Unbiased Black-Box Complexity of Parallel Search

Golnaz Badkobeh[1], Per Kristian Lehre[2], and Dirk Sudholt[1]

[1] University of Sheffield, UK
[2] University of Nottingham, UK

Abstract. We propose a new black-box complexity model for search algorithms evaluating λ search points in parallel. The parallel unbiased black-box complexity gives lower bounds on the number of function evaluations *every* parallel unbiased black-box algorithm needs to optimise a given problem. It captures the inertia caused by offspring populations in evolutionary algorithms and the total computational effort in parallel metaheuristics. Our model applies to all unary variation operators such as mutation or local search. We present lower bounds for the Leading-Ones function and general lower bound for all functions with a unique optimum that depend on the problem size and the degree of parallelism, λ. The latter is tight for OneMax; we prove that a $(1+\lambda)$ EA with adaptive mutation rates is an optimal parallel unbiased black-box algorithm.

1 Introduction

Black-box optimisation describes a challenging realm of problems where no algebraic model or gradient information is available. The problem is regarded a black box, and knowledge about the problem in hand can only be obtained by evaluating candidate solutions. General-purpose metaheuristics like evolutionary algorithms, simulated annealing, ant colony optimisers, tabu search, and particle swarm optimisers are well suited for black-box optimisation as they generally work well without any problem-dependent knowledge.

A lot of research has focussed on designing powerful metaheuristics, yet it is often unclear which search paradigm works best for a particular problem class, and whether and how better performance can be obtained by tailoring a search paradigm to the problem class in hand.

The black-box complexity of search algorithms captures the difficulty of problem classes in black-box optimisation. It describes the minimum number of function evaluations that *every* black-box algorithm needs to make to optimise a problem from a given class. It provides a rigorous theoretical foundation through capturing limits to the efficiency of all black-box search algorithms, providing a baseline for performance comparisons across all known and future metaheuristics as well as tailored black-box algorithms. Also it prevents algorithm designers from wasting effort on trying to achieve impossible performance.

The first black-box complexity model by Droste et al. [6] makes no restriction on the black-box algorithm. This leads to some unrealistic results, such

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 892–901, 2014.

as polynomial black-box complexities of NP-hard problems [6]. Subsequent research introduced refined models that restrict the power of black-box algorithms, leading to more realistic results [4–6, 18]. Lehre and Witt introduced the unbiased black-box model [13] where black-box algorithms may only use operators without a search bias (see Section 2). This model initially considered unary operators (such as mutation) and was later extended to higher arity operators (such as crossover) [3] and more general search spaces [17]. It also led to the discovery of more efficient EA variants [2].

A shortcoming of the above models is that they do not capture the implicit or explicit parallelism at the heart of many common search algorithms. Evolutionary algorithms (EAs) such as $(\mu+\lambda)$ EAs or (μ,λ) EAs generate λ offspring in parallel. Using a large offspring population in many cases can decrease the number of generations needed to find an optimal solution[1]. However, the number of function evaluations may increase as evolution can only act on information from the previous generation. A large offspring population can lead to inertia that slows down the optimisation process. Existing black-box models are unable to capture this inertia as they assume all search points being created in sequence.

The same goes for parallel metaheuristics such as island models evolving multiple populations in parallel (see, e. g. [14]). Parallelisation can decrease the number of generations, or parallel time. But the overall computational effort, the number of function evaluations across all islands, may increase. Lässig and Sudholt [11] used the following notion. Let T_λ be the random number of generations an island model with λ islands (each creating one offspring) needed to find a global optimum for a given problem. If using λ islands can decrease the parallel time by a factor of order λ, compared to just one island, $\lambda \cdot \mathrm{E}(T_\lambda) = O(\mathrm{E}(T_1))$, this is called a *linear speedup* (with regards to the parallel time, the number of generations). A linear speedups means that the total number of function evaluations, $\lambda \cdot \mathrm{E}(T_\lambda)$, does not increase beyond a constant factor.

Recent work [11,12,15] considered illustrative problems from pseudo-Boolean optimisation and combinatorial optimisation, showing sufficient conditions for linear speedups. However, the absence of matching lower bounds makes it impossible to determine exactly for which parameters λ linear speedups are achieved.

We provide a parallel black-box model that captures and quantifies the inertia caused by offspring populations of size λ and parallel EAs evaluating λ search points in parallel. We present lower bounds on the black-box complexity for the well known LO problem and for the general class of functions with a unique optimum, revealing how the number of function evaluations increases with the problem size n and the degree of parallelism, λ. The results complement existing upper bounds [11], allowing us to characterise the realm of linear speedups, where parallelisation is effective.

Our lower bound for functions with a unique optimum is asymptotically tight: we show that for the ONEMAX problem, a $(1+\lambda)$ EA with an adaptive mutation rate is an optimal parallel unbiased black-box algorithm. Adaptive mutation

[1] This does not hold for all problems; De Jong, Jansen, and Wegener constructed problems where offspring populations drastically increase the number of generations [9].

rates decrease the expected running time by a factor of $\ln \ln \lambda$, compared to the $(1+\lambda)$ EA with the standard mutation rate $1/n$ (see He, Chen, and Yao [7]).

2 A Parallel Black-Box Model

Following Lehre and Witt [13], we only use unary unbiased variation operators, i. e., operators creating a new search point out of one search point. This includes local search, mutation in evolutionary algorithms, but it does not include recombination. Unbiasedness means that there is no bias towards particular regions of the search space; in brief, for $\{0,1\}^n$, unbiased operators must treat all bit values $0, 1$ and all bit positions $1, \ldots, n$ symmetrically (see [13, 17] for details). This is the case for many common operators such as standard bit mutation.

Unbiased black-box algorithms query new search points based on the past history of function values, using unbiased variation operators. We define a λ-*parallel unbiased black-box algorithm* in the same way, with the restriction that in each round λ queries are made in parallel (see Algorithm 1). These λ queries only have access to the history of evaluations from previous rounds; they cannot access information from queries made in the same round. We refer to these λ search points as *offspring* to indicate search points created in the same round.

Algorithm 1. λ-parallel unbiased black-box algorithm

1. Let $t := 0$. Choose $x^1(0), \ldots, x^\lambda(0)$ uniformly at random, compute $f(x^1(0)), \ldots, f(x^\lambda(0))$, and let $I(0) := \{f(x^1(0)), \ldots, f(x^\lambda(0))\}$.
2. **repeat**
3. **for** $1 \leq i \leq \lambda$ **do**
4. Choose an index $0 \leq j \leq t$ according to $I(t)$.
5. Choose an unbiased variation operator $p_v(\cdot \mid x(j))$ according to $I(t)$.
6. Generate $x^i(t+1)$ according to p_v.
7. **end for**
8. **for** $1 \leq i \leq \lambda$ **do**
9. Compute $f(x^i(t))$ and let $I(t) := I(t) \cup \{f(x^i(t))\}$.
10. **end for**
11. Let $t := t + 1$.
12. **until** termination condition met

This black-box model includes offspring populations in evolutionary algorithms, for example $(\mu+\lambda)$ EAs or (μ,λ) EAs (modulo minor differences in the initialisation). It can further model parallel evolutionary algorithms such as cellular EAs with λ cells, or island models with λ islands, each of which generates one offspring in each generation.

The *unbiased black-box complexity (uBBC)* of a function class $\mathcal{F}$ is the minimum worst-case runtime among all unbiased black-box algorithms [13] (equivalent to Algorithm 1 with $\lambda = 1$). The *unbiased λ-parallel black-box complexity (λ-upBBC)* of a function class $\mathcal{F}$ is defined as the minimum worst-case number

of function evaluations among all unbiased λ-parallel algorithms satisfying the framework of Algorithm 1.

With increasing λ access to previous queries becomes more and more restricted. It is therefore not surprising that the black-box complexity is non-decreasing with growing λ. For every family of function classes $\mathcal{F}_n$ and all $\lambda \in \mathbb{N}$,

$$\text{uBBC}(\mathcal{F}_n) = 1\text{-upBBC}(\mathcal{F}_n) \leq 2\text{-upBBC}(\mathcal{F}_n) \leq 3\text{-upBBC}(\mathcal{F}_n) \dots \quad (1)$$

$$\text{and } \text{uBBC}(\mathcal{F}_n) \leq \lambda\text{-upBBC}(\mathcal{F}_n) \leq \lambda \cdot \text{uBBC}(\mathcal{F}_n) \quad (2)$$

as any unbiased algorithm can be simulated by a λ-parallel unbiased black-box algorithm using one query in each round. Due to (1), there is a *cut-off point*

$$\lambda^* := \sup\{\lambda \mid \exists c > 0, n_0 \; \forall n \geq n_0 \colon \lambda\text{-upBBC}(\mathcal{F}_n) \leq c \cdot \text{uBBC}(\mathcal{F}_n)\}$$

such that c is a constant and for all $\lambda \leq \lambda^*$ the λ-parallel unbiased black-box complexity of $\mathcal{F}$ is asymptotically equal to the regular unbiased black-box complexity. In this realm, parallelisation is most effective as the number of function evaluations does not increase (beyond constant factors). The number of rounds for an optimal black-box algorithm, $\text{uBBC}(\mathcal{F}_n)/\lambda$, corresponds to the parallel time if all λ evaluations are performed on parallel processors. By (2) $\text{uBBC}(\mathcal{F}_n)/\lambda$ is non-increasing with λ, and for $\lambda \leq \lambda^*$ it decreases by a factor of $\Theta(\lambda)$. Such speedups were called *linear speedups* in [11].

The (1+λ) EA maintains the current best search point x and creates λ offspring by flipping each bit in x independently with probability p (with default $p = 1/n$). The best offspring replaces its parent if it has fitness at least $f(x)$.

3 Parallel Black-Box Complexity of LeadingOnes

We consider the function $\text{Lo}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$, counting the number of leading ones in x. Similarly, $\text{Lz}(x)$ counts the number of leading zeros in x. We first provide a tool for estimating the progress made by λ trials, which may or may not be independent. It is based on moment-generating functions (mgf).

Lemma 1. *Given $X_1, \dots, X_\lambda \in \mathbb{N}$, where X_is are random variables, not necessarily independent. Define $X_{(\lambda)} := \max_{i \in [\lambda]} X_i$, if there exists $\eta, D \geq 0$, such that for all $i \in [\lambda]$, it holds $E\left(e^{\eta X_i}\right) \leq D$, then $E\left(X_{(\lambda)}\right) \leq (\ln(D\lambda) + 1)/\eta$.*

Proof. Note first that for any $i \in [\lambda]$ and $j \in \mathbb{N}$, it follows from Markov's inequality that $\Pr(X_i \geq j) = \Pr(e^{\eta X_i} \geq e^{\eta j}) \leq e^{-\eta j} \text{E}\left(e^{\eta X_i}\right) \leq e^{-\eta j} D$. Now, let $k := \ln(D\lambda)/\eta$. It then follows by a union bound that

$$\begin{aligned}
\text{E}\left(X_{(\lambda)}\right) &= \sum_{i=1}^{\infty} \Pr(X_{(\lambda)} \geq i) \leq k + \sum_{i=1}^{\infty} \Pr(X_{(\lambda)} \geq k + i) \\
&\leq k + \sum_{i=1}^{\infty} \sum_{j=1}^{\lambda} \Pr(X_j \geq k + i) \leq k + \sum_{i=1}^{\infty} \lambda e^{-\eta(k+i)} D \\
&= k + e^{-\eta k} \frac{D\lambda}{e^{\eta} - 1} \leq k + e^{-\eta k} D\lambda/\eta = (\ln(D\lambda) + 1)/\eta.
\end{aligned}$$

For the LO function, the λ-parallel black-box complexity is as follows.

Theorem 1. *The λ-parallel unbiased black-box complexity of* LO *is*

$$\Omega\left(\frac{\lambda n}{\ln(\lambda/n)} + n^2\right) \quad \text{and} \quad O\big(\lambda n + n^2\big).$$

The cut-off point is $\lambda^*_{\text{LO}} = n$. The corresponding parallel time for an optimal algorithm is $\Omega\left(\frac{n}{\ln(\lambda/n)} + \frac{n^2}{\lambda}\right)$ and $O\left(n + \frac{n^2}{\lambda}\right)$.

This result solves an open problem from [11], confirming that the analysis of the realm of linear speedups for LO from [11] is tight.

Proof (of Theorem 1). The upper bound follows from Lässig and Sudholt [12, Theorem 1] for a (1+λ) EA, as within the context of this bound the (1+λ) EA is equivalent to an island model with complete communication topology.

A lower bound $\Omega(n^2)$ follows from [13], hence the statement holds for the case $\lambda = O(n)$. In case that $\lambda = \omega(n)$, we proceed by drift analysis. Let the "potential" of a search point x be $\max_{0\le j\le t, 1\le i\le\lambda}\{\text{LO}(x^i(j)), \text{LZ}(x^i(j)), n/2\}$, and define the potential of the algorithm, P_t at time t to be the largest potential among all search points produced until time t.

Assume that the potential in generation t is $P_t = k$. In any generation t, let X_i for $i \in [\lambda]$ be the indicator variable for the event that all of the first $k+1$ bit-positions in individual i are 1-bits (or 0-bits). Furthermore, let Y_i be the number of consecutive 1-bits (respectively 0-bits) from position $k+2$ and onwards, ie., the number of "free riders".

Following the same arguments as in [13], the probability that $X_i = 1$ is no more than $1/(k+1) = O(1/n)$. Defining $M := \sum_{i=1}^{\lambda} X_i$, we therefore have $\mathrm{E}(M) = O(\lambda/n)$. Each random variable Y_i, $i \in [\lambda]$, is stochastically dominated by a geometric random variable Z_i with parameter 1/2. The expected progress in potential is therefore

$$\mathrm{E}\left(\Delta_{(\lambda)}\right) = \mathrm{E}\left(\max_{i\in[\lambda]} X_i Y_i\right) \le \mathrm{E}\left(\max_{i\in[M]} Z_i\right).$$

The mgf of the geometric random variable Z_i is $M_{Z_i}(\eta) = 1/(2-e^{\eta})$. The tower property of the expectation and Lemma 1 with $\eta := \ln(3/2)$ and $D := 2$ give

$$\begin{aligned}\mathrm{E}\left(\Delta_{(\lambda)}\right) &\le \mathrm{E}\left(\mathrm{E}\left(\max_{i\in[M]} Z_i \mid M\right)\right)\\ &\le \mathrm{E}\left((\log(DM)+1)/\eta\right) \le (\log(\mathrm{E}(DM))+1)/\eta = O(\log(\lambda/n)),\end{aligned}$$

where the last inequality follows from Jensen's inequality and the last equality follows from $\log(\lambda/n) = \Omega(1)$. With overwhelmingly high probability, the initial potential is at least $n/2$. Hence, by classical additive drift theorems [8], the expected number of rounds to reach the optimum is $\Omega(n/\log(\lambda/n))$. Multiplying by λ gives the number of function evaluations.

4 Parallel Black-Box Complexity of Functions with Unique Optimum

De Jong, Jansen, and Wegener [9] considered the (1+λ) EA and established a cut-off point for λ where the running time increases from $\Theta(n \log n)$ to $\omega(n \log n)$:

$$\lambda^*_{(1+\lambda) \text{ EA on OneMax}} = \Theta((\ln n)(\ln \ln n)/(\ln \ln \ln n)) \tag{3}$$

Recently, He, Chen, and Yao [7] presented the following tight bound for all λ:

Theorem 2 (He, Chen, Yao [7]). *The expected optimisation time of the (1+λ) EA on* OneMax *for $\lambda \geq 3$ is*

$$\Theta\left(n \cdot \frac{\lambda \ln \ln \lambda}{\ln \lambda} + n \log n\right).$$

We show that the parallel black-box complexity is lower than the bound from Theorem 2 for large λ by a factor of $\ln \ln \lambda$.

Theorem 3. *For any $\lambda \leq e^{\sqrt{n}}$ the λ-parallel unbiased unary black-box complexity for any function with a unique optimum is at least*

$$\Omega\left(\frac{\lambda n}{\ln \lambda} + n \log n\right).$$

This bound is tight for OneMax*, where the cut-off point is*

$$\lambda^*_{\text{OneMax}} = \Theta(\log(n) \cdot \log \log n).$$

The corresponding parallel time for an optimal algorithm is $\Omega\left(\frac{n}{\ln \lambda} + \frac{n \log n}{\lambda}\right)$.

Note that the cut-off point is higher than the cut-off point for the (1+λ) EA with the standard mutation rate $p = 1/n$ from (3) and [9].

For the proof we consider the progress made during a round of λ variations. Let the 0-"potential" of a search point x be $\min\{|x|_0, n/(8e)\}$, where $|x|_0$ is the number of 0-bits in x. Similarly, define the 1-"potential" of a search point x as $\min\{|x|_1, n/(8e)\}$. Let s be the minimum 0-potential among all search points queried in past rounds. Let r be the number of flipped bits during a variation, then for any search point with m number of zeros, denote the progress by $\Delta(s, m, r)$. The progress is the difference between s and the potential of the new generated point, there is no progress if this difference is negative. Let Z be the number of 0-bits that flipped to 1, then there are $r - Z$ new 0-bits that were originally 1. Therefore, the number of 0-bits in the new generated search point is $m - Z + (r - Z)$ where Z can be described by the hypergeometric distribution with parameters n, m and r. We only make progress if the number of 0-bits in the new search point is less than s. Hence the progress (decrease in potential) is

$$\Delta(s, m, r) = \max\{Z - (r - Z) + (s - m), 0\} = \max\{2Z - r + s - m, 0\}.$$

We show a tail inequality for hypergeometric variables that is more precise than Chvátal's bound [1] and use this to derive a progress bound. A proof of the former is omitted due to space restrictions.

Lemma 2. *Let Z be a hypergeometrically distributed random variable with parameters n (number of balls), m (number of red balls), and r (number of balls drawn). If $m < n/(2e)$ then for any $z \geq r/2$, $Pr(Z = z) \leq (2em/n)^z$.*

Lemma 3. *Let $\Delta_{(\lambda)} = \Delta_{(\lambda)}(s, m_i, r_i)$ be the maximum of λ random variables $\Delta(s, m_i, r_i)$ for arbitrary $s \leq m_i \leq n/2$ and r_i, $1 \leq i \leq \lambda$. For $s \leq n/(8e)$ we have $E\left(\Delta_{(\lambda)}\right) = O(\log(\lambda))$.*

Proof. If $\frac{n}{4e} < m_i \leq n/2$ then we use Chvátal's tail bound [1]: $\Pr\left(Z \geq \mathrm{E}\left(Z\right) + r\delta\right) \leq \exp(-2\delta^2 r)$, where $\mathrm{E}\left(Z\right) = \frac{rm}{n}$, then:

$$\begin{aligned}\Pr\left(\Delta(s, m_i, r_i) > 0\right) &= \Pr\left(Z \geq \frac{r_i + m_i - s}{2}\right) \\ &= \Pr\left(Z \geq \frac{r_i m_i}{n} + r_i \cdot \left(\frac{r_i + m_i - s}{2r_i} - \frac{m_i}{n}\right)\right) \\ &\leq \Pr\left(Z \geq \mathrm{E}\left(Z\right) + r_i \cdot \frac{n}{8er_i}\right) \leq \exp\left(-\frac{n^2}{32e^2 r_i}\right)\end{aligned}$$

This means that the probability of making any progress is exponentially small, for any r_i. Thus we assume that $m_i \leq \frac{n}{4e}$ for all i in the following. Applying Lemma 2 to a hypergeometric random variable Z_i with parameters m_i and r_i we have, for all $z \in \mathbb{N}_0$,

$$\begin{aligned}&\Pr\left(\Delta(s, m_i, r_i) = z\right) \\ &= \Pr\left(Z_i = \frac{z + r_i + m_i - s}{2}\right) \leq \left(\frac{2em_i}{n}\right)^{(z + r_i + m_i - s)/2} \leq \left(\frac{1}{2}\right)^{z/2}\end{aligned}$$

hence $\mathrm{E}\left(e^{\eta Z_i}\right) \leq D$ for $\eta := \ln(4/3)$ and $D := 9 + 6\sqrt{2}$. Applying Lemma 1 proves $\mathrm{E}\left(\Delta_{(\lambda)}\right) = O(\log \lambda)$.

Proof (of Theorem 3). The upper bound for OneMax will be shown later in Theorem 4. The lower bound $\Omega(n \log n)$ follows from unbiased unary black-box complexity [13]. Hence, it suffices to prove the lower bound $\Omega(\lambda n / \ln \lambda)$.

Without loss of generality, we assume that the search point 1^n is the optimum. Following [13], we assume a "mirrored" sampling process, where every time a bit string x is queried (including in the initial generation), the algorithm queries the complement bit string $\overline{x}$ for "free". Hence, the 1-potential and the 0-potential (as defined above) are the same after each generation, and we apply drift analysis with respect to this potential. Variation of a search point with m 1-bits is symmetric to a variation of a search point with $n - m$ 1-bits, hence we can assume $m \leq n/2$. By a Chernoff bound, the initial potential is $n/(8e)$ with overwhelmingly high probability. Let Δ_0 be the progress due to reduction of the 0-potential, and Δ_1 be the progress due to reduction of the 1-potential. By Lemma 3, the expected change in potential per round is no more than $\max\{\Delta_0, \Delta_1\} = O(\log \lambda)$. Hence, by the additive drift theorem [8], the expected number of rounds until one of the search points 0^n or 1^n is obtained is $\Omega(n / \log \lambda)$. Multiplying by λ proves the claim.

5 An Optimal Parallel Black-Box Algorithm for OneMax

The following theorem shows that the lower bound on the black-box complexity from Theorem 3 is tight. We show that the (1+λ) EA has a better optimisation time if the mutation rate is chosen adaptively, according to the current best fitness. This is similar to common ideas from artificial immune systems, particularly the clonal selection algorithm. Adaptive mutation rates for OneMax have been studied by Zarges [19], however the standard parameters for the clonal selection algorithm were too drastic to even obtain polynomial running times. Better results were obtained when using a population-based adaptation [20].

The following result reveals an optimal choice for the mutation rate of the (1+λ) EA, depending on n and λ.

Theorem 4. *On OneMax, the expected number of function evaluations of the (1+λ) EA with an adaptive mutation rate $p = \max\{\ln(\lambda)/(n\ln(en/i)),\ 1/n\}$, where i is the number of zeros in the current search point, for any $\lambda \le e^{\sqrt{n}}$, is at most*

$$O\left(\frac{\lambda n}{\ln \lambda} + n \log n\right).$$

The parallel time (number of generations) is $O\left(\frac{n}{\ln \lambda} + \frac{n \log n}{\lambda}\right)$.

Proof. Let i be the current number of zeros and p be the mutation rate. The probability of decreasing the number of zeros by any $k \in \mathbb{N}$ with $k \le i$ is at least

$$\begin{aligned}\Pr(\Delta \ge k) &\ge \binom{i}{k} \cdot p^k \cdot (1-p)^{n-k} \\ &\ge \frac{i^k}{k^k} \cdot p^k \cdot (1-p)^{n-k} = (1-p)^{n-k} \cdot \left(\frac{ip}{k}\right)^k.\end{aligned}$$

Then the probability that one of λ offspring will decrease the number of zeros by at least k is at least, using $1-(1-p)^\lambda \ge 1-e^{-p\lambda} \ge 1-1/(1+p\lambda) = p\lambda/(1+p\lambda)$,

$$\Pr\left(\Delta_{(\lambda)} \ge k\right) \ge 1-(1-\Pr(\Delta \ge k))^\lambda \ge \frac{\lambda(1-p)^{n-k} \cdot (ip/k)^k}{1+\lambda(1-p)^{n-k} \cdot (ip/k)^k}.$$

Hence for any $k \le i$ the expected drift is at least

$$\mathrm{E}\left(\Delta_{(\lambda)}\right) \ge k \cdot \frac{\lambda(1-p)^{n-k} \cdot (ip/k)^k}{1+\lambda(1-p)^{n-k} \cdot (ip/k)^k}.$$

For $i > en/\ln\lambda$, which implies $pn > 1$, we set $k := pn = \ln(\lambda)/\ln(en/i)$. We have $k \le i$ since $k \le \ln(\lambda) \le \sqrt{n} \le en/\ln\lambda$. We use $k := 1$ for $i \le en/\ln\lambda$, the realm where $p = 1/n$. This results in the following drift function h:

$$h(i) := \begin{cases} \frac{\lambda(1-1/n)^{n-1} \cdot i/n}{1+\lambda(1-1/n)^{n-1} \cdot i/n} & \text{if } i \le en/\ln\lambda \\ pn \cdot \frac{\lambda(1-p)^{n-pn} \cdot (i/n)^{pn}}{1+\lambda(1-p)^{n-pn} \cdot (i/n)^{pn}} & \text{otherwise} \end{cases}$$

We estimate the number of function evaluations by multiplying the number of generations by λ. The number of generations is estimated using Johannsen's variable drift theorem [10] in the variant from [16], with the above function h. This gives an upper bound of

$$\begin{aligned}\frac{\lambda}{h(1)} + \int_1^n \frac{\lambda}{h(i)}\,\mathrm{d}i &= \frac{1+\lambda(1-1/n)^{n-1}\cdot 1/n}{(1-1/n)^{n-1}\cdot 1/n} + \lambda\int_1^n \frac{1}{h(i)}\,\mathrm{d}i \\ &\leq \lambda + en + \lambda\int_1^{en/\ln\lambda} \frac{1}{h(i)}\,\mathrm{d}i + \lambda\int_{en/\ln\lambda}^n \frac{1}{h(i)}\,\mathrm{d}i.\end{aligned}$$

The first terms are at most

$$\begin{aligned}&\lambda + en + \lambda\int_1^{en/\ln\lambda} \frac{1+\lambda(1-1/n)^{n-1}\cdot i/n}{\lambda(1-1/n)^{n-1}\cdot i/n}\,\mathrm{d}i \\ \leq\ & \frac{\lambda en}{\ln\lambda} + en\left(1 + \int_1^{en/\ln\lambda}\frac{1}{i}\,\mathrm{d}i\right) \leq \frac{\lambda en}{\ln\lambda} + en\cdot(2+\ln n).\end{aligned}$$

The second integral is bounded as

$$\begin{aligned}&\int_{en/\ln\lambda}^n \frac{1+\lambda(1-p)^{n-pn}\cdot(i/n)^{pn}}{pn\cdot(1-p)^{n-pn}\cdot(i/n)^{pn}}\,\mathrm{d}i \\ \leq\ & \int_0^n \frac{\lambda\ln(en/i)}{\ln\lambda}\,\mathrm{d}i + \frac{1}{\ln\lambda}\int_{en/\ln\lambda}^n \frac{\ln(en/i)}{e^{-pn}\cdot(i/n)^{pn}}\,\mathrm{d}i \\ =\ & \frac{2\lambda n}{\ln\lambda} + \frac{1}{\ln\lambda}\int_{en/\ln\lambda}^n \ln(en/i)\cdot(en/i)^{pn}\,\mathrm{d}i \\ =\ & \frac{2\lambda n}{\ln\lambda} + \frac{1}{\ln\lambda}\int_{en/\ln\lambda}^n \ln(en/i)\cdot\lambda\,\mathrm{d}i \leq \frac{3\lambda n}{\ln\lambda}.\end{aligned}$$

Together, we get an upper bound of $(3+e)\lambda n/\ln(\lambda) + en\cdot(2+\ln n)$.

6 Conclusions

We have introduced the parallel unbiased black-box complexity to quantify the limits on the performance of parallel search heuristics, including offspring populations. We proved that *every* λ-parallel unbiased black-box algorithm needs at least $\Omega(\lambda n/\log(\lambda) + n\log n)$ function evaluations on every function with unique optimum, and at least $\Omega(\lambda n/(\log(\lambda/n)) + n^2)$ function evaluations on Lo. Corresponding parallel times are by a factor of λ smaller. For Lo and OneMax we identified the cut-off point for λ, above which the asymptotic number of function evaluations increases, compared to non-parallel algorithms ($\lambda = 1$). All smaller λ allow for linear speedups with regard to the parallel time. For OneMax this cut-off point is higher than that for the standard (1+λ) EA; optimal performance for all λ is achieved by a (1+λ) EA with an adaptive mutation rate.

References

1. Chvátal, V.: The tail of the hypergeometric distribution. Discrete Math. 25(3), 285–287 (1979)
2. Doerr, B., Doerr, C., Ebel, F.: Lessons from the black-box: fast crossover-based genetic algorithms. In: Proc. of GECCO 2013, pp. 781–788. ACM (2013)
3. Doerr, B., Johannsen, D., Kötzing, T., Lehre, P.K., Wagner, M., Winzen, C.: Faster black-box algorithms through higher arity operators. In: Proc. of FOGA 2011, pp. 163–172. ACM (2011)
4. Doerr, B., Winzen, C.: Towards a complexity theory of randomized search heuristics: Ranking-based black-box complexity. In: Kulikov, A., Vereshchagin, N. (eds.) CSR 2011. LNCS, vol. 6651, pp. 15–28. Springer, Heidelberg (2011)
5. Doerr, B., Winzen, C.: Playing Mastermind with Constant-Size Memory. Theory of Computing Systems (2012)
6. Droste, S., Jansen, T., Wegener, I.: Upper and lower bounds for randomized search heuristics in black-box optimization. Theory of Computing Systems 39(4), 525–544 (2006)
7. He, J., Chen, T., Yao, X.: Average drift analysis and its application. CoRR, abs/1308.3080 (2013)
8. He, J., Yao, X.: A Study of Drift Analysis for Estimating Computation Time of Evolutionary Algorithms. Natural Computing 3(1), 21–35 (2004)
9. Jansen, T., De Jong, K.A., Wegener, I.: On the choice of the offspring population size in evolutionary algorithms. Evolutionary Computation 13, 413–440 (2005)
10. Johannsen, D.: Random Combinatorial Structures and Randomized Search Heuristics. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany and the Max-Planck-Institut für Informatik (2010)
11. Lässig, J., Sudholt, D.: General upper bounds on the running time of parallel evolutionary algorithms. Evolutionary Computation (in press), `http://www.mitpressjournals.org/doi/pdf/10.1162/EVCO_a_00114`
12. Lässig, J., Sudholt, D.: Analysis of speedups in parallel evolutionary algorithms for combinatorial optimization. In: Asano, T., Nakano, S.-i., Okamoto, Y., Watanabe, O. (eds.) ISAAC 2011. LNCS, vol. 7074, pp. 405–414. Springer, Heidelberg (2011)
13. Lehre, P.K., Witt, C.: Black-box search by unbiased variation. Algorithmica 64(4), 623–642 (2012)
14. Luque, G., Alba, E.: Parallel Genetic Algorithms–Theory and Real World Applications. Springer (2011)
15. Mambrini, A., Sudholt, D., Yao, X.: Homogeneous and heterogeneous island models for the set cover problem. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part I. LNCS, vol. 7491, pp. 11–20. Springer, Heidelberg (2012)
16. Rowe, J.E., Sudholt, D.: The choice of the offspring population size in the (1,λ) EA. In: Proc. of GECCO 2012, pp. 1349–1356 (2012)
17. Rowe, J.E., Vose, M.D.: Unbiased black box search algorithms. In: Proc. of GECCO 2011, p. 2035. ACM, New York (2011)
18. Teytaud, O., Gelly, S.: General lower bounds for evolutionary algorithms. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 21–31. Springer, Heidelberg (2006)
19. Zarges, C.: Rigorous runtime analysis of inversely fitness proportional mutation rates. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 112–122. Springer, Heidelberg (2008)
20. Zarges, C.: On the utility of the population size for inversely fitness proportional mutation rates. In: Proc. of FOGA 2009, pp. 39–46. ACM (2009)