up with what we hope are the best choice of values. A selection of our results – the best 5 choices each together with the respective average costs – is shown in Table 4. There, an arrow marks the theoretically optimal function; since the optimal functions for the Pareto and the Weibull distributions can not be written in that form, we simply copied the corresponding performance data for the optimal functions from Table 3.

We see that in the case of the one-sided normal and the geometric distributions, the best step function and the theoretically optimal function (almost) coincide. In the case of the uniform and the split uniform distributions, these two function differ slightly. However,

	b	aver. # bs	aver. # gs	aver. # (bs+gs)
Uniform distribution: $E(s) = 49999.5$				
Pract.	Terr	299.13	299.138	598.269
Theor. $(h = 224)$	Terr	297.908	396.789	694.698
Theor. $(h = 100)$	Terr	297.812	340.653	638.465
Theor. $(h = 50)$	Terr	297.836	318.762	616.598
Theor. $(h = 20)$	Terr	297.924	306.895	604.819
One sided normal distribution: $\sigma = 25000$, $E(s) = 19941.4$				
Pract.	Terr	184.805	184.812	369.617
Theor. $(h = 141)$	Terr	183.511	244.451	427.962
Theor. $(h = 100)$	Terr	183.469	225.983	409.453
Theor. $(h = 50)$	Terr	183.428	204.192	387.621
Theor. $(h = 20)$	Terr	183.505	192.415	375.92
Pareto distribution: $d = 1.2$, $E(s) = 2453.9$				
Pract.	optimal	27.363	25.675	53.038
Theor. $(h = 50)$	optimal	25.441	31.739	57.18
Theor. $(h = 20)$	optimal	26.396	28.75	55.146
Pract.	Terr	29.016	28.858	57.875
Theor. $(h = 50)$	Terr	27.371	36.198	63.569
Theor. $(h = 20)$	Terr	27.955	31.885	59.84
Weibull distribution: $\beta = 0.5$, $\eta = 10000$, $E(s) = 12882.4$				
Pract.	Terr	123.536	123.56	247.096
Theor. $(h = 114)$	Terr	121.468	166.963	288.432
Theor. $(h = 100)$	Terr	121.597	161.398	282.995
Theor. $(h = 50)$	Terr	122.011	141.49	263.501
Theor. $(h = 20)$	Terr	122.221	130.488	252.71
Geometric distribution: $\sigma = 0.9999$, $E(s) = 9994.5$				
Pract.	Terr	126.283	126.295	252.579
Theor. $(h = 100)$	Terr	124.918	166.92	291.838
Theor. $(h = 50)$	Terr	124.925	145.333	270.259
Theor. $(h = 20)$	Terr	124.931	133.676	258.607
Split uniform distribution: $\ell = 20000$, $p_A = 1/25000$; $E(s) = 19999.5$				
Pract.	Terr	175.519	175.528	351.047
Theor. $(h = 141)$	Terr	174.181	235.003	409.184
Theor. $(h = 100)$	Terr	174.224	216.587	390.812
Theor. $(h = 50)$	Terr	174.127	194.828	368.955
Theor. $(h = 20)$	Terr	174.189	183.076	357.265

TABLE 2. Theoretical versus practical algorithm: varying h.

	b	aver. # bs	aver. # gs	aver. # (bs+gs)
Uniform distribution: $E(s) = 49999.5$				
Pr. = Th.	optimal	223.75	223.712	447.462
Pr. = Th.	Shanks, 317	316.499	158.225	474.725
Pract.	Terr	299.13	298.138	597.269
Theor. ($h = 224$)	Terr	297.908	395.789	693.698
One sided normal distribution: $\sigma = 25000$, $E(s) = 19941.4$				
Pr. = Th.	optimal	140.678	141.92	282.599
Pr. = Th.	Shanks, 317	315.386	63.395	378.782
Pract.	Terr	184.805	183.812	368.617
Theor. ($h = 141$)	Terr	183.511	243.451	426.962
Pareto distribution: $d = 1.2$, $E(s) = 2453.9$				
Pract.	optimal	27.017	25.881	52.898
Theor. ($h = 50$)	optimal	24.963	30.739	55.702
Pr. = Th.	Shanks, 50	23.013	49.164	72.177
Pr. = Th.	Shanks, 317	91.717	7.777	99.494
Pract.	Terr	28.82	27.858	56.678
Theor. ($h = 50$)	Terr	27.175	35.198	62.373
Weibull distribution: $\beta = 0.5$, $\eta = 10000$, $E(s) = 12882.4$				
Pract.	optimal	107.756	105.219	212.975
Theor. ($h = 114$)	optimal	105.544	110.365	215.909
Pr. = Th.	Shanks, 114	106.764	113.427	220.191
Pr. = Th.	Shanks, 317	282.521	41.009	323.53
Pract.	Terr	123.536	122.56	246.096
Theor. ($h = 114$)	Terr	121.468	165.963	287.432
Geometric distribution: $\sigma = 0.9999$, $E(s) = 9994.5$				
Pr. = Th.	optimal	99.502	100.435	199.937
Pr. = Th.	Shanks, 317	312.029	31.999	344.028
Pract.	Terr	126.283	125.295	251.579
Theor. ($h = 100$)	Terr	124.918	165.92	290.838
Split uniform distribution: $\ell = 20000$, $p_A = 1/25000$; $E(s) = 19999.5$				
Pract.	optimal	137.657	138.861	276.518
Theor. ($h = 141$)	optimal	137.303	138.939	276.242
Pr. = Th.	Shanks, 141	140.605	142.338	282.943
Pr. = Th.	Shanks, 317	314.996	63.579	378.575
Pract.	Terr	175.518	174.528	350.047
Theor. ($h = 141$)	Terr	174.181	234.003	408.184

TABLE 3. Testing the b's. $n = 100000$. With shortcut

Distribution	$b(0)$	$b(20000)$	$b(40000)$	$b(60000)$	$b(80000)$	aver. #(bs+gs)
Uniform	227	228	228	228	228	447.179
	226	227	227	227	227	447.189
	225	226	226	226	226	447.198
	231	231	231	231	231	447.204
	224	225	225	225	225	447.205
	⋮	⋮	⋮	⋮	⋮	⋮
$\rightarrow$	224	224	224	224	224	447.249
One-s. normal	142	142	142	142	142	282.611
$\rightarrow$	141	141	141	141	141	282.612
	142	142	142	142	143	282.612
	141	141	141	141	142	282.613
	142	142	142	142	144	282.613
Pareto $\rightarrow$						52.898
	71	167	167	167	167	59.910
	71	166	166	166	166	59.910
	71	165	165	165	165	59.911
	71	164	164	164	164	59.912
	72	167	167	167	167	59.912
Weibull $\rightarrow$						212.975
	106	155	155	155	155	216.734
	107	156	156	156	156	216.738
	107	153	153	153	153	216.740
	106	156	156	156	156	216.741
	107	154	155	155	155	216.742
Geometric $\rightarrow$	100	100	100	100	100	199.946
	100	100	100	100	102	199.947
	100	100	100	101	101	199.947
	100	100	100	100	103	199.947
	100	100	100	101	102	199.947
Split uniform	124	194	194	194	194	276.265
	124	195	195	195	195	276.265
	124	196	196	196	196	276.265
	124	197	197	197	197	276.266
	124	198	198	198	198	276.267
	⋮	⋮	⋮	⋮	⋮	⋮
$\rightarrow$	122	200	200	200	200	276.422

TABLE 4. Optimal step functions (5 steps). With shortcut.

the difference in the average costs for the best step function and the theoretically optimal function is very small! Even in the case of the Pareto and the Weibull distributions, where the theoretically optimal function is much more complex, the best step functions yield average costs that are only slightly higher. Finally, it may be interesting to note that for the Pareto distribution from Section 4.1 we have $b_{\text{opt}}(0) = 3$, $b_{\text{opt}}(20000) = 166$ and $b_{\text{opt}}(40000) = b_{\text{opt}}(60000) = b_{\text{opt}}(80000) = 184$, while for the Weibull distribution we have $b_{\text{opt}}(0) = 14$, $b_{\text{opt}}(20000) = 153$ and $b_{\text{opt}}(40000) = b_{\text{opt}}(60000) = b_{\text{opt}}(80000) = 166$.

5. Conclusion

We have examined, in theory and practice, the question of which strategy of computing the baby step set in the baby-step giant-step method yields the best average performance, given the probability distribution of the solution. Our results can be used to considerably speed up the baby-step giant-step method, and to save storage space as well.

In particular, our findings show that the original method of computing $\sqrt{n}$ baby steps at the very beginning, and no other baby steps later (where n is the length of the interval in which the solution lies) is not optimal for any of the standard distributions, even the uniform distribution. Here, optimality is measured in terms of minimising the expected number of operations. Indeed, if the probability distribution of the solution has increasing hazard rate (which is the case for the uniform distribution, the one-sided normal distribution and the geometric distribution) and expected value E, the best average performance is obtained when about $\sqrt{E}$ baby steps are computed first, and then giant steps are computed until the solution is found. In particular, this means that for IHR distributions, the only information needed to execute the optimal strategy is the expected value. On the other hand, if the distribution has decreasing hazard rate (such as the Pareto distribution or the Weibull distribution), the optimal strategy consists in computing some baby steps, then some giant steps, then some more baby steps and so on, where the exact optimal number of steps depends on the whole sequence (p_i) of probabilities. However, if this information is not available, applying the same strategy as for IHR distributions still gives much better results than the original version. Finally, it is worth noting that Terr's algorithm is close to optimal for the Pareto distribution.

Acknowledgments. Many thanks to Sean Murphy, Andreas Stein and Albyn Jones for very useful comments during the writing of this paper.

References

[1] H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer, Berlin, 1993.
[2] N.L. Johnson and S. Kotz and A.W. Kemp, *Univariate discrete distributions*, Wiley Interscience, 1992.
[3] D.B. Kececioglu, *Reliability Engineering Handbook*, vol.1, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
[4] LiDIA Group, *LiDIA - A library for computational number theory, Version 1.3*, Technische Universität Darmstadt, Darmstadt, Germany, 1997. Available from `http://www.informatik.tu-darmstadt.de/TI/LiDIA`.

[5] A. Stein and E. Teske, 'Explicit bounds and heuristics on class numbers in hyperelliptic function fields', University of Waterloo, Centre for Applied Cryptographic Research, Technical Report, CORR 99-26.

[6] A. Stein and E. Teske, 'Optimized baby step-giant step methods for hyperelliptic function fields', *preprint.*

[7] C.D. Terr, 'A modification of Shanks' baby-step giant-step algorithm', *Math. Comp.*, posted on March 4, 1999, PII: S 0025-5718(99)01141-2 (to appear in print).

Department of Mathematics, Royal Holloway, University of London, Egham, Surrey TW20 0EX, United Kingdom

Dept. of Combinatorics and Optimization/Centre for Applied Cryptographic Research, University of Waterloo, Waterloo, ON, N2L 3G1 Canada