Eur. Phys. J. B **65**, 131–140 (2008)
DOI: 10.1140/epjb/e2008-00320-9

THE EUROPEAN PHYSICAL JOURNAL B

# Analysis of the Karmarkar-Karp differencing algorithm

S. Boettcher[1,a] and S. Mertens[2]

[1] Department of Physics, Emory University, Atlanta GA 30322-2430, USA
[2] Institut für Theoretische Physik, Otto-von-Guericke Universität, PF 4120, 39016 Magdeburg, Germany
and
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA

Received 27 February 2008 / Received in final form 7 May 2008
Published online 9 August 2008 – © EDP Sciences, Società Italiana di Fisica, Springer-Verlag 2008

**Abstract.** The Karmarkar-Karp differencing algorithm is the best known polynomial time heuristic for the number partitioning problem, fundamental in both theoretical computer science and statistical physics. We analyze the performance of the differencing algorithm on random instances by mapping it to a nonlinear rate equation. Our analysis reveals strong finite size effects that explain why the precise asymptotics of the differencing solution is hard to establish by simulations. The asymptotic series emerging from the rate equation satisfies all known bounds on the Karmarkar-Karp algorithm and projects a scaling $n^{-c\ln n}$, where $c = 1/(2\ln 2) = 0.7213\ldots$. Our calculations reveal subtle relations between the algorithm and Fibonacci-like sequences, and we establish an explicit identity to that effect.

**PACS.** 02.60.Pn Numerical optimization – 89.75.Da Systems obeying scaling laws – 89.75.Fb Structures and organization in complex systems

## 1 Introduction

Consider a list of $n$ positive numbers. Replacing the two largest numbers by their difference yields a new list of $n-1$ numbers. Iterating this operation $n-1$ times leaves us with a single number. Intuitively we expect this number to be much smaller than all the numbers in the original list. But how small? This is the question that we address in the present paper.

The operation that replaces two numbers in a list by their difference is called *differencing*, and the procedure that iteratively selects the two *largest* numbers for differencing is known as *largest differencing method* or LDM. This method was introduced in 1982 by Karmarkar and Karp [1] as an algorithm for solving the number partitioning problem (NPP): Given a list $a_1, a_2, \ldots, a_n$ of positive numbers, find a partition, i.e. a subset $A \subset \{1, \ldots, n\}$ such that the discrepancy

$$D(A) = \Big|\sum_{i\in A} a_i - \sum_{i\notin A} a_i\Big|, \qquad (1)$$

is minimized. Obviously, LDM amounts to deciding iteratively that the two largest numbers will be put on different sides of the partition, but to defer the decision on what side to put each number. The final number then represents the discrepancy.

Despite its simple definition, the NPP is of considerable importance both in theoretical computer science and statistical physics. The NPP is NP-hard, which means (a) that no algorithm is known that is essentially faster than exhaustively searching through all $2^n$ partitions, and (b) that the NPP is computationally equivalent to many famous problems like the Traveling Salesman Problem or the Satisfiability Problem [2]. In fact, the NPP is one of Garey and Johnson's six basic NP-hard problems that lie at the heart of the theory of NP-completeness [3], and it is the only one of these problems that actually deals with numbers. Hence it is often chosen as a base for NP-hardness proofs of other problems involving numbers, like bin packing, multiprocessor scheduling [4], quadratic programming or knapsack problems. The NPP was also the base of one of the first public key crypto systems [5].

In statistical physics, the significance of the NPP results from the fact that it was the first system for which the local REM scenario was established [6,7]. The notion local REM scenario refers to systems which locally (on the energy scale) behaves like Derrida's random energy model [8,9]. It is conjectured to be a universal feature of random, discrete systems [10]. Recently, this conjecture has been proven for several spin glass models [11,12] and for directed polymers in random media [13].

Considering the NP-hardness of the problem it is no surprise that LDM (which runs in polynomial time) will generally not find the optimal solution but an approximation. Our initial question asks for the quality of the LDM

[a] e-mail: sboettc@emory.edu

solution to NPP, and to address this question we will focus on random instances of the NPP where the numbers $a_j$ are independent, identically distributed (i.i.d.) random numbers, uniformly distributed in the unit interval. Let $L_n$ denote the output of LDM on such a list. Yakir [14] proved that the expectation $\mathrm{E}\left[L_n\right]$ is asymptotically bounded by

$$n^{-b\ln n} \le \mathrm{E}\left[L_n\right] \le n^{-a\ln n}\,, \tag{2}$$

where $a$ and $b$ are (unknown) constants such that

$$b \ge a \ge \frac{1}{2\ln 2} = 0.7213\ldots\,. \tag{3}$$

In this contribution we will argue that $b = a = \frac{1}{2\ln 2}$.

The paper is organized as follows. We start with a comprehensive description of the differencing algorithm, a simple (but wrong) argument that yields the scaling (2) and a presentation of simulation data that seems to violate the asymptotic bound (3). In Section 3 we reformulate LDM in terms of a stochastic recursion on parameters of exponential variates. This recursion will then be simplified to a deterministic, nonlinear rate equation in Section 4. A numerical investigation of this rate equation reveals a structure in the dynamics of LDM that can be used as an Ansatz to simplify both the exact recursions and the rate equation. This will lead to a simple, Fibonacci like recursion (Sect. 5) and to an analytic solution of the rate equation (Sect. 6). In both cases we can derive the asymptotics including the corrections to scaling, and we claim that a similar asymptotic expansion holds for the original LDM. The latter claim is corroborated by fitting the asymptotic expansion to the available numerical data on LDM.

## 2 Differencing algorithm

The differencing scheme as described in the introduction gives the value of the discrepancy, but not the actual partition. For that we need some additional bookkeeping, which is most easily implemented in terms of graphs (Fig. 1). The algorithm maintains a list of rooted trees where each root is labeled with a number. The algorithm starts with $n$ trees of size one and the roots labeled with the numbers $a_i$. Then the following steps are iterated until a single rooted tree of size $n$ remains:

1. Among all roots, find those with the largest ($x$) and second largest ($y$) label.
2. Join nodes $x$ and $y$ with an edge, declare node $x$ as the root of the new tree and relabel it with $x - y$.

After $n - 1$ iterations all nodes are spanned by a tree whose root is labeled by the final discrepancy. This tree can easily be two-colored, and the colors represent the desired partition.

Figure 1 illustrates this procedure on the instance (4,5,6,7,8). The final two coloring corresponds to the partition (4,5,7) versus (6,8) with discrepancy 2. Note that the optimum partition (4,5,6) versus (7,8) achieves discrepancy 0.

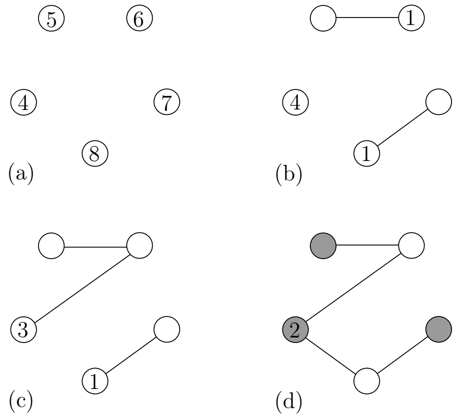

**Fig. 1.** The differencing algorithm in action.

Technically, LDM boils down to deleting items from and inserting items into a sorted list of size $n$. This can be done in time $\mathcal{O}\left(n\ln n\right)$ using an advanced data structure like a heap [15]. Hence LDM is very efficient, but how good is it? As we have already seen in the example, LDM can miss the optimal partition. And for random instances, the corridor in Eq. (2) is far above the true optimum, which is known to scale like $\Theta(\sqrt{n}\,2^{-n})$ [7]. Yet LDM yields the best-known results that can be achieved in polynomial time. Many alternative algorithms have been investigated in the past [16,17], but they all produce results worse than (2). The few algorithms that can actually compete with the Karmarkar-Karp procedure use the same elementary differencing operation [18,19]. It seems as if the differencing scheme marks an inherent barrier for polynomial time algorithms.

The following argument explains the scaling (2). The typical distance between adjacent pairs of the $n$ numbers in the interval $[0,1]$ is $n^{-1}$. Hence after $n/2$ differencing operations we are left with $n/2$ numbers in the interval $[0,n^{-1}]$. The typical distance between pairs is now $2n^{-2}$. After another round of $n/4$ differencing operations we get $n/4$ numbers in the range $[0,8n^{-3}]$. In general, after $2^k$ differencing operations we are left with $n/2^k$ numbers in the range $[0,2^{\binom{k}{2}}n^{-k}]$. Reducing the original list to a single number requires $k = \log_2 n$ differencing operations, and applying the above argument all the way down suggests that

$$\mathrm{E}\left[L_n\right] \propto n^{-c\ln n} \tag{4}$$

with

$$c = \frac{1}{2\ln 2} = 0.721\ldots\,. \tag{5}$$

As we will see, this is the right scaling, yet the argument cannot be correct. This follows from the fact that it predicts the same scaling for the paired differencing method (PDM). Here in each round all pairs of adjacent numbers are replaced by their difference in parallel. This method, however, yields an average discrepancy of order $\Theta(n^{-1})$ [20]. Yet, our analysis below suggests that (4) and (5) indeed describe the asymptotic behavior correctly, although a far more subtle treatment is required.

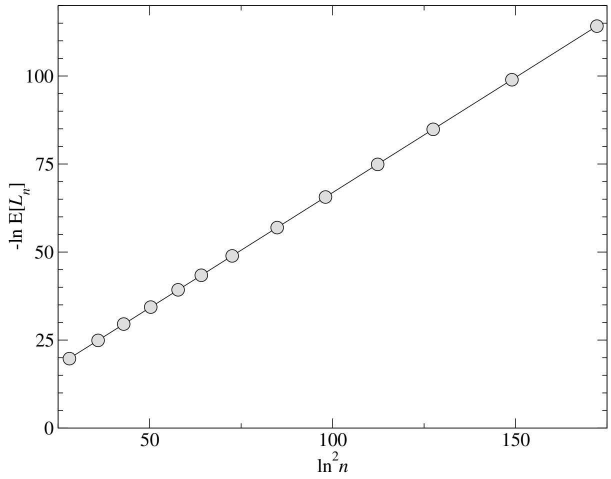

**Fig. 2.** Results of LDM applied to $n$ random i.i.d. numbers, uniformly drawn from the unit interval. Each data point represents between $10^5$ (large $n$) and $10^7$ samples (small $n$). The solid line is the linear fit $-\ln \mathrm{E}\left[L_n\right] = 1.42 + 0.65\ \ln^2 n$.

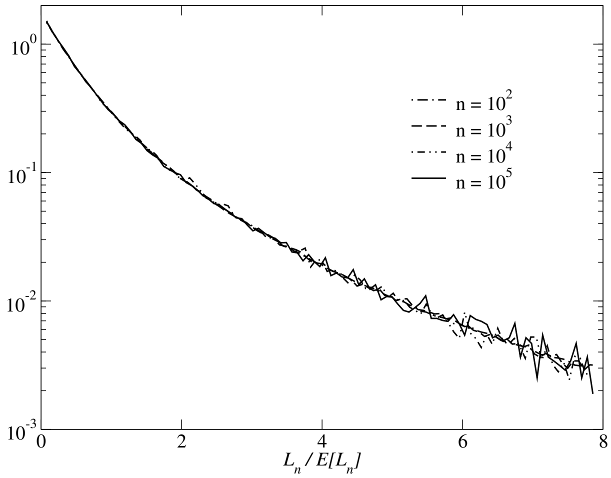

**Fig. 3.** Probability density function of $L_n/\mathrm{E}\left[L_n\right]$.

An obvious approach to find the quality of LDM are simulations. We ran LDM on random instances of varying size $n$, and Figure 2 shows the results for $\mathrm{E}\left[L_n\right]$. Apparently $\ln \mathrm{E}\left[L_n\right]$ scales like $\ln^2 n$, in agreement with (2) and (4). A linear fit seems to yield

$$c \simeq 0.65$$

for the constant in (4), which clearly violates the bound $c \geq 1/2 \ln 2$. Apparently even $n = 10^6$ is too small to see the true asymptotic behavior. This may be the reason why Monte Carlo studies of LDM never have been published.

A plot of the probability density function (pdf) of $L_n/\mathrm{E}\left[L_n\right]$ reveals a data collapse for varying values of $n$ (Fig. 3). Apparently the complete statistics of $L_n$ is asymptotically dominated by a single scale $n^{-c \ln n}$.

Some technical notes about simulating LDM are appropriate. Differencing means subtracting numbers over and over again. The numerical precision must be adjusted carefully to support this and to be able to represent the final discrepancy of order $n^{-c \ln n}$. We used the freely available GMP library [21] for the required multiple precision arithmetic and ran all simulations on $\ell$-bit integers where the number of bits ranges from $\ell = 40$ (for $n = 20$) to $\ell = 300$ for $n = 1.5 \times 10^7$. The integer discrepancies were then rescaled by $2^{-\ell}$. The pseudo random number generator was taken from the TRNG library [22].

## 3 Exact recursions

A common problem in the average-case analysis of algorithms like LDM is that numbers become conditioned and cease to be independent as the algorithm proceeds. Lueker [20] proposed to use exponential instead of uniform variates to cope with this problem. Let $X_1, \ldots, X_{n+1}$ be i.i.d. random exponentials with mean 1 and consider the partial sums $S_k = \sum_{i=1}^{k} X_i$. Then the joint distribution of the ratios $S_k/S_{n+1}$, $k = 1, \ldots, n$, is the same as that of the order statistics of $n$ i.i.d. uniform variates from $[0, 1]$ [23]. As a consequence, LDM will produce the same distribution of data no matter whether it is run on uniform variates or on $S_k/S_{n+1}$. Let $\hat{L}_n$ denote the result of LDM on the partial sums $S_1, S_2, \ldots, S_n$. Since the output of LDM is linear in its input, we have

$$\hat{L}_n \overset{D}{=} S_{n+1} L_n \,, \tag{6}$$

where $S_{n+1}$ is the sum of $n+1$ i.i.d. exponential variates and the notation $X \overset{D}{=} Y$ indicates that the random variable $X$ and $Y$ have the same distribution. The probability density of $S_{n+1}$ is the gamma density

$$g_{n+1}(s) = \frac{s^n}{n!}\,\mathrm{e}^{-s}. \tag{7}$$

Taking expectations of both sides of (6) we get

$$\mathrm{E}\left[L_n\right] = \frac{\mathrm{E}\left[\hat{L}_n\right]}{n+1}. \tag{8}$$

This allows us to derive the asymptotics of $\mathrm{E}\left[L_n\right]$ from the asymptotics of $\mathrm{E}\left[\hat{L}_n\right]$.

Exponential variates are well suited for the analysis of LDM because the sum and difference of two exponential variates are again exponential variates. Once started on exponential variates, LDM keeps working on exponentials all the time. This allows us to express the operation of LDM in terms of a recursive equation for the parameters of exponential densities [14]. We start with the following Lemma:

**Lemma 1.** *Let $X_1$ and $X_2$ be independent exponential random variables with parameter $\lambda_1$ and $\lambda_2$, resp. The probability of the event $X_1 < X_2$ is given by*

$$\mathrm{P}\left(X_1 < X_2\right) = \frac{\lambda_1}{\lambda_1 + \lambda_2}\,. \tag{9}$$

*Furthermore, conditioned on the event $X_1 < X_2$, the variables $X_1$ and $X_2 - X_1$ are independent exponentials with parameters $\lambda_1 + \lambda_2$ (for $X_1$) and $\lambda_2$ for $X_2 - X_1$.*

The proof of Lemma 1 consists of trivial integrations of the exponential densities and is omitted here.

Next we consider generalized partial sums of exponentials, described by $n$-tuples

$$(\lambda_1, \lambda_2, \ldots, \lambda_n).$$

This $n$-tuple is shorthand for the sequence of partial sums

$$\left(X_1, X_1 + X_2, \ldots, \sum_{i=1}^{n} X_i\right)$$

with $X_i = \exp(\lambda_i)$.

Now let us look at the result of one iteration of LDM on $(\lambda_1, \lambda_2, \ldots, \lambda_n)$. The two largest numbers are removed and replaced by their difference $X_n$ which is an $\exp(\lambda_n)$ variate. Lemma 1 tells us, that the probability that this number is the smallest in the list is

$$\mathrm{P}\left(X_n < X_1\right) = \frac{\lambda_n}{\lambda_1 + \lambda_n},$$

and conditioned on that event, the smallest number is an $\exp(\lambda_1 + \lambda_n)$ variate and the increment to the 2nd smallest number $X_1 - X_n$ is an independent $\exp(\lambda_1)$ variate. Conditioned on $X_n < X_1$ we get another $\lambda$-tuple as the input for the next iteration:

$$X_n < X_1 \Rightarrow (\lambda_1 + \lambda_n, \lambda_1, \lambda_2, \ldots, \lambda_{n-2}).$$

The probability that $X_n \geq X_1$ is

$$\mathrm{P}\left(X_n \geq X_1\right) = \frac{\lambda_1}{\lambda_1 + \lambda_n},$$

and in this case $X_1$ is an $\exp(\lambda_n + \lambda_1)$ variate, whereas the difference $X_n - X_1$ is an $\exp(\lambda_n)$ variate. Now the probability that the new number $X_n$ is second in the new list reads

$$\mathrm{P}\left(X_n \geq X_1 \cap X_n < X_1 + X_2\right) = \mathrm{P}\left(X_n \geq X_1 \cap X_n - X_1 < X_2\right) = \frac{\lambda_1}{\lambda_1 + \lambda_n} \frac{\lambda_n}{\lambda_2 + \lambda_n}$$

and conditioned on that event the input for the new iteration is

$$\left(\lambda_1 + \lambda_n, \lambda_2 + \lambda_n, \lambda_2, \ldots, \lambda_{n-2}\right).$$

This argument can be iterated to calculate the probability of $X_n$ becoming the $k$-th number in the new list. Denoting the partial sums by $S_k$ we get

$$\mathrm{P}\left(X_n \geq S_{k-1} \cap X_n < S_k\right) = \frac{\lambda_n}{\lambda_k + \lambda_n} \prod_{i=1}^{k-1} \frac{\lambda_i}{\lambda_i + \lambda_n} \tag{10}$$

for $k = 1, \ldots, n-2$ and conditioned on that event the new list is

$$(\lambda_1 + \lambda_n, \ldots, \lambda_k + \lambda_n, \lambda_k, \lambda_{k+1}, \ldots, \lambda_{n-2}). \tag{11}$$

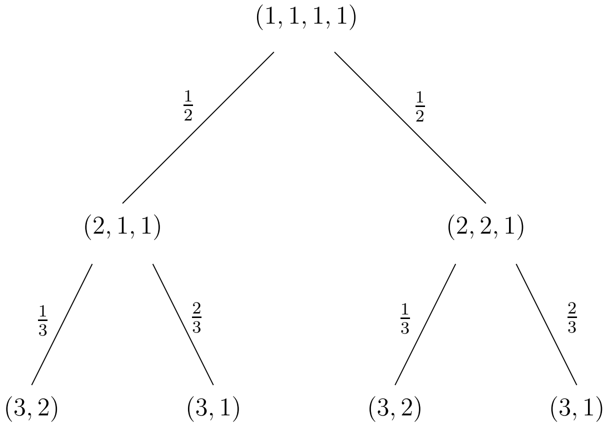

**Fig. 4.** Statistics of LDM on $n = 4$. The final difference is distributed according to $p_4(x) = \frac{2}{3}\mathrm{e}^{-x} + \frac{1}{3}2\mathrm{e}^{-2x}$.

The final case is that $X_n$ becomes the largest number in the new list. This happens with probability

$$\mathrm{P}\left(X_n \geq S_{n-2}\right) = \prod_{i=1}^{n-2} \frac{\lambda_i}{\lambda_i + \lambda_n} \tag{12}$$

and leads to the list

$$\left(\lambda_1 + \lambda_n, \ldots, \lambda_{n-2} + \lambda_n, \lambda_n\right). \tag{13}$$

In all cases we stay within the set of instances given by partial sums of independent exponentials, and we can apply equations (10–13) recursively until we have reduced the original problem to a $(\lambda_1, \lambda_2)$-instance which tells us that the final difference is an $\exp(\lambda_2)$ variate.

Figure 4 shows the result of this analysis on the input $(1, 1, 1, 1)$, our original problem with $n = 4$. We have to explore the tree that branches according to the position that is taken by the new number inserted in the shortened list. The numbers written on the edges of the tree are the probabilities for the corresponding transition. Note that we have combined the two branches emerging from the root that both lead to a $(2, 2, 1)$-configuration into a single one by adding their probabilities. In the end we get

$$p_4(x) = \frac{2}{3}\mathrm{e}^{-x} + \frac{2}{3}\mathrm{e}^{-2x},$$

for the probability density function (pdf) of $\hat{L}_4$. In general, the pdf of $\hat{L}_n$ is a sum of exponentials,

$$p_n(x) = \sum_k a_k^{(n)} k\, \mathrm{e}^{-kx} \tag{14}$$

where $a_k^{(n)}$ is the probability of LDM returning an $\exp(k)$-variate. For small values of $n$, these probabilities can be calculated by expanding the recursions explicitly (Tab. 1), but for larger values of $n$ this approach is prohibitive due the exponential growth of the number $K(n)$ of branches that have to be explored.

**Table 1.** Coefficients $a_k^{(n)}$ in (14).

| $k \setminus n$ | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|
| 1 | $\frac{2}{3}$ | $\frac{13}{24}$ | $\frac{41}{120}$ | $\frac{49}{180}$ | $\frac{431}{2520}$ |
| 2 | $\frac{1}{3}$ | $\frac{1}{6}$ | $\frac{5}{18}$ | $\frac{1}{8}$ | $\frac{527}{3456}$ |
| 3 | | $\frac{7}{24}$ | $\frac{7}{72}$ | $\frac{1073}{4320}$ | $\frac{3079}{38880}$ |
| 4 | | | $\frac{41}{180}$ | $\frac{47}{720}$ | $\frac{1229}{5600}$ |
| 5 | | | $\frac{1}{18}$ | $\frac{53}{360}$ | $\frac{149}{2100}$ |
| 6 | | | | $\frac{7}{72}$ | $\frac{486359}{5443200}$ |
| 7 | | | | $\frac{161}{4320}$ | $\frac{343}{4320}$ |
| 8 | | | | $\frac{1}{135}$ | $\frac{11}{144}$ |
| 9 | | | | | $\frac{26083}{604800}$ |
| 10 | | | | | $\frac{859}{77760}$ |
| 11 | | | | | $\frac{941}{155520}$ |
| 12 | | | | | $\frac{1}{1050}$ |
| 13 | | | | | $\frac{1}{1800}$ |

Alternatively we can explore the tree of $\lambda$-tuples by walking it randomly. Given a tuple $(\lambda_1 \ldots, \lambda_n)$, we generate a random integer $1 \leq k \leq n-1$ with probability

$$\mathrm{P}\left(k \leq \ell\right) = \begin{cases} 1 - \prod_{j=1}^{\ell} \frac{\lambda_j}{\lambda_j + \lambda_n} & (\ell < n-1) \\ 1 & (\ell = n-1) \end{cases} \tag{15}$$

and using this random $k$ we generate a new tuple of size $n-1$ according to equations (11) or (13). This process is iterated until the tuple size is two, and the final value of $\lambda_2$ is the parameter for the statistics of $\hat{L}$. The probability density of $\lambda_2/\mathrm{E}\left[\lambda_2\right]$ is shown in Figure 5. Again the data collapse corroborates the claim that the statistics of *LDM* is dominated by a single scale.

## 4 Rate equation

We can turn the exact recursions from Section 3 into a set of rate equations for the time-evolution of the *average* $\lambda$-tuple. Let $\lambda_i^t$ denote the value of $\lambda_i$ after $t$ iterations, such that

$$\left(\lambda_1^t, \lambda_2^t, \ldots, \lambda_{n-t}^t\right) \to \left(\lambda_1^{t+1}, \lambda_2^{t+1}, \ldots, \lambda_{n-t-1}^{t+1}\right) . \tag{16}$$

As explained in Section 3, at "time" $t$ a number $k$, $1 \leq k \leq n-1-t$ is chosen with probability

$$\mathrm{P}_t\left(k \leq \ell\right) = \begin{cases} 1 - \prod_{j=1}^{\ell} \frac{\lambda_j^t}{\lambda_j^t + \lambda_{n-t}^t} & (\ell < n-1-t) \\ 1 & (\ell = n-1-t) \end{cases} . \tag{17}$$

Depending on the choice of $k$, equations (11) and (13) suggest that $\lambda_i^{t+1}$ only takes on one of *two* possible values. For $1 \leq i < n-t-1$, these are

$$\lambda_i^{t+1} = \begin{cases} \lambda_i^t + \lambda_{n-t}^t & (i \leq k \leq n-t-1) \\ \lambda_{i-1}^t & (1 \leq k < i) \end{cases} , \tag{18}$$

whereas for $i = n-t-1$, the two values are

$$\lambda_{n-t-1}^{t+1} = \begin{cases} \lambda_{n-t}^t & (k = n-t-1) \\ \lambda_{n-t-2}^t & (1 \leq k < n-t-1) \end{cases} . \tag{19}$$

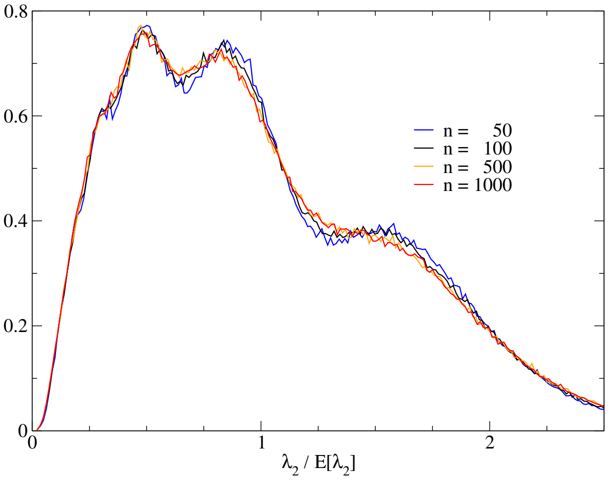

**Fig. 5.** Probability density function of $\lambda_2/\mathrm{E}\left[\lambda_2\right]$.

We introduce the shorthand

$$\mathcal{P}_i^t = \prod_{j=1}^{i-1} \frac{\lambda_j^t}{\lambda_j^t + \lambda_{n-t}^t} , \tag{20}$$

for the probability of $k \geq i$ at iteration $t$. On average, the evolution of $\lambda_i^t$ is given by the *rate equation*

$$\lambda_i^{t+1} = \lambda_{i-1}^t \left(1 - \mathcal{P}_i^t\right) + \left(\lambda_i^t + \lambda_{n-t}^t\right) \mathcal{P}_i^t , \tag{21}$$

for all $1 \leq i < n-1-t$, and at the upper boundary

$$\lambda_{n-(t+1)}^{t+1} = \lambda_{n-2-t}^t \left(1 - \mathcal{P}_{n-1-t}^t\right) + \lambda_{n-t}^t \mathcal{P}_{n-1-t}^t . \tag{22}$$

These equations are defined on the triangular domain $0 \leq t \leq n-1$, $1 \leq i \leq n-t$. The initial conditions are

$$\lambda_i^{t=0} = 1 \qquad (1 \leq i \leq n) . \tag{23}$$

As described in Section 3, the process terminates at $t = n-2$ with $\lambda_2^{n-2}$ characterizing the exponential variate for the final difference in LDM. Yet, equation (22) for $t = n-2$ implies $\lambda_1^{n-1} = \lambda_2^{n-2}$, reflecting the final, trivial differencing step, and it will prove conceptually advantageous to focus on the asymptotic properties of $\lambda_1^{n-1}$ instead.

Since the rate equation is an approximation to the exact recursion, we need to check how accurate it is. We have solved the rate equations (21–23) numerically up to $n = 5 \times 10^6$. Figure 8 shows

$$\frac{\ln\left(\lambda_1^{n-1}\left(n+1\right)\right)}{\ln^2 n}$$

from the rate equation versus $1/\ln n$. If $\lambda_1^{n-1}$ were calculated as an average from the exact recursion, the previous expression should be equal to

$$-\frac{\ln \mathrm{E}\left[L_n\right]}{\ln^2 n}$$

from the direct simulation of LDM. Figure 8 shows this quantity, too. Apparently the error introduced by approximating the exact recursion by the rate equation vanishes

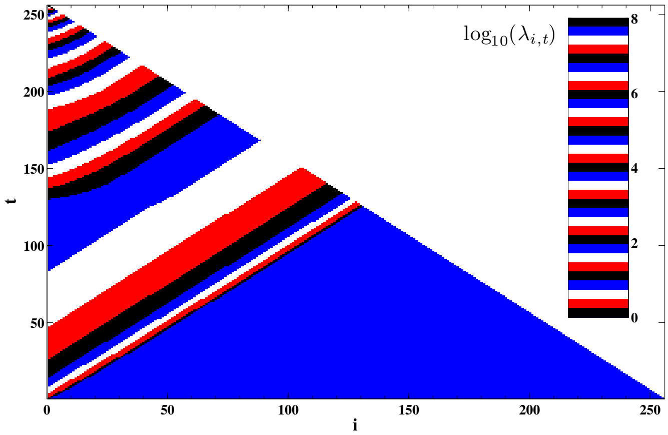

**Fig. 6.** Contour plot on a logarithmic scale for the numerical solution $\lambda_i^t$ of the rate equations (21–23) at $n = 256$. The solution is $\lambda_i^t \simeq 1$ throughout the entire lower triangle, and it increases monotonically for increasing $t$ above that. The solution rises by about a decade between each repeat of a band color. Note the ever more-rapid alternation between narrowing and widening bands, signifying regions of rapid gain interrupted by extended plateaus. The regular banded structure along diagonals $t - i =$ const justifies the similarity solution in equation (38). The only notable exceptions occur in asymptotically diminishing regions near $i = 1$ and $t = n/2, 3n/4, 7n/8, \ldots$.

**Table 2.** Parameters for (51) used in Figure 8.

| $Z$ | $f$ | $F$ | $\lambda_1^{n-1}$ | $\mathrm{E}\left[n\,L_n\right]^{-1}$ |
|---|---|---|---|---|
| $c_1$ | –1.44 | –1.45 | –1.22 | –1.24 |
| $c_2$ | –1.00 | –1.42 | –3.06 | –3.86 |
| $c_3$ | 0.72 | 1.01 | 1.23 | 1.55 |

for $n \to \infty$, and our conjecture is that the rate equation and the exact recursion are asymptotically equivalent. Judging from our numerical studies below, see Table 2, both asymptotic series have a relative difference of size $\ln\ln n/\ln^2(n)$.

The time to solve the rate equation numerically scales like $\mathcal{O}\left(n^2\right)$, so it is actually more efficient to simulate LDM directly, not least because the sampling for the latter can be done efficiently on a parallel machine. For analytic approaches, however, the rate equation is more convenient. The initial probabilities decay exponentially,

$$\mathcal{P}_i^0 = 2^{-i}, \tag{24}$$

which implies that only the first values $\lambda_1, \lambda_2, \ldots$ increase. Everywhere else, $\mathcal{P}_i$ is essentially zero, and those entries will not increase until the first term of (21) has copied the values from the low-index boundary. Hence we expect a "wavefront" of increased $\lambda$-values to travel with a velocity of one index per time step toward the upper boundary, which in turn travels with the same velocity towards the lower boundary. As can be seen from Figure 6, this traveling wavefronts of increasing heights are a hallmark of the rate equation for all times $t$. We will use this intuitive picture for an Ansatz to analyze both the exact recursion and the rate equation in the next two sections.

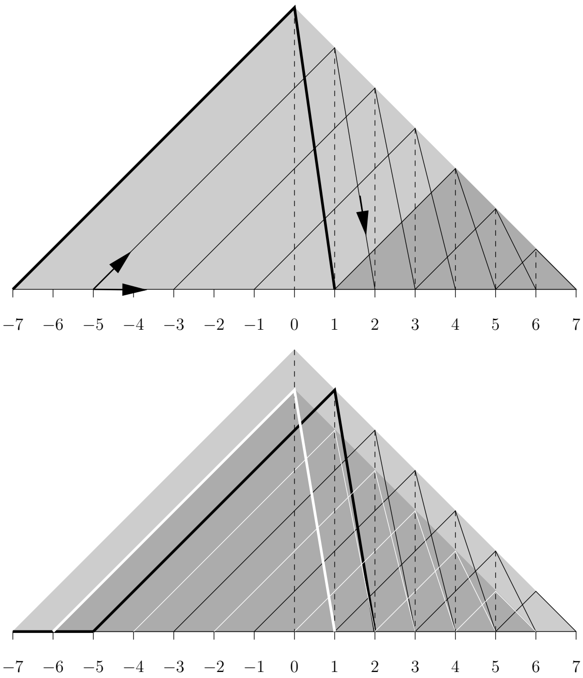

**Fig. 7.** Proof of the Fibonacci recursion: The number of different paths from the leftmost point to the rightmost point in the triangle for $n$ is the sum of the number of paths in the corresponding triangle of size $[n/2]$ (top) plus the number of paths in the triangle of size $n - 1$ (bottom).

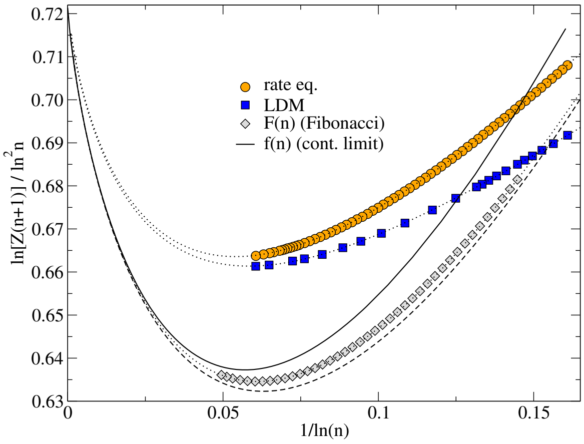

**Fig. 8.** Four models of LDM: Direct simulation ($Z = 1/\mathrm{E}\left[n\,L_n\right]$), rate equation ($Z = \lambda_1^{n-1}$), the Fibonacci model $Z = F(n)$ from (29) and the similarity solution $Z = f(n)$ of the continuous rate equation, given by (49). The dashed line represents (50). All dotted lines are numerical fits of the type (51).

## 5 Fibonacci model

Both the exact recursion and the rate equations yield

$$\lambda_1^{t+1} = \lambda_1^t + \lambda_{n-t}^t \tag{25}$$

for the lower boundary that we are ultimately interested in. This recursion connects the lower and the upper boundaries at $i = 1$ and at $i = n - t$. Unfortunately, $\lambda_{n-1}^t$

depends in a complicated way on entries of the $\lambda$-tuple at different times and different places. However, Figure 6 suggests a *similarity Ansatz*

$$\lambda_i^t = \lambda_{i-x}^{t-x}\,, \tag{26}$$

which makes the upper boundary readily available:

$$\begin{aligned} \lambda_1^{t+1} &= \lambda_1^t + \lambda_1^{2t-n+1} \qquad (0 \le t < n-1) \\ \lambda_1^t &= 1 \qquad\qquad\qquad\qquad (t \le 0). \end{aligned} \tag{27}$$

Note that we have extended the initial conditions $\lambda_i^t = 1$ to hold for all negative times, too.

It turns out that one can express the final value $\lambda_1^{n-1}$ of this recursion in terms of the corresponding values in smaller systems, which leads to a simple recursion in $n$. To derive this recursion it is convenient to visualize (27) in terms of paths in a right-angled triangle $\Delta_n$ (Fig. 7). The hypotenuse of $\Delta_n$ represents $t$ and ranges from $-n+1$ to $n-1$, the height is $n-1$. Let us discuss the basic mechanism for the example $n=8$. The final recursion reads

$$\lambda_1^7 = \lambda_1^6 + \lambda_1^5\,,$$

and the two terms on the right hand side correspond to two paths: one that connects 6 with 7 along the hypotenuse, the other connects 5 with 7 along the path that is "reflected" at the right leg of $\Delta_8$. In our case a reflected path moves diagonally upward until it touches the right leg above point $i$. From there it moves downward to point $i+1$. This peculiar "law of refraction" implies that only every second point of the left half of the hypotenuse is connected to the right half by a reflected path.

We can apply the recursion again and write

$$\begin{aligned} \lambda_1^7 &= \lambda_1^6 + \lambda_1^5 \\ &= \lambda_1^5 + \lambda_1^3 + \lambda_1^4 + \lambda_1^1. \end{aligned}$$

Here we have connected 6 with 5 along the hypotenuse and with 3 along a reflected path, and similarly for 5. We iterate this path finding process until all paths end on the left half of the hypotenuse (negative $t$). Here the paths collect the initial values $\lambda_1^t = 1$, hence $\lambda_1^7$ equals the number of different paths that connect the points $-7, -5, \ldots, -1$ to the point 7 on the hypotenuse. Instead of considering each paths that starts on the left half of the hypotenuse separately we let all paths start in the leftmost point $-7$. The rule for path finding then is: if you are on an even index, move one unit to the right. If you are on an odd index, there are two branches: one to the right, the other 45 degrees upward and reflected down to the hypotenuse. Obviously, $\lambda_1^7$ equals the number of different paths that connects the leftmost point of $\Delta_8$ to the rightmost point according to this rules. Let $T_n(i)$ denote the number of paths that connect the point $i$ with $n-1$ in $\Delta_n$. Then we have

$$T_n(-n+1) = \lambda_1^{n-1}.$$

Now, starting at $-n+1$, we have two choices: move upward for a reflection that will take us to point 1 or move along the hypotenuse to point $-n+3$:

$$T_n(-n+1) = T_n(1) + T_n(-n+3)\,.$$

As we can see in Figure 7 (top), the number of paths from 1 to $n-1$ is exactly the same as the total number of paths in $\Delta_{n/2}$. Hence

$$T_n(1) = T_{n/2}(-n/2+1)\,.$$

Similarly, the number of paths from $-n+3$ to $n-1$ equals the total number of paths in a slightly smaller triangle, as can be seen in Figure 7 (bottom). Hence we have

$$T_n(-n+3) = T_{n-1}(-n+2)\,,$$

and all three equations yield

$$T_n(-n+1) = T_{n/2}(-n/2+1) + T_{n-1}(-n+2).$$

The derivation of a corresponding equation for odd values of $n$ is straightforward. If we define

$$F(n) := T_n(-n+1) = \lambda_1^{n-1}, \tag{28}$$

the recursion for $T_n$ translates into the Fibonacci like recursion

$$\begin{aligned} F(n) &= F(n-1) + F([n/2]) \\ F(1) &= 1 \end{aligned} \tag{29}$$

where $[x]$ refers to the integer part of $x$. The resulting sequence is known as A033485 in [24]. The generating function $g(z) = \sum_n F(n)\, z^n$ satisfies the functional equation

$$g(z)\,(1-z) = z + (1+z)\,g(z^2)\,, \tag{30}$$

and is given by

$$g(z) = \frac{1}{2}\left(\frac{(1-z)^{-1}}{\prod_{k\ge 0}(1-z^{2^k})} - 1\right)\,. \tag{31}$$

$F(n)$ can be evaluated numerically for values of $n$ that are larger than the values feasible for simulations of LDM or for solving the rate equation. The bottleneck for calculating $F(n)$ is memory, not CPU time, since $n/2$ values must be stored to get $F(n)$. With 3 GByte of memory, we managed to calculate $F(n)$ for $n \le 6 \times 10^8$. We will derive the asymptotics of $F(n)$ in the next section.

Figure 8 shows $F(n)$ within the same scaling as the simulations of LDM and the numerical solution of the rate equation. Apparently the similarity Ansatz does not capture the full complexity of the LDM algorithm or the rate equation. Yet it yields a very similar *qualitative* behavior. And in the next section we will show that

$$\lim_{n\to\infty} \frac{\ln F(n)}{\ln^2 n} = \frac{1}{2\ln 2}\,. \tag{32}$$

## 6 Continuum limit

To analyze the rate equations (21–23), it is convenient to consider the continuum limit for $n \to \infty$. Asymptotically, a continuum solution may differ from the discrete problem in corrections of order $1/n$. As we will see, such corrections are inaccessible, as the asymptotic expansion is a series in terms of $1/\ln(n)$.

We rewrite equation (21) in terms of discrete differences,

$$\lambda_i^{t+1}-\lambda_i^t=-\left(\lambda_i^t-\lambda_{i-1}^t\right)+\left(\lambda_i^t-\lambda_{i-1}^t+\lambda_{n-t}^t\right)\mathcal{P}_i^t\,. \quad (33)$$

Setting

$$\begin{aligned} t&=sn(0\le s\le 1)\,,\\ i&=xn(0\le x\le 1-s)\,,\\ \lambda_i^t&=y(x,s)\,, \end{aligned} \quad (34)$$

we obtain for large $n$

$$\frac{1}{n}\left[\frac{\partial}{\partial x}+\frac{\partial}{\partial s}\right]y(x,s)=\Pi(x,s)\left[\frac{\partial}{n\partial x}y(x,s)+y(1-s,s)\right], \quad (35)$$

where we have set

$$\mathcal{P}_i^t\to\Pi(x,s)=\exp\left\{n\int_0^x d\xi\;\ln\alpha(\xi,s)\right\} \quad (36)$$

with

$$\alpha(x,s)=\frac{y(x,s)}{y(x,s)+y(1-s,s)}\le 1\,. \quad (37)$$

The left-hand side of equation (35), as well as the numerical solution of the full rate equations (21–23) displayed in Figure 6, again suggest a *similarity Ansatz*

$$y(x,s)=\gamma(s-x)\,. \quad (38)$$

This Ansatz yields immediately for equation (35):

$$0=\Pi(x,s)\left[-\frac{1}{n}\gamma'(s-x)+\gamma(2s-1)\right]. \quad (39)$$

For almost all $x>0$, the right-hand side vanishes by virtue of $\Pi(x,s)\to 0$, as indicated by equation (36) for $\alpha<1$ and $n\to\infty$. Correspondingly, $\Pi(x=1-s,s)=0$ at the upper boundary, which justifies the similarity solution for the continuum limit of equation (22). Yet, $\Pi(x=0,s)=1$ for all $s$, hence we are left with

$$\frac{1}{n}\gamma'(s)=\gamma(2s-1)\,, \quad (40)$$

which can be interpreted as the continuous version of (27). From the initial conditions of the discrete problem in (23) it is clear that $y(x,0)=1$. For the similarity solution, this implies that

$$\gamma(s)=1,\qquad(-1\le s\le 0)\,. \quad (41)$$

Integrating (40), we formally obtain

$$\gamma(s)=\gamma(0)+n\int_0^s d\xi\,\gamma(2\xi-1)\,. \quad (42)$$

Thus, we can evaluate the integral for $0\le s\le\frac{1}{2}$ to get

$$\gamma(s)=1+ns,\qquad\left(0\le s\le\frac{1}{2}\right). \quad (43)$$

We can continue this process for $\frac{1}{2}\le s\le\frac{3}{4}$, i.e., $0\le 2s-1\le\frac{1}{2}$, *exactly* the domain of validity of (43), to obtain

$$\begin{aligned}\gamma(s)&=\gamma(0)+n\int_0^{\frac{1}{2}}d\xi\,\gamma(2\xi-1)+n\int_{\frac{1}{2}}^s d\xi\,\gamma(2\xi-1)\,,\\ &=1+ns+\frac{n^2}{4}(2s-1)^2\qquad\left(\frac{1}{2}\le s\le\frac{3}{4}\right). \end{aligned} \quad (44)$$

The emergent pattern is best represented by defining

$$\gamma_k(s)=\gamma(s),\qquad\left(1-2^{1-k}\le s\le 1-2^{-k}\right), \quad (45)$$

for $k=0,1,2,\ldots$, where equations (41–44) represent $k=0$, 1 and 2. In general, we find that

$$\gamma_{k+1}(s)=\gamma_k\left(1-2^{-k}\right)+n\int_{1-2^{-k}}^s d\xi\,\gamma_k(2\xi-1)\,, \quad (46)$$

which is solved by

$$\gamma_k(s)=\sum_{j=0}^k\frac{n^j}{j!\,2^{\binom{j}{2}}}\left(2^{j-1}s-2^{j-1}+1\right)^j. \quad (47)$$

For any $n$, we are interested in $\gamma(s\to 1)\sim\lim_{t\to n-1}\lambda_1^t$, hence

$$\gamma(1)=\lim_{k\to\infty}\gamma_k\left(1-2^{-k}\right)=\sum_{j=0}^{\infty}\frac{n^j}{j!\,2^{\binom{j}{2}}}\,, \quad (48)$$

which concludes our solution of (40). The sum for $\gamma(1)$ still depends on $n$, hence we define

$$f(n)=\sum_{j=0}^{\infty}\frac{n^j}{j!\,2^{\binom{j}{2}}}\,. \quad (49)$$

Now $f(n)$ can be evaluated numerically for very large values of $n$. Figure 8 shows the result for $n\le 2^{2000}$. Don't try this at home unless you have a computer algebra system. Interestingly, $\ln f(n)/\ln^2 n$ asymptotically approaches a value that is extremely close to $1/2\ln 2$. In fact, an asymptotic analysis (see Appendix) reveals

$$\begin{aligned}\frac{\ln\left[f(n)(n+1)\right]}{\ln^2 n}&\simeq\frac{1}{2\ln 2}+\frac{1}{\ln n}\left(\frac{\ln\ln 2+1}{\ln 2}+\frac{3}{2}\right)\\ &+\frac{1}{\ln^2 n}\left(\frac{\ln 2+4\ln\ln 2}{8}-\frac{\ln^2\ln 2}{2\ln 2}\right)\\ &-\frac{\ln\ln n}{\ln n}\frac{1}{\ln 2}-\frac{\ln\ln n}{\ln^2 n}+\frac{\ln^2\ln n}{\ln^2 n}\frac{1}{2\ln 2}, \end{aligned} \quad (50)$$

which is the dashed line in Figure 8. The dotted lines are numerical least square fits of the $\ln\ln n$ terms of this scaling, i.e., fits of the form

$$\begin{aligned}\frac{\ln\left[Z(n)(n+1)\right]}{\ln^2 n}&\simeq\frac{1}{2\ln 2}+\frac{1}{\ln n}\left(\frac{\ln\ln 2+1}{\ln 2}+\frac{3}{2}\right)\\ &+\frac{1}{\ln^2 n}\left(\frac{\ln 2+4\ln\ln 2}{8}-\frac{\ln^2\ln 2}{2\ln 2}\right)\\ &+\frac{\ln\ln n}{\ln n}c_1+\frac{\ln\ln n}{\ln^2 n}c_2+\frac{\ln^2\ln n}{\ln^2 n}c_3\,. \end{aligned} \quad (51)$$

with values for $c_i$ as shown shown in Table 2. Note that the series (49) is a solution of (40) and the first terms of the asymptotic expansion (50) have been derived independently in the context of dynamical systems [25].

## 7 Conclusion

The numerical data supports the claim that the complete statistics of LDM is dominated by a single scale $\sim n^{-c\ln n}$, not just the expectation as described in (2). The available data is not sufficient to pin down the precise asymptotic scaling, however. In fact a naive extrapolation of the available data even contradicts the known asymptotic bound (3). With its $\mathcal{O}(n \ln n)$ complexity, LDM is a very efficient algorithm, but probing the asymptotics requires $\ln n$ to be large. This discrepancy of scales eliminates simulations as a means to study the asymptotics of LDM and calls for alternative approaches.

We have taken a step in the direction of a rigorous asymptotic analysis by mapping the differencing algorithm onto a rate equation. The structure seen in the evolution of this rate equation (Fig. 6) suggests a similarity Ansatz (26). With the help of this Ansatz we could reduce the exact recursion in $\lambda$-space to the Fibonacci model (29). The asymptotics of this model can be calculated, and it agrees with (2) and (3). The same Ansatz plugged into the rate equation even allows us to calculate the first terms of an asymptotic expansion (50). Although our Ansatz does not yield a proof, the extracted asymptotic behavior satisfies all previous constraints and provides a consistent interpretation of the numerical results. Hence, our rate equations pave the way for further systematic investigations.

We appreciate stimulating discussions George E. Hentschel and Cris Moore. S.M. enjoyed the hospitality of the Cherry L. Emerson Center for Scientific Computation at Emory University, where part of this work was done. Most simulations were run on the Linux-Cluster TINA at Magdeburg University. S.M. was sponsored by the European Community's FP6 Information Society Technologies programme, contract IST-001935, EVERGROW.

## Appendix A: Asymptotic analysis

To evaluate the series (49) we apply Laplace's saddle-point method for sums as described on p. 304 of reference [26]. For

$$a_j = \frac{n^j}{j!\,2^{\binom{j}{2}}} = e^{\phi_j}\,,$$

the saddle point is determined by $D\phi_j = \phi_j - \phi_{j-1} = 0$, i.e., $0 = D\ln(a_j) = \ln(a_j/a_{j-1})$, or

$$1 = \frac{a_j}{a_{j-1}} = \frac{n}{j\,2^{j-1}}. \tag{A.1}$$

Hence, we obtain a moving ($n$-dependent) saddle point at

$$j_0 \sim \frac{\ln n}{\ln 2} - \frac{\ln\left(\frac{\ln n}{\ln 2}\right)}{\ln 2} + 1 + \frac{\ln\left(\frac{\ln n}{\ln 2}\right)}{\ln 2 \ln n} - \frac{1}{\ln n} + \dots, \tag{A.2}$$

including terms to the order needed to determine $f(n)$ up to the correct prefactor. We keep the $1/\ln(n)$-corrections, since $\phi_j$ contains terms like $j_0 \ln(n)$. In particular, it is

$$\phi_j = j\,\ln n - \ln j! - \frac{j(j-1)}{2}\ln 2\,. \tag{A.3}$$

As the saddle point $j_0$ is large for large $n$, we can replace $j!$ by its Stirling-series [26]. Then, we expand around the saddle point by substituting $j = j_0 + \eta$, keeping only terms to *2nd* order in $\eta$ and those that are non-vanishing for $n \to \infty$. We find

$$\phi_{j_0+\eta} \sim \frac{\ln^2 n}{2\ln 2} - \frac{1}{2}\ln(2\pi) - \frac{\ln 2}{2}\eta(\eta+1) + \mathcal{C}(n)\,,$$

with log-polynomial corrections

$$\begin{aligned}\mathcal{C}(n) \sim &-\frac{\ln n}{\ln 2}\left[\ln\left(\frac{\ln n}{\ln 2}\right) - 1 - \frac{\ln 2}{2}\right] \\ &+ \left[\frac{1}{2\ln 2}\ln^2\left(\frac{\ln n}{\ln(2)}\right) - \ln\left(\frac{\ln n}{\ln(2)}\right)\right].\end{aligned} \tag{A.4}$$

We finally obtain for the asymptotic expansion of (50):

$$\begin{aligned} f(n) &\sim \int_{-\infty}^{\infty} d\eta\, \exp\left(\phi_{j_0+\eta}\right) \\ &\sim \frac{2^{\frac{1}{8}}}{\sqrt{\ln 2}}\exp\left\{\frac{\ln^2 n}{2\ln 2} + \mathcal{C}(n)\right\}. \end{aligned} \tag{A.5}$$

## References

1. N. Karmarkar, R.M. Karp, *The differencing method of set partitioning*. Technical Report UCB/CSD 81/113, Computer Science Division (University of California, Berkeley, 1982)
2. S. Mertens, C. Moore, *The Nature of Computation* (Oxford University Press, Oxford, 2008)
3. M.R. Garey, D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness.* edited by W.H. Freeman (New York, 1997)
4. H. Bauke, S. Mertens, A. Engel, Phys. Rev. Lett. **90**, 158701 (2003)
5. R.C. Merkle, M.E. Hellman, IEEE Trans. Info. Th. **24**, 525 (1978)
6. S. Mertens, Phys. Rev. Lett. **84**, 1347 (2000)
7. C. Borgs, J. Chayes, B. Pittel, Rand. Struct. Alg. **19**, 247 (2001)
8. B. Derrida, Phys. Rev. Lett. **45**, 79 (1980)
9. B. Derrida, Phys. Rev. B **24**, 2613 (1981)
10. H. Bauke, S. Mertens, Phys. Rev. E **70**, 025102(R) (2004)
11. A. Bovier, I. Kurkova, Commun. Math. Phys. **263**, 513 (2006)

12. A. Bovier, I. Kurkova, J. Stat. Phys. **126**, 933 (2007)
13. I. Kurkova, Elect. J. Probability **13**, 5 (2008)
14. B. Yakir, Math. Oper. Res. **21**, 85 (1996)
15. J. Kleinberg, É. Tardos, *Algorithm Design* (Addison Wesley, Boston, 2006)
16. D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevron, Operations Research **39**, 378 (1991)
17. W. Ruml, J.T. Ngo, J. Marks, S.M. Shieber, J. Opti. Th. Appl. **89**, 251 (1996)
18. R.E. Korf, Art. Intell. **106**, 181 (1998)
19. R.H. Storer, S.W. Flanders, S.D. Wu, Annals of Operations Research **63**, 465 (1996)
20. G.S. Lueker, Oper. Res. Lett. **6**, 285 (1987)
21. The GNU Multiple Precision Arithmetic Library. `http://gmplib.org`
22. TRNG - portable random number generators for parallel computing. `http://trng.berlios.de/`
23. W. Feller, *An Introduction to Probability and Its Applications*, 2nd edn. (John Wiley & Sons, 1972) Vol. 2
24. N.J.A. Sloane, The on-line encyclopedia of integer sequences `http://www.research.att.com/~njas/sequences`
25. C. Moore, P. Lakdawala, Phys. D **135**, 24 (2000)
26. C.M. Bender, S.A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers* (McGraw-Hill, New York, 1978)