

ОБОБЩЕННАЯ МУТАЦИЯ С ТЯЖЕЛЫМИ
ХВОСТАМИ ДЛЯ ЭВОЛЮЦИОННЫХ
АЛГОРИТМОВА.В. ЕРЕМЕЕВ,  Д.В. СИЛАЕВ, В.А. ТОПЧИЙ *Представлено А.В. Пяткиным*

Abstract: The heavy-tailed mutation operator, proposed by Doerr, Le, Makhmara, and Nguyen (2017) for evolutionary algorithms, is based on the power-law assumption of mutation rate distribution. Here we generalize the power-law assumption using a regularly varying constraint on the distribution function of mutation rate. In this setting, we generalize the upper bounds on the expected optimization time of the $(1 + (\lambda, \lambda))$ genetic algorithm obtained by Antipov, Buzdalov and Doerr (2022) for the OneMax function class parametrized by the problem dimension n . In particular, it is shown that, on this function class, the sufficient conditions of Antipov, Buzdalov and Doerr (2022) on the heavy-tailed mutation, ensuring the $O(n)$ optimization time in expectation, may be generalized as well. This optimization time is known to be asymptotically faster than what can be achieved by the $(1 + (\lambda, \lambda))$ genetic algorithm with any static mutation rate. A new version of the heavy-tailed mutation operator is proposed, satisfying the generalized conditions, and promising results of computational experiments are presented.

EREMEEV, A.V., SILAEV, D.V., TOPCHII, V.A., GENERALIZED HEAVY-TAILED MUTATION FOR EVOLUTIONARY ALGORITHMS.

© 2024 ЕРЕМЕЕВ А.В., СИЛАЕВ Д.В., ТОПЧИЙ В.А..

Работа выполнена при поддержке Математического Центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2022-282.

Поступила 15 апреля 2024 г., опубликована 1 ноября 2024 г.

Keywords: Evolutionary algorithms, regularly varying functions, heavy-tailed mutation, optimization time.

1 Введение

Отличительной особенностью эволюционных алгоритмов (ЭА) является имитация процесса эволюционной адаптации биологической популяции к условиям окружающей среды. При этом особи соответствуют пробным точкам в пространстве решений задачи оптимизации, а *приспособленность* особей определяется значениями целевой функции с учетом штрафа за нарушение ограничений задачи, если такие имеются. Построение новых пробных точек в ЭА осуществляется посредством операторов *мутации* и *кроссинговера*. При использовании последнего ЭА принято называть *генетическими алгоритмами*. При решении безусловных задач псевдодулевой максимизации $\max\{f(x) : x \in \{0, 1\}^n\}$, или минимизации $\min\{f(x) : x \in \{0, 1\}^n\}$, где целевая функция $f : \{0, 1\}^n \rightarrow \mathbb{R}$, одним из наиболее часто используемых операторов мутации является *стандартная мутация* [11], где каждый бит имеющейся строки $x \in \{0, 1\}^n$ независимо от других меняет свое значение с заданной вероятностью p . В данной работе мы будем полагать, что при стандартной мутации на каждой итерации ЭА с распределением $\text{Bin}(n, p)$ выбирается число мутлируемых битов ℓ , и очередной потомок получается из родительского решения внесением изменений в случайно выбранных ℓ битах.

Основной характеристикой работоспособности эволюционных алгоритмов при решении задач оптимизации является *время оптимизации*, обозначаемое далее T , которое исчисляется в количестве вычислений функции приспособленности до первого достижения оптимума с начала работы алгоритма [7]. Обычно исследуется математическое ожидание времени оптимизации или среднее время оптимизации. В настоящей работе исследуется время оптимизации генетического алгоритма с вычислительной схемой из [2] при максимизации функции приспособленности $\text{ONEMAX}(x) = \sum_{i=1}^n x_i$. Семейство таких функций, параметризованное размерностью n , является одним из базовых тестовых примеров в теории эволюционных вычислений, на котором оценивается эффективность ЭА при решении простых задач. В частности, при высоком значении среднего времени оптимизации на ONEMAX отдельные алгоритмы или значения настраиваемых параметров ЭА признаются заведомо неэффективными [14, 21].

Для описания асимптотического поведения величин, связанного с неограниченным ростом размерности n , будут использоваться стандартные обозначения $O(\cdot)$. Кроме того, полагаем что функция $t(n)$ принадлежит множеству $\Theta(g(n))$, если существуют положительные константы c_1 и c_2 , а также неотрицательное целое число n_0 такие, что $c_2g(n) \leq t(n) \leq c_1g(n)$ для всех $n \geq n_0$ (см., например, [3], гл. 3). При

вводе новых обозначений в формулах используется символ $:=$, и двоеточие находится со стороны определяемой величины.

1.1. Известные результаты. Из соображений симметрии следует вывод о том, что ЭА, основанные на стандартной мутации, одинаково ведут себя как на ONEMAX, так и на любой функции $\text{ONEMAX}_z : \{0, 1\}^n \rightarrow \mathbb{R}$, определенной как

$$\text{ONEMAX}_z(x) := |\{i \in \{1, \dots, n\} : x_i = z_i\}|$$

при любой битовой строке $z \in \{0, 1\}^n$. Из работы Эрдеша и Реньи [9] вытекает, что любой алгоритм, получающий информацию только из запросов к функции приспособленности ONEMAX_z , в среднем потребует $\Omega(n/\log n)$ вычислений функции приспособленности до первого попадания в оптимальное решение z , причем эта оценка не может быть улучшена.

Простейший вариант ЭА, где популяция состоит из одной особи, известен еще из работы [10] Растригина, как *алгоритм локального поиска с пересчетом при неудачном шаге*. Частный случай такого алгоритма обозначается (1+1) EA и представляет собой рандомизированный вариант локального поиска, где переход к новой пробной точке осуществляется посредством стандартной мутации текущего решения. Как было показано в [22], в случае ONEMAX алгоритм (1+1) EA является наилучшим по времени оптимизации в классе эволюционных алгоритмов с данным оператором мутации.

Из теории и практики использования ЭА известно, что выбор численности популяции λ и значений параметров мутации и кроссинговера зачастую является нетривиальной задачей и требует трудоемких экспериментальных исследований или глубокого понимания свойств решаемой задачи [6, 4, 8, 13].

Что касается выбора параметра p для стандартного оператора мутации, большое количество классических результатов основывается на предположении, что значение $p = 1/n$ или около того – хороший выбор по умолчанию (см., например, [15]). При функции приспособленности ONEMAX наиболее точная оценка времени оптимизации для (1+1) EA и $p = 1/n$ получена в [12], она равна $en \log n + c_1 n + 0,5e \log n + c_2 + O(n^{-1} \log n)$, где $c_1 \approx 1,8925$, $c_2 \approx 0,5979$. Кроме того, известно [16], что для (1+1) EA на ONEMAX только выбор $p = \Theta(1/n)$ может обеспечить среднее время оптимизации порядка $O(n \log n)$. Если же рассмотреть линейные функции на $\{0, 1\}^n$, то как показано в [17], $p = (1 + o(1))/n$ – асимптотически наилучший параметр мутации для (1+1) EA.

Авторы [5] разработали генетический алгоритм $(1 + (\lambda, \lambda))$ GA с новым оператором кроссинговера, устранивающим «неудачные» мутации. На

каждой итерации алгоритма $(1 + (\lambda, \lambda))$ GA от единственной родительской особи порождаются λ потомков независимо друг от друга на равном расстоянии Хэмминга ℓ от родителя, при этом $\ell \sim \text{Bin}(n, p)$. Далее отбирается лучшее по приспособленности из этих решений и к нему применяется оператор кроссинговера, имеющий параметр c . С вероятностью c оператор кроссинговера использует биты из лучшего потомка, а с вероятностью $1 - c$ – биты из родительского решения. Таким образом создается еще λ особей, и лучшая из этих λ особей принимается в качестве нового родителя, если она не уступает прежнему родителю по приспособленности. Теоретический анализ времени оптимизации показал, что алгоритм $(1 + (\lambda, \lambda))$ GA при многих значениях настраиваемых параметров оказывается асимптотически быстрым на ONEMAX, чем большинство классических эволюционных алгоритмов.

Как было продемонстрировано в [18], выбор параметра мутации для многоэкстремальных задач значительно более сложен, чем для ONEMAX. Чтобы преодолеть эту трудность, в [18] было предложено использовать случайный выбор для параметра мутации p в соответствии с распределением с тяжелым хвостом, а именно, с усеченным степенным распределением с показателем $-\beta < -1$. При этом сначала выбирается случайное число $\alpha \in \{1, \dots, \lfloor \frac{n}{2} \rfloor\}$, так что вероятность выбора $\alpha = k$ пропорциональна $k^{-\beta}$, $k \leq \lfloor \frac{n}{2} \rfloor$, затем полагают $p = \alpha/n$ и применяют стандартную мутацию с этим значением p . На каждой итерации ЭА значения α выбираются независимо.

Как показано в [19] для случая ONEMAX, оптимальный выбор фиксированного значения параметра мутации p на все время работы $(1 + (\lambda, \lambda))$ GA дает время оптимизации

$$E[T] = \Theta \left(n \sqrt{\frac{\log(n) \log \log \log(n)}{\log \log n}} \right), \quad (1)$$

что асимптотически меньше, чем среднее время оптимизации многих известных эволюционных алгоритмов.

В работе [2] Антипин, Буздалов и Доерр показали эффективность быстрой мутации при оптимизации функции ONEMAX. Здесь рассматривался генетический алгоритм $(1 + (\lambda, \lambda))$ GA из [5], совмещенный с оператором быстрой мутации. В [2] для $(1 + (\lambda, \lambda))$ GA с быстрой мутацией при специфическом выборе распределений случайных величин λ и p получена верхняя оценка среднего времени оптимизации порядка $O(n)$ на ONEMAX, что меньше чем среднее время оптимизации $(1 + (\lambda, \lambda))$ GA с любой фиксированной вероятностью мутаций, как следует из (1). В данном алгоритме и численность популяции λ и параметр быстрой мутации p имеют усеченное степенное распределение с верхними границами $\lambda \leq u_n$ и $p \leq u_n/n$, соответственно. Линейная оценка в [2] имеет место, когда степенной показатель β удовлетворяет неравенствам $2 < \beta < 3$ и $u_n \geq \ln^{1/(3-\beta)} n$.

Основной результат настоящей работы, представленный в теоремах 1 и 3, показывает что верхние оценки на математическое ожидание времени оптимизации $(1 + (\lambda, \lambda))$ GA, аналогичные полученным в [2], справедливы не только для усеченных степенных распределений случайных величин λ и p , но и для более широкого класса распределений, описываемого в терминах правильно меняющихся ограничений на функцию распределения этой величины. Как следует из (1), полученная нами линейная оценка на среднее время оптимизации, также как линейная оценка из [2], оказывается асимптотически меньше, чем среднее время оптимизации $(1 + (\lambda, \lambda))$ GA при любом неизменном в ходе выполнения алгоритма параметре мутации p . Частный случай данного результата, полученный без использования аппарата правильно меняющихся функций, представлен в [20].

1.2. Обозначения и определения. Обозначим $\mathbb{N}_m := \{k : k \in \mathbb{N}, k \leq m\}$, $S := \{0, 1\}^n$, $|S| = 2^n$. Введем норму и расстояние Хэмминга $|x| = \sum_{i=1}^n x_i$ и $|x - y| = \sum_{i=1}^n |x_i - y_i|$ для $x, y \in S$. Обозначим через x^* решение с единицами на всех позициях, а $Z_s = \{x \in S : |x - x^*| = s\}$ – совокупность решений, имеющих ровно s нулей, $s = 0, \dots, n$. В частности, $Z_0 = \{x^*\}$.

Пусть $\lambda(n)$, $n \in \mathbb{N}$ – набор независимых в совокупности случайных величин (далее СВ) с областями допустимых значений из подмножества \mathbb{N}_{u_n} , где $u_n \leq 0.5n$ – максимально возможное значение $\lambda(n)$ с положительной вероятностью. Другими словами, вероятности $\mathbf{P}(\lambda(n) = k) = p_{n,k} \geq 0$ только при $k \in \mathbb{N}_{u_n}$ и $p_{n,u_n} > 0$. Ограничения на $p_{n,k}$ введем позже. СВ $\lambda_*(n)$ с различными индексами вместо * независимы и одинаково распределены с $\lambda(n)$. Мы исследуем случай $u = u(n) \rightarrow \infty$ при $n \rightarrow \infty$.

Опишем исследуемый алгоритм \mathcal{A} , предшественником которого является $(1 + (\lambda, \lambda))$ EA из [5] с детерминированными вероятностями мутации $p = \lambda(n)/n$ и параметром кроссинговера $c = \lambda^{-1}(n)$. Позднее в [2] была предложена рандомизация $(1 + (\lambda, \lambda))$ EA из [5] по численности популяции $\lambda(n)$, где вероятности $\mathbf{P}(\lambda(n) = k) = p_{n,k}$ выражаются через степенные функции. Данный алгоритм в [2] получил название *быстрый $(1 + (\lambda, \lambda))$ генетический алгоритм*. При выборе параметра $\lambda(n)$ по степенному закону и параметре мутации $p = \lambda(n)/n$ алгоритм из [2] имеет мутацию с тяжелыми хвостами, совпадающую с предложенной в [18]. В настоящей работе мы отказываемся от явных выражений $p_{n,k}$, а приводим только ограничения на функцию распределения СВ $\lambda(n)$. В остальном рассматриваемый алгоритм \mathcal{A} совпадает с быстрым $(1 + (\lambda, \lambda))$ генетическим алгоритмом.

Алгоритм \mathcal{A} : $(1 + (\lambda, \lambda))$ генетический алгоритм с правильно меняющимися ограничениями на функцию распределения $\lambda(n)$ при верхнем пределе ее значений u_n , максимизирующий функцию $f : \{0, 1\}^n \rightarrow \mathbb{R}$

1. $x \leftarrow$ random bit string of length n ;
2. **while** not terminated **do**
3. Choose λ from $[1..u]$ with $\mathbf{P}(\lambda = k) = p_{n,k}$;
4. Choose $\ell \sim \text{Bin}(n, \lambda(n)/n)$;
5. **for** $i \in [1..\lambda]$ **do**
6. $x^{(i)} \leftarrow$ a copy of x ;
7. Flip ℓ bits in $x^{(i)}$ chosen uniformly at random;
8. **end**
9. $x' \leftarrow \arg \max_{z \in \{x^{(1)}, \dots, x^{(\lambda)}\}} f(z)$;
10. **for** $i \in [1..\lambda]$ **do**
11. Create $y^{(i)}$ by taking each bit from x' with probability λ^{-1} and from x with probability $(\lambda - 1)\lambda^{-1}$;
12. **end**
13. $y \leftarrow \arg \max_{z \in \{y^{(1)}, \dots, y^{(\lambda)}\}} f(z)$;
14. **if** $f(y) \geq f(x)$ **then**
15. $x \leftarrow y$;
16. **end**
17. **end**

Алгоритм \mathcal{A} начинает работу со случайной начальной битовой строки x . Каждая итерация алгоритма состоит из получения случайной реализации СВ $\lambda(n)$ для размера популяции λ , затем следует фаза мутации, фаза кроссинговера и фаза отбора. На этапе мутации (строки 4–8) после получения реализации ℓ с распределением $\text{Bin}(n, p)$, создается λ потомков от x внесением изменений в ℓ случайно выбранных битов в каждом из них. Из этих новых λ особей выбирается одна с наибольшей приспособленностью для участия далее в фазе кроссинговера. Если имеется более одного потомка с максимальной приспособленностью, выбираем одного из них равновероятно. Обозначим выбранного потомка через x' . В фазе кроссинговера (строки 10–12) создается λ новых потомков от родительской особи x и победителя x' из фазы мутации. Из этих λ потомков выбирается строка с наибольшей приспособленностью. Если таких несколько, то выбираем равновероятно одну из них (строки, совпадающие с родительским решением x , при этом не рассматриваются). На этапе отбора (строка 14) родительский x заменяется особью-победителем фазы кроссинговера y , если приспособленность y не ниже, чем приспособленность x . В алгоритме \mathcal{A} , как и во многих других упомянутых выше алгоритмах, не указан критерий останова. Это связано с тем, что при теоретическом исследовании нас главным образом будет интересовать время первого достижения оптимума. В практическом применении алгоритма \mathcal{A} естественно будет указать критерий останова.

2 Основные свойства алгоритма

Обозначим через $\ell_{\lambda(n)}(s)$ число итераций алгоритма 1 из [5], стартовой с решения $x \in Z_s$, до первого улучшения функции приспособленности при фиксированных n и $\lambda(n)$, а вероятность улучшения на первой итерации – через $p_{\lambda(n)}(s) = \mathbf{P}(\ell_{\lambda(n)}(s) = 1)$. Далее мы исследуем алгоритм \mathcal{A} с функцией приспособленности ONEMAX, а вероятности $p_{\lambda(n)}(s)$ совпадают для всех $x \in Z_s$. Отметим, что при случайном выборе $\lambda(n)$ СВ $\ell_{\lambda(n)}(s)$ не несет смысловой нагрузки, а важно только событие $\{\ell_{\lambda(n)}(s) = 1\}$.

Приведем лемму 7 из [5] с фиксированным (не случайным) для каждого n значением λ при параметре мутации $p = \lambda/n$ и параметре кроссинговера $c = \lambda^{-1}$.

Лемма 1. *Одна итерация алгоритма \mathcal{A} с фиксированным λ в случае функции приспособленности ONEMAX, стартовая с решения $x \in Z_s$, приводит к улучшению текущего значения функции приспособленности с вероятностью, удовлетворяющей неравенству*

$$p_{\lambda}(s) \geq C \left(1 - (1 - s/n)^{\lambda^2/2}\right) \left(1 - e^{-1/8}\right) \quad (2)$$

для некоторой не зависящей от n постоянной $C > 0$.

Далее без дополнительных комментариев для не зависящих от n постоянных будем использовать обозначения $C > 0$ и $c > 0$ с индексами или без них. Эти постоянные могут быть взаимосвязаны, но эти связи нас не очень интересуют, а принципиально только наличие этих положительных постоянных. В некоторых случаях для упрощения изложения для различных постоянных мы будем использовать символы C^* и c^* без указания их явного вида и даже в одном уравнении в разных частях равенства эти значения могут быть различными.

Неравенство (2) с учетом оценки

$$1 - (1 - p)^{\lambda} \geq \frac{\lambda p}{1 + \lambda p}, \quad \forall p \in (0, 1), \quad \lambda > 0,$$

из леммы 2 [2] запишем в виде

$$p_{\lambda(n)}(s) \geq C_1 \frac{0.5\lambda^2(n)s/n}{1 + 0.5\lambda^2(n)s/n}, \quad (3)$$

в частности, неравенство (3) можно записать в виде

$$p_{\lambda(n)}(s) \geq C_2 \lambda^2(n)s/n, \quad \text{при } \lambda^2(n)s/n < 1, \quad (4)$$

$$p_{\lambda(n)}(s) \geq C_3, \quad \text{при } \lambda^2(n)s/n \geq 1. \quad (5)$$

Пусть $\lambda(n)$ случайно. Формально на каждой итерации с номером t реализуется СВ $\lambda_t(n)$ с тем же распределением, что и у $\lambda(n)$. Учитывая, что зависимость от t отсутствует мы этот индекс опускаем. Воспользовавшись формулой полной вероятности (усредняя по $\lambda(n)$) обозначим

вероятность улучшения за одну итерацию через $p_n(s) = \mathbf{E}_{\lambda(n)} p_{\lambda(n)}(s) = \mathbf{P}(\ell_n(s) = 1)$, где $\ell_n(s)$ – число итераций до улучшения функции приспособленности. Формально в определении вероятности $p_n(s)$, не зависящей от номера итерации t , событие $\{\ell_n(s) = 1\}$ зависит от этого номера t , а $\ell_n(s)$ равно количеству неудачных итераций в серии итераций с появлением первой удачной. Подсчет количества итераций ведется с начального индивидуума или после очередного увеличения функции приспособленности.

Вычислим среднее $\mathbf{E}\ell_n(s)$. Случайная величина $\ell_n(s)$ имеет геометрическое распределение с параметром $p_n(s)$. По определению имеем

$$\begin{aligned} \mathbf{P}(\ell_n(s) = k) &= (1 - p_n(s))^{k-1} p_n(s), \quad k \in \mathbb{N}, \\ \mathbf{E}\ell_n(s) &= (1 - p_n(s))p_n^{-1}(s) + 1 = p_n^{-1}(s). \end{aligned} \quad (6)$$

Пусть для некоторых фиксированных $m_0, m_1 \in \mathbb{N}$ существует не зависящая от n постоянная $c > 0$ такая, что при всех $n \in \mathbb{N} \setminus \mathbb{N}_{m_1}$ верны неравенства

$$\mathbf{P}(\lambda(n) \in \mathbb{N}_{m_0}) \geq c. \quad (7)$$

Достаточным условием для выполнения неравенства (7) является равномерная по n ограниченность $\mathbf{E}\lambda(n)$, т. е. выполнение условий $\mathbf{E}\lambda(n) \leq c_1$, $\forall n \in \mathbb{N}$, для некоторого $0 < c_1 < \infty$. Тогда по неравенству Маркова для неотрицательных случайных величин $\lambda(n)$ при любом $c_0 > c_1$

$$\mathbf{P}(\lambda(n) < c_0) \geq 1 - \mathbf{E}\lambda(n)/c_0 \geq 1 - c_1/c_0 > 0.$$

Условия для выполнения неравенства (7) будем формулировать так: при достаточно больших n . Очевидно, что если неравенство (7) верно при некоторых фиксированных m_0 и m_1 , то оно верно и при замене этих значений на любые фиксированные значения, превосходящие их.

Оценим дробь из правой части оценки (3) при $\lambda(n) \in \mathbb{N}_{m_0}$. Если $s \leq 0.5nm_0^{-2}$, то $0.5\lambda^2(n)s/n \leq 0.25$ и справедлива оценка

$$\frac{0.5\lambda^2(n)s/n}{1 + 0.5\lambda^2(n)s/n} \geq 0.4\lambda^2(n)s/n \geq 0.4sn^{-1}.$$

В противном случае $s > 0.5nm_0^{-2}$ и $(0.5\lambda^2(n)s/n)^{-1} < 4$ и справедлива оценка

$$\frac{0.5\lambda^2(n)s/n}{1 + 0.5\lambda^2(n)s/n} > 0.2 = 0.2sn^{-1}s^{-1}n \geq 0.2sn^{-1}.$$

При выполнении неравенства (7) в силу двух последних оценок и соотношения (3) при всех $n \in \mathbb{N} \setminus \mathbb{N}_{m_1}$ и $s \in \mathbb{N}_n$ выполняются неравенства

$$p_n(s) \geq \mathbf{E}_{\lambda(n)}\{p_{\lambda(n)}(s); \lambda(n) \in \mathbb{N}_{m_0}\} \geq \frac{0.2C_1s}{n}\mathbf{P}(\lambda(n) \in \mathbb{N}_{m_0}) = \frac{C_4s}{n}. \quad (8)$$

Лемма 2. В случае функции приспособленности ONEMAX среднее число итераций алгоритма \mathcal{A} , стартового с решения $x \in Z_s$, до улучшения целевой функции при выполнении условия (7) удовлетворяет неравенству

$$\mathbf{E}l_n(s) \leq C_4^{-1}n/s, \quad s \in \mathbb{N}_n, \quad (9)$$

где C_4 определено в соотношении (8).

Доказательство. Оценка (9) следует из соотношений (6) и (8). \square

Пусть $\tau_i(n)$ – количество итераций до первого попадания в x^* при условии, что процесс начинается из индивидуума $x^{(0)} \in Z_i$. Если начальное решение выбирать из популяции равномерно, то

$$\mathbf{P}(x^{(0)} \in Z_i) = C_n^i 2^{-n}, \quad \mathbf{E}\tau_i(n) \leq \sum_{s=1}^i \mathbf{E}l_n(s) = \sum_{s=0}^i \mathbf{E}l_n(s). \quad (10)$$

Пусть $\tau(n)$ – количество итераций до первого попадания в x^* из случайно выбранного индивидуума $x^{(0)}$. По формуле полной вероятности верно представление

$$\begin{aligned} \mathbf{E}\tau(n) &= 2^{-n} \sum_{i=0}^n C_n^i \mathbf{E}\tau_i(n) \leq 2^{-n} \sum_{i=0}^n C_n^i \sum_{s=0}^i \mathbf{E}l_n(s) \\ &= 2^{-n} \sum_{s=1}^n \mathbf{E}l_n(s) \sum_{i=s}^n C_n^i = 2^{-n} \sum_{s=1}^{n\epsilon} \mathbf{E}l_n(s) \sum_{i=s}^n C_n^i \\ &\quad + 2^{-n} \sum_{s=n\epsilon+1}^n \mathbf{E}l_n(s) \sum_{i=s}^n C_n^i \leq \sum_{s=1}^{n\epsilon} \mathbf{E}l_n(s) + C_5 n, \end{aligned} \quad (11)$$

где $\epsilon \in (0, 1)$ – произвольное фиксированное число и сумма сочетаний, деленная на 2^n , будет суммой вероятностей, что не превосходит 1, а $\mathbf{E}l_n(s) \leq C_4^{-1}\epsilon^{-1} =: C_5$ при $n\epsilon + 1 \leq s \leq n$. Здесь и далее в суммах по подмножествам натуральных чисел и функций от натурального аргумента s не обязательно целыми пределами или значениями считаем, что эти величины равны ближайшему к ним меньшему целому числу.

3 Среднее число итераций до попадания в оптимум

3.1. Правильно меняющиеся функции. Приведем несколько определений и свойств правильно меняющихся функций. Описанию этих функций посвящена, например, монография [23] и §9 из главы VIII в [24].

Определение 1. Измеримая функция $g(v) > 0$, определенная при достаточно больших $v \in \mathbb{R}^+$ или $v \in \mathbb{N}$ называется **правильно меняющейся на бесконечности** с показателем $\alpha \in \mathbb{R}$, если при любом фиксированном $c \in \mathbb{R}^+$ выполняется условие

$$\lim_{v \rightarrow +\infty} \frac{g(cv)}{g(v)} \rightarrow c^\alpha, \quad (12)$$

где cv следует заменить на $[cv]$ – целую часть числа в случае $v \in \mathbb{N}$.

В случае $\alpha = 0$ функция называется **медленно меняющейся на бесконечности**.

Свойство правильного изменения асимптотическое и функцию $g(v) > 0$ не обязательно определять на любом начальном отрезке полуоси. Правильно меняющиеся функции при $v \in \mathbb{N}$ и $v \in \mathbb{N}$ будем называть правильно меняющимися последовательностями. (Существует и другое определение правильно меняющихся последовательностей при $c \in \mathbb{N}$ в соотношении (12), но во избежание излишних усложнений мы ограничиваемся простым случаем.)

Произвольная правильно меняющаяся на бесконечности функция $g(v)$, $v \in \mathbb{R}^+$, будет асимптотически эквивалентной правильно меняющейся на бесконечности ступенчатой функции $g([v])$, которую можно интерпретировать как правильно меняющуюся на бесконечности последовательность $g(z)$, $z \in \mathbb{N}$.

Очевидно, что верно и обратное утверждение.

Лемма 3. *Для правильно меняющейся на бесконечности последовательности $g(z)$, $z \in \mathbb{N}$, существует правильно меняющаяся на бесконечности функция $\hat{g}(v)$, $v \in \mathbb{R}^+$, такая, что $g([v]) = \hat{g}([v])$ при достаточно больших $v \in \mathbb{R}^+$.*

Далее не будем различать обозначения для последовательностей и функций, задаваемых друг через друга, т. е. отождествим символы \hat{g} и g .

Определение 2. *Измеримая функция $g(v) > 0$, определенная при достаточно малых $v \in \mathbb{R}^+$, называется **правильно меняющейся в нуле справа** с показателем $\alpha \in \mathbb{R}$, если при любом фиксированном $c \in \mathbb{R}^+$ выполняется условие*

$$\lim_{v \rightarrow +0} \frac{g(cv)}{g(v)} \rightarrow c^\alpha$$

В случае $\alpha = 0$ функция называется **медленно меняющейся в нуле справа**.

Последнее определение можно сформулировать для любой точки $a \in \mathbb{R}$, как в одностороннем, так и в двухстороннем варианте, заменив $g(v) > 0$ из определения 2 на $g(v - a) > 0$ и сходимость в том или ином смысле $v - a$ к нулю.

Приведем несколько примеров правильно меняющихся функций. Этот класс является обобщением семейства степенных функций $g(v) = v^\alpha$, $v \in \mathbb{R}^+$, и последовательностей при $v \in \mathbb{N}$.

Функции v^α , $v \in \mathbb{R}^+$, при любом $\alpha \in \mathbb{R}$ являются правильно меняющимися в 0 и на бесконечности.

Функции $|\ln v|$, $|\ln |\ln v||$ и их любые степени являются медленно меняющимися в 0 и на бесконечности.

Если медленно меняющуюся функцию в 0 или на бесконечности обозначить $\ell(v)$, то при любом $\alpha \in \mathbb{R}$ функция $g(v) = v^\alpha \ell(v)$ будет правильно меняющейся в нуле или на бесконечности с показателем α . При этом все правильно меняющиеся функции представимы только в данном виде.

3.2. Обобщение условий на численность популяции и настройки мутации. Следующее определение задает условия на распределение численности популяции $\lambda(n)$ (а значит и распределение параметра мутации $p = \lambda(n)/n$), позволяющие обобщить предположения из [2].

Определение 3. Будем говорить, что для случайной последовательности $\lambda(n)$ выполняются условия:

- \mathcal{A}_2^c , если при всех $n \in \mathbb{N}$ второй момент удовлетворяет неравенству $\mathbf{E}\lambda^2(n) < C$ при некоторой постоянной $C > 0$.

- \mathcal{A}_2^∞ , если $\mathbf{E}\lambda^2(n) \geq \psi(n) = \mathcal{L}(u_n) \rightarrow \infty$, где $\mathcal{L}(m)$ – не зависящая от n правильно меняющаяся с показателем $3 - \beta \in (0, 2)$ последовательность при $m \rightarrow \infty$, $u_n \leq n/2$, и $u_n \rightarrow \infty$ – правильно меняющаяся при $n \rightarrow \infty$ последовательность. Кроме этого, для некоторой не зависящей от n постоянной $C > 0$ и любых $b \in \mathbb{N}_{u_n} \setminus \mathbb{N}_{m_0}$ (при достаточно больших n) выполняется неравенство

$$\mathbf{P}(b/2 \leq \lambda(n) \leq b) \geq Cb^{-2}\mathcal{L}(b). \quad (13)$$

Целью данной работы является обобщение теорем 5 и 6 из [2] на более широкие классы распределений серий случайных величин $\lambda(n)$, $n \in \mathbb{N}$. Приведем явный вид распределений из [2]

$$\mathbf{P}(\lambda(n) = k) = p_{n,k} = C_{\beta, u_n} k^{-\beta}, \quad k \in \mathbb{N}_{u_n}, \quad (14)$$

где $u_n \rightarrow \infty$ при $n \rightarrow \infty$ и $C_{\beta, u_n} = \sum_{k=1}^{u_n} k^{-\beta}$. Мы рассматриваем только случай $\beta > 1$. В силу сходимости ряда $\sum_{k=1}^{\infty} k^{-\beta}$ последовательность C_{β, u_n} асимптотически постоянна. С точки зрения правильно меняющихся функций, вероятности $p_{n,k}$ из представления (14) описываются в терминах правильно меняющихся функций с показателем $-\beta$, в которых медленно меняющийся сомножитель является асимптотически постоянной функцией от u_n . Второй момент $\mathbf{E}\lambda^2(n) = C_{\beta, u_n} \sum_{k=1}^{u_n} k^{2-\beta}$, как показано в [2], имеет порядок $u_n^{3-\beta}$, т.е. будет правильно меняющейся функцией с показателем $3 - \beta$ при $\beta < 3$. Этим мотивируется выбор параметра $3 - \beta$ в определении \mathcal{A}_2^∞ .

Условие \mathcal{A}_2^c означает, что второй момент случайной величины $\lambda(n)$ равномерно ограничен и нет других ограничений на $p_{n,k}$, что реализуется при $\beta > 3$ в условии (14).

Условие \mathcal{A}_2^∞ означает, что правильно меняющаяся на бесконечности функция $\mathcal{L}(v)$ имеет вид $\mathcal{L}(v) = v^{3-\beta}\ell(v)$, $v \in \mathbb{R}^+$, где $\ell(v)$ медленно меняется на бесконечности. Условие (13) можно записать в виде

$$\mathbf{P}(b/2 \leq \lambda(n) \leq b) \geq Cb^{1-\beta}\ell(b).$$

В терминах допущений (14) эти условия выполняются при $1 < \beta < 3$ и $\ell(v)$ асимптотически постоянной, но явный вид $p_{n,k}$ – вероятностей конкретных значений для $\lambda(n)$ не важен, а используются только оценки для их сумм по длинным кускам, которые будут правильно меняющимися на бесконечности функциями с показателем $1 - \beta$.

Очевидно, что $\mathcal{L}(u_n)$, как суперпозиция правильно меняющихся последовательностей, будет правильно меняющейся последовательностью при $n \rightarrow \infty$ с естественно вычисляемым показателем.

В условии \mathcal{A}_2^∞ можно было бы включить и некоторые случаи $\beta = -1, -3$, при которых получаются явные оценки для $\mathbf{E}\tau(n)$, но эти доказательства громоздки и требуют работы с тонкими структурными теоремами для медленно меняющихся функций.

Приведем обобщение утверждения теоремы 5 из [2].

Теорема 1. *Среднее число итераций τ в алгоритме \mathcal{A} с функцией приспособленности ONEMAX до попадания в оптимум оценивается величинами*

$$\mathbf{E}\tau(n) = O(n \ln n), \text{ при условии } \mathcal{A}_2^c; \tag{15}$$

$$\mathbf{E}\tau(n) = O(n) + O\left(\frac{n \ln n}{\psi(n)}\right), \text{ при условии } \mathcal{A}_2^\infty, \tag{16}$$

где $\psi(n)$ из определения 3.

Доказательство. Отметим, что при выполнении условия \mathcal{A}_2^c будет равномерно ограничен по n и первый момент $\mathbf{E}\lambda(n)$. Следовательно, по лемме 2 при выполнении условий \mathcal{A}_2^c верно неравенство

$$\mathbf{E}\ell_n(s) \leq C_4^{-1} \frac{n}{s}. \tag{17}$$

Заметим, что сумма $\sum_{s=1}^u s^{-1}$ от правильно меняющейся ступенчатой функции при $u \rightarrow \infty$ будет асимптотически эквивалентна интегралу $\int_1^u x^{-1} dx$, т.е. при $u \rightarrow \infty$

$$\sum_{s=1}^u s^{-1} \sim \int_1^u x^{-1} dx = \ln u. \tag{18}$$

Оценка (15) следует из соотношений (11), (17) и (18).

Перейдем к доказательству неравенства (16).

Рассмотрим сначала случай $u_n^2 s/n < 1$. При выполнении условия \mathcal{A}_2^∞ оценка (4) приводит к неравенству

$$p_n(s) = \mathbf{E}_{\lambda(n)} p_{\lambda(n)}(s) \geq C^* \mathcal{L}(u_n) s/n = C^* \psi(n) s/n. \tag{19}$$

Потому при $u_n^2 s/n < 1$ и выполнении условия \mathcal{A}_2^∞ из соотношений (6) и (19) получаем

$$\mathbf{E}\ell_n(s) \leq C^* \frac{n}{\psi(n)s}, \tag{20}$$

где формально вместо C^* должно стоять $1/C^*$ с постоянной из оценки (19), но в соответствии с нашими допущениями мы используем для нее то же обозначение C^* .

Данная оценка при $s > n\epsilon$, где $\epsilon \in (0, 1)$ – произвольное фиксированное число, и $u_n^2 \rightarrow \infty$, не применима, но если $s > n\epsilon$, то сумма, содержащая такие слагаемые $\mathbf{E}l_n(s)$ уже оценена сверху величиной Cn в соотношениях (11) и по порядку не может быть улучшена.

Рассмотрим теперь случай $u_n^2 s/n \geq 1$. В данном случае при выполнении условий \mathcal{A}_2^∞ из неравенства (13) при $m_0 < b \leq u_n$ следует оценка

$$\mathbf{E}_{\lambda(n)} \lambda^2(n) \geq \mathbf{E}_{\lambda(n)} \{ \lambda^2(n); \lambda(n) \in [b/2, b] \} \geq C\mathcal{L}(b). \quad (21)$$

В рассматриваемом случае при достаточно больших n выполняются неравенства $m_0 < \sqrt{n/s} \leq u_n$. Поэтому, используя неравенство (21) при $b = \sqrt{n/s}$, $s \in \mathbb{N}_{\epsilon n}$, оценим среднее $p_{\lambda(n)}(s)$ на множестве $\lambda^2(n) < n/s$

$$\mathbf{E}_{\lambda(n)} p_{\lambda(n)}(s) \geq C\mathcal{L}(b)s/n.$$

Отсюда и из соотношения (6) получаем

$$\mathbf{E}l_n(s) \leq C^* \frac{n}{\mathcal{L}(\sqrt{n/s})s}. \quad (22)$$

Неравенства (20) и (22), полученные в двух рассмотренных выше случаях, влекут оценку

$$\sum_{s=1}^{n\epsilon} \mathbf{E}l_n(s) \leq \frac{C^* n}{\psi(n)} \sum_{s=1}^{n/u_n^2-1} s^{-1} + C^* \sum_{s=n/u_n^2}^{n\epsilon} \frac{n}{\mathcal{L}(\sqrt{n/s})s} \quad (23)$$

$$\leq C^* \frac{n}{\psi(n)} \ln(n/u_n^2) + C_1^* n. \quad (24)$$

Поясним неравенство (24). Первая сумма из правой части выражения (23) оценена здесь с использованием соотношения (18). Слагаемые второй суммы из (23) могут быть представлены посредством функции $\mathcal{L}_0(x) := x\mathcal{L}^{-1}(\sqrt{x})$, где $x = n/s$, которая будет правильно меняющейся на бесконечности с показателем $\beta_0 := 1 - 0.5(3 - \beta) = 0.5(\beta - 1)$. При замене переменных $y = x^{-1}$ функция $\mathcal{L}_0(y^{-1})$ будет правильно меняющейся в нуле с показателем $-1 < -\beta_0 < 0$. По функции $\mathcal{L}_0(y^{-1})$ определяем эквивалентную правильно меняющуюся в нуле функцию $\tilde{\mathcal{L}}_0(y^{-1})$ с значениями $\mathcal{L}_0(s/n)$ на множестве $y^{-1} \in [s/n, (s+1)/n]$. Сумма правильно меняющейся последовательности $\mathcal{L}_0(s/n)$ совпадает с интегралом от $\tilde{\mathcal{L}}_0(y^{-1})$, которая постоянна на полуинтервалах длины n^{-1} и будет правильно меняющейся в нуле с показателем $-1 < -\beta_0 < 0$. После замены переменных $u = yn^{-1}$ последний интеграл превращается в

$$n \int_{u_n^{-2}}^{\epsilon} \tilde{\mathcal{L}}_0(u) du \leq n \int_0^{\epsilon} \tilde{\mathcal{L}}_0(u) du,$$

который сходится в нуле при $-1 < -\beta_0 < 0$ по лемме из [24, гл. VIII, §9]. Равномерная ограниченность второй суммы доказана, что завершает доказательство оценки (24).

Если $\psi(n)$ правильно меняется с положительным показателем, то первое слагаемое из (24) будет иметь порядок $o(n)$, а интеграл будет сходиться. В итоге, при выполнении условий \mathcal{A}_2^∞ для функций $\psi(n)$, не являющихся медленно меняющимися, имеем

$$\sum_{s=1}^{n\epsilon} \mathbf{E} \ell_n(s) \leq C^* n. \tag{25}$$

Для медленно меняющихся неограниченно растущих функций u_n по свойству 2° из [23, гл. 1, разд. 1.5] верно соотношение $\ln u_n = o(\ln n)$, что совместно с оценками (11) и (24) доказывает соотношение (16). \square

Оценка (16) для количества итераций до первого попадания в x^* при переходе к произвольным правильно меняющимся на бесконечности последовательностям u_n и $\mathcal{L}(n) = n^{1-\beta} \ell(n)$ качественно отличается от полученной в теореме 5 из [2], где отсутствует второе слагаемое, которое исчезает при $\psi^{-1}(n) \ln n = \mathcal{L}^{-1}(u_n) \ln n = O(1)$. В случае выполнения условий (14) последняя оценка превращается в $u_n^{\beta-3} \ln n = O(1)$, что влечет основное условие теоремы 5 из [2], а именно, $u_n \geq \ln^{1/(3-\beta)} n$.

4 Верхние оценки для времени оптимизации и трудоемкости

Пусть T^{op} – количество вычислительных операций до первого попадания в x^* в модели вычислений RAM с произвольным доступом к памяти [1], где стандартные арифметические операции имеют константную длительность. Распределения случайных величин T и T^{op} зависят от размерности битовых строк индивидуумов n . Поэтому для них далее используем обозначения $T(n)$ и $T^{op}(n)$, соответственно. Обозначим через $T_s(n)$ и $T_s^{op}(n)$ количество вычислений функции приспособленности и вычислительных операций до первого попадания в x^* , соответственно, при условии, что процесс начинается из индивидуума $x^{(0)} \in Z_s$.

Обозначим число вычислений функции приспособленности и количество вычислительных операций в одной итерации, стартующей с индивидуума $x \in Z_s$, заканчивающейся сохранением, возможно другого, индивидуума x из Z_s для следующей операции при фиксированном $\lambda_{i,s}(n)$ через $\mu_{\lambda(n)}(i, s)$ и $\nu_{\lambda(n)}(i, s, n)$. Здесь случайные величины $\lambda_{i,s}(n)$ независимы по i и s и имеют такое же распределение, что и $\lambda(n)$, а индекс i означает порядковый номер старта из индивидуума x (возможно отличного от исходного, но с таким же значением функции приспособленности) с неудачным исходом – без перехода на более высокий уровень. Усреднения по $\lambda_{i,s}(n)$ для $\mu_{\lambda(n)}(i, s)$ и $\nu_{\lambda(n)}(i, s, n)$ обозначим через $\mu_n(i, s) = \mathbf{E}_{\lambda(n)} \mu_{\lambda(n)}(i, s)$ и $\nu_n(i, s) = \mathbf{E}_{\lambda(n)} \nu_{\lambda(n)}(i, s, n)$, соответственно.

В силу одинаковой распределенности СВ $\lambda_{i,s}(n)$ по i последние средние совпадают при различных i с фиксированными s и n .

Для $\mu_{\lambda(n)}(i, s)$ на первом шаге цикла (мутации индивидуума x) производится $\lambda(n)$ вычислений значений функции приспособленности $\mathbf{E}\lambda(n) \leq \mu_n(i, s)$ и на втором, соответствующем кроссинговеру, этих вычислений не более, чем $\lambda(n)$. Поэтому верны ограничения

$$\mu_n(i, s) \leq 2\mathbf{E}\lambda(n). \quad (26)$$

С оценками для $\nu_{\lambda(n)}(i, s)$ сложнее. Это значение зависит от реализации вычислительного алгоритма. Допустим, что существуют функция $\phi(\lambda(n), n)$ такая, что

$$\nu_{\lambda(n)}(i, s, n) \leq \phi(\lambda_{i,s}(n), n).$$

Обозначая через $\phi_n = \mathbf{E}_{\lambda(n)}\phi(\lambda_{i,s}(n), n)$, получаем неравенства

$$\nu_n(i, s) \leq \phi_n. \quad (27)$$

Через $\mu_{\lambda(n)}(s)$ и $\nu_{\lambda(n)}(s, n)$ обозначим случайные количества вычислений функции приспособленности и вычислительных операций в одной итерации стартовой с любого индивидуума x из Z_s , заканчивающейся переходом к другому индивидууму с большим значением функции приспособленности. Их средние по $\lambda(n)$ обозначим через $\mu_n(s) = \mathbf{E}_{\lambda(n)}\mu_{\lambda(n)}(s)$ и $\nu_n(s) = \mathbf{E}_{\lambda(n)}\nu_{\lambda(n)}(s, n)$, соответственно. Для этих средних выполняются неравенства аналогичные (26) и (27)

$$\mu_n(s) \leq 2\mathbf{E}\lambda(n), \quad \nu_n(s) \leq \phi_n. \quad (28)$$

Пусть $\zeta_n(s)$ и $\eta_n(s)$ – число вычислений целевой функции и количество вычислительных операций за время пребывания очередных индивидуумов x в A_s , заканчивающейся переходом к индивидууму с большим значением функции приспособленности. Конкретнее, верны представления

$$\zeta_n(s) = \sum_{i=1}^{\ell_n(s)-1} \mu_n(i, s) + \mu_n(s), \quad \eta_n(s) = \sum_{i=1}^{\ell_n(s)-1} \nu_n(i, s) + \nu_n(s), \quad (29)$$

где сумма равна нулю, если верхний индекс меньше нижнего.

Приведем частный случай утверждения теоремы 2 Колмогорова – Прохорова [25, гл. 4, §4] для неотрицательных случайных величин.

Теорема 2. *Для последовательности неотрицательных независимых случайных величин ξ_i , $i \in \mathbb{N}$, с произвольными средними $\mathbf{E}\xi_i \leq \infty$ и не зависящей от будущего случайной величины $\nu \in \mathbb{N}$ (событие $\{\nu = k\}$ не зависит от ξ_i , $i \in \mathbb{N} \setminus \mathbb{N}_k$) с математическим ожиданием $\mathbf{E}\nu = \sum_{i=1}^{\infty} \mathbf{P}(\nu \geq i) \leq \infty$ верно тождество*

$$\mathbf{E} \sum_{i=1}^{\nu} \xi_i = \sum_{i=1}^{\infty} \mathbf{P}(\nu \geq i) \mathbf{E}\xi_i. \quad (30)$$

Если в теореме 2 все средние $\mathbf{E}\xi_i$ равны, то соотношение (30) превращается в тождество Вальда

$$\mathbf{E} \sum_{i=1}^{\nu} \xi_i = \mathbf{E}\nu \mathbf{E}\xi_1,$$

где средние могут быть и бесконечными.

Соотношения (26), (27), (28) и (29) и теорема 2 позволяют записать неравенства

$$\mathbf{E}\ell_n(s)\mathbf{E}\lambda(n) \leq 2\mathbf{E}\ell_n(s)\mathbf{E}\lambda(n), \quad \mathbf{E}\eta_n(s) \leq \mathbf{E}\ell_n(s)\phi_n. \quad (31)$$

Приведем обобщение утверждения теоремы 6 из [2].

Теорема 3. *Среднее число вычислений функции приспособленности $T(n)$ и числа операций $T^{op}(n)$ в алгоритме \mathcal{A} с функцией приспособленности ONEMAX оценивается сверху величинами*

$$\begin{aligned} \mathbf{E}T(n) &= O(n \ln n), \text{ при условии } \mathcal{A}_2^c; \\ \mathbf{E}T(n) &= O\left(n + \frac{n \ln n}{\psi(n)}\right) \mathbf{E}\lambda(n), \text{ при условиях } \mathcal{A}_2^\infty; \\ \mathbf{E}T^{op}(n) &= O(n \ln n)\phi_{2,n}, \text{ при условии } \mathcal{A}_2^c; \\ \mathbf{E}T^{op}(n) &= O\left(n + \frac{n \ln n}{\psi(n)}\right) \phi_{2,n}, \text{ при условиях } \mathcal{A}_2^\infty. \end{aligned} \quad (32)$$

Доказательство. Применяя формулу полной вероятности по аналогии с (11) и используя неравенств (31) имеем

$$\mathbf{E}T(n) = 2^{-n} \sum_{s=0}^{n-1} C_n^s \mathbf{E}T_s(n) \leq C^* \mathbf{E}\tau(n) \mathbf{E}\lambda(n), \quad (33)$$

$$\mathbf{E}T^{op}(n) = 2^{-n} \sum_{s=0}^{n-1} C_n^s \mathbf{E}T_s^{op}(n) \leq C^* \mathbf{E}\tau(n) \phi_n. \quad (34)$$

Теорема 1 и оценки (33) и (34) влекут утверждение теоремы 3. \square

Нам не важен явный вид среднего $\mathbf{E}\lambda(n)$. Утверждение теоремы 6 [2] соответствует соотношению (32) при выполнении условий

$$\mathbf{E}\lambda(n) \leq C < \infty, \quad \forall n \in \mathbb{N},$$

(или $\beta > 2$), $\psi(n) = u_n^{3-\beta}$ и $u_n \geq \ln^{1/(3-\beta)} n$, что является частным случаем условия $u_n^{\beta-3} \ln n = O(1)$.

Из теоремы 3 непосредственно вытекает

Следствие 1. *Если в алгоритме \mathcal{A} в качестве носителя распределения с.в. $\lambda(n)$ выбрать множество $\mathbb{N}^{(2)} = \{2^\ell, \ell = 0, 1, 2, \dots\}$ и положить $p_{n,k} := C^{(*)}(\beta, u_n)k^{1-\beta}$ при $k \in \mathbb{N}_{u_n}^{(2)}$ и $u_n \in \mathbb{N}^{(2)}$, то в случае функции приспособленности ONEMAX время оптимизации имеет оценку $\mathbf{E}T(n) = O(n)$.*

В следующем разделе проводится экспериментальное сравнение затрат времени ЦПУ до первого получения оптимума функции ONEMAX при генерации с.в. $\lambda(n)$ как указано в следствии 1 или согласно степенному закону с носителем $\{0, 1, 2, \dots, n\}$ из [2].

5 Вычислительный эксперимент

В ходе вычислительного эксперимента рассматривался алгоритм А, где с.в. $\lambda(n)$ генерировалась как указано в следствии 1 (далее - алгоритм А) и алгоритм $(1 + (\lambda, \lambda))$ GA, где $\lambda(n)$ выбирается согласно степенному закону с носителем $\{0, 1, 2, \dots, n\}$ из [2] (далее - алгоритм В). Критерием оптимизации являлась функция ONEMAX, а алгоритм имел настраиваемый параметр $\beta = 2.75$ и верхнюю границу на число потомков $u_n = n$.

За основу был взят программный код на языке Scala, предложенный авторами [2]. Изменения касались только процедуры генерации $\lambda(n)$: в случае алгоритма А ее удается реализовать с трудоемкостью $O(\log \log(u_n))$, тогда как в исходной реализации из [2] для выбора λ требуется $O(\log(u_n))$ операций. С целью сравнения вычислительных затрат по времени ЦПУ до первого получения оптимума были проведены эксперименты при $n = 2^{15}, 2^{16}, 2^{17}, 2^{18}$ и 2^{19} . Вычисления проводились на сервере с процессором AMD EPYC 7502 с использованием 5 независимых параллельных потоков. Для каждого примера оба алгоритма выполнялись 10^5 раз. Среднее арифметическое \hat{T}_A, \hat{T}_B и оценка стандартного отклонения $\hat{\sigma}_A, \hat{\sigma}_B$ для измеренного времени счета (в миллисекундах) приводятся в табл. 1. Диаграммы размаха, построенные по тем же измерениям, приводятся на рис. 1. Прямоугольники здесь покрывают значения от 25 до 75 перцентиля, а вертикальные интервалы вместе с прямоугольниками покрывают 99.3 % наблюдений. Как видно из таблицы и рисунка, алгоритм А имеет преимущество по среднему времени вычислений и более стабильные результаты (меньшее стандартное отклонение в табл. 1 и меньший разброс на рис. 1).

n	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}
\hat{T}_A	93,91	201,65	476,82	1141,03	2790,51
$\hat{\sigma}_A$	225,58	593,88	2005,57	6629,19	22902,68
\hat{T}_B	117,11	266,83	608,05	1492,12	3783,67
$\hat{\sigma}_B$	420,09	1495,47	5007,79	18652,82	64356,55

Таблица 1. Среднее время ЦПУ (мс) до первого получения оптимума и оценка его стандартного отклонения для алгоритмов А и В в зависимости от размеров задачи.

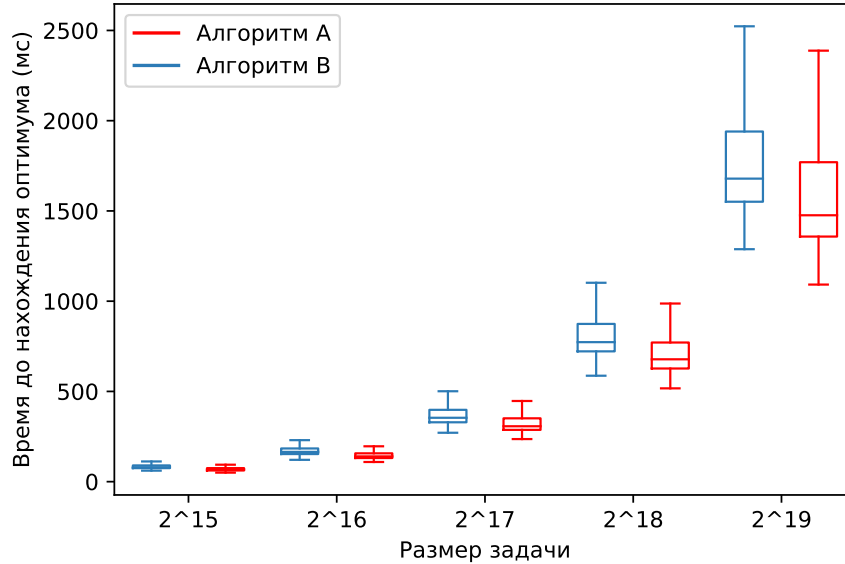


Рис. 1. Диаграммы размаха для времени первого получения оптимума алгоритмов А и В (в миллисекундах) при различных размерностях задачи ONEMAX.

6 Заключение

В работе исследован вопрос о среднем времени получения оптимального решения простой модельной задачи максимизации ONEMAX с помощью известного варианта генетического алгоритма $(1 + (\lambda, \lambda))$ GA. Основной результат данной работы состоит в том, что приведенная верхняя оценка, полученная в статье Антипова, Буздalова и Доерра [2] для $(1 + (\lambda, \lambda))$ GA с оператором быстрой мутации остается справедливой и в более общем случае, когда распределения для численности промежуточной популяции λ и параметра мутации p выбираются из более широкого класса распределений. Проведенный вычислительный эксперимент показал обнадеживающие результаты, позволяющие предположить что предложенный метод случайного выбора численности популяции $\lambda(n)$ окажется полезен на практике.

Рассматриваемая здесь функция ONEMAX имеет только один локальный оптимум, который является и глобальным. Как следует из теоретических результатов, полученных в [18] для $(1 + (\lambda, \lambda))$ GA на модельной функции JUMP с многочисленными локальными оптимумами, использование быстрой мутации на этой функции снимает проблему точного подбора численности популяции и параметра мутации. В связи с

этим, в дальнейших исследованиях имеет смысл рассмотреть возможности ослабления требований к распределению этих параметров в операторе быстрой мутации при оптимизации функции JUMP и других многоэкстремальных функций, в частности, при решении NP-трудных задач псевдодобулевой оптимизации.

References

- [1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, Mass. etc., 1974. Zbl 0326.68005
- [2] D. Antipov, M. Buzdalov, B. Doerr, *Fast mutation in crossover-based algorithms*, *Algorithmica*, **84**:6 (2022), 1724–1761. Zbl 1537.68227
- [3] T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to algorithms. 3rd Edition*, MIT Press, Cambridge, 2009. Zbl 1187.68679
- [4] D.-C. Dang, A.V. Eremeev, P.K. Lehre, X. Qin, *Fast non-elitist evolutionary algorithms with power-law ranking selection*, Proc. GECCO'22—Proceedings of the Genetic and Evolutionary Computation Conference, 1372–1380. MR4536989
- [5] B. Doerr, C. Doerr, F. Ebel, *From black-box complexity to designing new genetic algorithms*, *Theor. Comput. Sci.*, **567** (2015), 87–104. Zbl 1314.68290
- [6] B. Doerr, M. Künnemann, *Optimizing linear functions with the $(1 + \lambda)$ evolutionary algorithm—different asymptotic runtimes for different instances*, *Theor. Comput. Sci.*, **561**, Part A (2015), 3–23. Zbl 1303.68120
- [7] S. Droste, T. Jansen, I. Wegener, *Upper and lower bounds for randomized search heuristics in black-box optimization*, *Theory Comput. Syst.*, **39**:4 (2006), 525–544. Zbl 1103.68115
- [8] T. Jansen, K.A. De Jong, I. Wegener, *On the choice of the offspring population size in evolutionary algorithms*, *Evol. Comput.*, **13** (2005), 413–440.
- [9] P. Erdős, A. Rényi, *On two problems of information theory*, *Publ. Math. Inst. Hung. Acad. Sci., Ser. A*, **8** (1963), 229–243. Zbl 0119.34001
- [10] L.A. Rastrigin, *Statisticheskie metody poiska*, Theoretical Foundations of Technical Cybernetics, **10**, Nauka, Moscow, 1968. MR0301888
- [11] D.E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, 1989. Zbl 0721.68056
- [12] H.K. Hwang, A. Panholzer, N. Rolin, T.H. Tsai, W.M. Chen, *Probabilistic analysis of the $(1+1)$ -evolutionary algorithm*, *Evolutionary Computation*, **26**:2 (2018), 299–345.
- [13] P.K. Lehre, *Negative drift in populations*, In: Schaefer R. (ed.) et al., *Parallel Problem Solving from Nature, PPSN XI. PPSN 2010*. Lecture Notes in Computer Science, **6238**, Springer, Berlin, Heidelberg, 2010, 244–253.
- [14] P.K. Lehre, *Fitness-levels for non-elitist populations*, In *Proceedings of the 2011 Genetic and Evolutionary Computation Conference (GECCO 2011)*, 2075–2082.
- [15] H. Mühlenbein, *How genetic algorithms really work: mutation and hillclimbing*, In *Parallel Problem Solving from Nature 2, PPSN-II*, Elsevier, Brussels, 1992, 15–26.
- [16] S. Droste, T. Jansen, I. Wegener, *On the analysis of the $(1+1)$ evolutionary algorithm*, *Theor. Comput. Sci.*, **276**:1-2 2002, 51–81. Zbl 1002.68037
- [17] C. Witt, *Tight bounds on the optimization time of a randomized search heuristic on linear functions*, *Comb. Probab. Comput.*, **22**:2 (2013), 294–318. Zbl 1258.68183
- [18] B. Doerr, H.P. Le, R. Makhmara, T.D. Nguyen, *Fast genetic algorithms*, In *Proceedings of the Genetic and Evolutionary Computation Conference, July 2017, GECCO 2017*, 777–784.
- [19] B. Doerr, C. Doerr, *Optimal static and self-adjusting parameter choices for the $(1 + (\lambda, \lambda))$ genetic algorithm*, *Algorithmica*, **80**:5 (2018), 1658–1709. Zbl 1391.68100

- [20] A. Eremeev, V. Topchii, *Generalization of the heavy-tailed mutation in the $(1 + (\lambda, \lambda))$ genetic algorithm*, In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, July 2024, GECCO 2024 Companion*, 93–94.
- [21] P.S. Oliveto, C. Witt, *Improved time complexity analysis of the simple genetic algorithm*, *Theor. Comput. Sci.*, **605** (2015), 21–41. Zbl 1330.68120
- [22] P.A. Borisovskii, A.V. Eremeev, *Comparison of certain evolutionary algorithms*, *Autom. Remote Control*, **65**:3 (2004), 357–362. Zbl 1075.90079
- [23] E. Seneta, *Regularly varying functions*, Nauka, Moscow, 1985. Zbl 0563.26002
- [24] W. Feller, *An introduction to probability theory and its applications. Vol II*, Mir, Moscow, 1984. Zbl 0598.60003
- [25] A.A. Borovkov, *Probability theory*, Nauka, Moscow, 1976. Zbl 0463.60001

ANTION VALENTINOVICH EREMEEV
NOVOSIBIRSK STATE UNIVERSITY,
1, PIROGOVA STR.,
NOVOSIBIRSK, 630090, RUSSIA
SOBOLEV INSTITUTE OF MATHEMATICS,
PR. КОПТУГА, 4,
630090, NOVOSIBIRSK, RUSSIA,
Email address: eremeev@ofim.oscsbras.ru

DMITRIY VYACHESLAVOVICH SILAEV
630090, NOVOSIBIRSK, RUSSIA,
NOVOSIBIRSK STATE UNIVERSITY,
1, PIROGOVA STR.,
NOVOSIBIRSK, 630090, RUSSIA
Email address: chuvakeu2001@gmail.com

VALENTIN ALEKSEEVICH ТОПЧИИ
SOBOLEV INSTITUTE OF MATHEMATICS,
PR. КОПТУГА, 4,
630090, NOVOSIBIRSK, RUSSIA,
NOVOSIBIRSK STATE UNIVERSITY,
1, PIROGOVA STR.,
NOVOSIBIRSK, 630090, RUSSIA
Email address: topchij@gmail.com