УДК 004.023 doi: 10.17586/2226-1494-2020-20-5-701-707

ПРИМЕНЕНИЕ МЕТОДА УРОВНЕЙ ПРИСПОСОБЛЕННОСТИ ДЛЯ АНАЛИЗА ДИНАМИКИ РАБОТЫ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

М.В. Буздалов, Д.В. Винокуров

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: dimaoq@gmail.com

Информация о статье
Поступила в редакцию 02.07.20, принята к печати 10.08.20
Язык статьи — русский

Ссылка для цитирования: Буздалов М.В., Винокуров Д.В. Применение метода уровней приспособленности для анализа динамики работы эволюционных алгоритмов // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20. № 5. С. 701–707. doi: 10.17586/2226-1494-2020-20-5-701-707

Аннотация
Предмет исследования. В области теории эволюционных алгоритмов в настоящее время становится актуальным анализ не только времени их работы, но и динамики. Двумя наиболее распространенными методами анализа динамики являются: анализ достижимой приспособленности при ограничении на время работы (fixed-budget analysis) и анализ времени работы, необходимого для достижения заданной цели (fixed-target analysis). До сих пор теоретические исследования систематически выполнялись только для первого из них. Настоящая работа направлена на устранение этого недостатка. **Метод.** Доказана теорема о том, что если для комбинации эволюционного алгоритма и задачи оптимизации ранее были доказаны оценки времени работы с помощью так называемого метода уровней приспособленности, то из необходимых для этого предпосылок автоматически следуют оценки динамики работы эволюционного алгоритма для рассматриваемой задачи. **Основные результаты.** В результате применения данной теоремы получены верхние оценки времени работы, которое необходимо для достижения заданной цели следующих пар алгоритмов и задач: эволюционные алгоритмы семейства $(1 + 1)$ на задачах LeadingOnes и OneMax, эволюционный алгоритм $(\mu + 1)$ на задаче OneMax. Данные оценки повторяют или уточняют существующие результаты, но получают их существенно более простым способом. **Практическая значимость.** Результаты работы позволяют упростить получение оценок динамики работы эволюционных алгоритмов. Подобные оценки являются более содержательным критерием для выбора эволюционного алгоритма с целью решения практической задачи, чем оценка времени нахождения оптимального решения, так как последнее на практике чаще всего недостижимо.

Ключевые слова
эволюционные алгоритмы, анализ времени работы, fixed-target анализ, fixed-budget анализ, метод уровней приспособленности

Благодарности
Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований и Национального центра научных исследований в рамках научного проекта № 20-51-15009.

doi: 10.17586/2226-1494-2020-20-5-701-707

METHOD OF ARTIFICIAL FITNESS LEVELS FOR DYNAMICS ANALYSIS OF EVOLUTIONARY ALGORITHMS

M.V. Buzdalov, D.V. Vinokurov

ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: dimaoq@gmail.com

Article info
Received 02.07.20, accepted 10.08.20
Article in Russian

For citation: Buzdalov M.V., Vinokurov D.V. Method of artificial fitness levels for dynamics analysis of evolutionary algorithms. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2020, vol. 20, no. 5, pp. 701–707 (in Russian). doi: 10.17586/2226-1494-2020-20-5-701-707

Abstract

Subject of Research. Currently, in the theory of evolutionary computation, it becomes relevant to analyze not just the runtime of evolutionary algorithms, but also their dynamics. The two most common methods for dynamics analysis are: fixed-budget analysis, which studies an algorithm reachable fitness in condition of operation time limit, and fixed-target analysis, which studies the time that an algorithm needs to reach some fixed fitness value. Until now, theoretical studies were systematically carried out only for the first type of analysis. The present work is focused on removal of this disadvantage. **Method.** We proved the following theorem: if the bounds on optimization time for some evolutionary algorithm on some problem are already proven using *artificial fitness levels*, than the bounds on this algorithm dynamics on the considered problem derive automatically from the same preconditions. **Main Results.** Using this theorem, we obtain the upper bounds on fixed-target runtime for the following pairs of algorithms and problems: the family of (1 + 1) evolutionary algorithms on LeadingOnes and OneMax functions, and (μ + 1) evolutionary algorithm on OneMax. These bounds either repeat or refine the existing results, but in a much simpler way. **Practical Relevance.** The main practical achievement of this paper is that it simplifies the proving of bounds on the dynamics of evolutionary algorithms. In turn, these bounds could be more meaningful for choosing between different evolutionary algorithms for some problem than the time for reaching the optimum, as the latter is mostly infeasible in practice.

Keywords

evolutionary algorithms, runtime analysis, fixed-target analysis, fixed-budget analysis, fitness-level method

Acknowledgments

This research was financially supported by the Russian Foundation for Basic Research and Centre National de la Recherche Scientifique (project No. 20-51-15009).

Введение

Наиболее часто используемой мерой производительности в теоретическом исследовании эволюционных алгоритмов (ЭА) является число вычислений функции приспособленности, которое необходимо изучаемому алгоритму для достижения оптимального решения рассматриваемой задачи. Данное число для простоты часто называют *временем работы* алгоритма на задаче. Так как это время — случайная величина, чаще всего исследуются: его математическое ожидание [1]; вероятность, что алгоритм закончит работу за заданное время [2], и др. Эти оценки сводят весь процесс оптимизации в одну точку – в момент, когда найдено оптимальное решение, но иногда может понадобиться более детальная информация обо всем ходе работы алгоритма. Мотивацией для этого могут служить следующие соображения.

1. ЭА имеют решение на каждой итерации: если прервать процесс работы, то всегда может быть получено решение некоторого качества, и время оптимизации как мера производительности ничего не говорит об этом решении. На практике же информация обо всем процессе работы может быть очень полезна, например, если заранее неизвестно, как долго будет исполняться алгоритм, чтобы достичь решения определенного качества.
2. При существовании нескольких эвристик (алгоритмов) решения задачи можно в определенные моменты времени переключаться между ними, чтобы повысить эффективность решения этой задачи. Но чтобы определить, когда нужно переключаться, необходим более полный анализ, чем анализ времени достижения оптимума.

Исследование процесса работы целиком, а не только в точке оптимума, довольно редко появляется в работах, посвященных теоретическим исследованиям, хотя давно и успешно применяется в экспериментальных исследованиях [3, 4]. Наиболее часто используются следующие два вида анализа:

1) fixed-budget analysis (FB-анализ) — анализ достижимого качества решения (приспособленности) при ограничении на время работы алгоритма;
2) fixed-target analysis (FT-анализ) — анализ времени работы, необходимого для достижения решения с заданной приспособленностью.

FB-анализ был представлен первым в [5]. В этой работе даны результаты для ряда простых алгоритмов, таких как RLS (Randomized Local Search) и (1 + 1)-ЭА, для самых часто анализируемых функций. Эти результаты использовали подходы, применимость которых ограничена исследуемыми функциями и алгоритмами, и даже для такого простого алгоритма как RLS, оптимизирующего функцию LeadingOnes, результаты являлись довольно трудными для получения и понимания, а для (1 + 1)-ЭА, оптимизирующего ту же функцию, диапазон покрываемого числа вычислений функции приспособленности был значительно меньше среднего числа итераций до достижения оптимума.

В работе [6] представлен первый более общий подход для нахождения оценок FB-анализа, который основан на оценках концентрации результатов FT-анализа. Метод сведения FB-анализа к FT-анализу также использован в работе [7] при рассмотрении поведения алгоритмов на линейных псевдобулевых функциях.

Сложно определить, в какой работе впервые был применен термин «FT-анализ», так как он является натуральным расширением анализа времени работы. Скорее всего, по этой причине авторы соответствующих публикаций не видели необходимости во введении нового термина.

Кроме того, метод уровней приспособленности, часто используемый в классическом анализе, крайне схож с непосредственным поиском результатов FT-анализа. Помимо неявно упоминающих этот анализ работ [6–9], он попал в центр внимания только в [10], при этом исследовались в основном простые функции: LeadingOnes [6, 8, 9, 11], OneMax [7, 9, 10] и BinVal [10].

Сравнивая FB- и FT-анализы, можно сделать вывод, что число работ про первый из них больше, чем про второй, но и это число является небольшим. С точки зрения сложности исследования, выполнение FB-анализа кажется более сложной процедурой, чем выполнение FT-анализа. С позиции разработки динамических алгоритмов, меняющих в процессе работы свои параметры и эвристики, результаты FB-анализа могут быть применены для выбора, основанного на уже выполненном числе вызовов функции приспособленности, тогда как результаты FT-анализа используются для выбора эвристик на основе лучшего найденного решения. С практической точки зрения алгоритмов, использующих последнюю схему действий, значительно больше [12–14].

В настоящей работе рассмотрен FT-анализ как способ исследования всего процесса оптимизации алгоритма. В традиционном анализе времени работы наиболее распространены два метода получения оценок: метод уровней приспособленности и дрейфовый анализ. Ранее ни один из этих методов не изучался в контексте FT-анализа, поэтому в данной работе ставится цель — адаптировать классический метод уровней приспособленности как наиболее натуральный метод получения результатов FT-анализа. С его помощью будут найдены некоторые новые оценки, а также показано, насколько просто можно получить новые результаты FT-анализа из существующих оценок на время оптимизации, полученных с применением метода уровней приспособленности.

Рассматриваемые алгоритмы и функции

В работе полагается, что n — размерность задачи, $f(x)$ — функция приспособленности, p — вероятность мутации (изменения) одного отдельного бита, и используются следующие математические обозначения: e — постоянная Эйлера, H_i — i-ое гармоническое число, а также используется стандартная нотация для обозначения асимптотического поведения функций: $O(f(x))$ означает функцию, растущую не быстрее $f(x)$; $\Omega(f(x))$ — это функция, растущая не медленнее $f(x)$; $o(f(x))$ — функция $g(x)$, такая, что $\lim\limits_{x\to\infty}\frac{|g(x)|}{|f(x)|}=0$.

Рассматриваются $(\mu + 1)$-ЭА, $(1 + 1)$-ЭА и его различные вариации, отличающиеся стратегиями выбора числа бит при мутации и особей при инициализации: число изменяемых бит генерируется из распределения $\mathcal{M}$, а начальные особи получаются согласно распределению $\mathcal{D}$, в котором либо область значений состоит из самой дальней особи, либо с одинаковой вероятностью отдается любая случайная возможная особь. Все эти алгоритмы могут быть представлены в общем виде $(\mu + \lambda)$-ЭА, приведенного на рисунке, где μ — размер родительской популяции (особи, сохраняемые между итерациями алгоритма), а λ — размер дочерней популяции (особи, создаваемые на каждой итерации).

Например, рассматриваемое множество $(1 + 1)$-ЭА может быть представлено с параметризованным распределением $\mathcal{M}$:

- RLS: распределение $\mathcal{M} = 1$;
- $(1 + 1)$-ЭА: распределение $\mathcal{M} = B(n, p)$, где $B(n, p)$ — биномиальное распределение;
- $(1 + 1)_{0\to1}$-ЭА: распределение $\mathcal{M} = \max\{1, B(n, p)\}$;
- $(1 + 1)_{>0}$-ЭА: распределение $\mathcal{M} = [x \sim B(n, p) | x > 0]$.

Также рассматриваются несколько функций приспособленности:

- LeadingOnes — число ведущих единичных бит в особи;
- OneMax — число единичных бит в особи.

Метод уровней приспособленности

В методе уровней приспособленностей все пространство значений функции приспособленности делится на уровни: пространство поиска A разбивается на уровни $A_1 \ldots A_m$, причем для всех особей $a \in A_i$ и $b \in A_{i+1}$ соблюдается $f(a) < f(b)$, где оптимум находится на последнем уровне A_m. Если текущая лучшая особь в популяции имеет приспособленности $f \in A_i$, тогда считается, что эта особь находится на уровне приспособленности A_i.

Есть множество теорем для получения верхних и нижних оценок с использованием метода уровней приспособленности. Здесь для примера приведены две теоремы, с помощью которых дальше будут получены результаты FT-анализа.

Теорема 1. Метод уровней приспособленности, верхняя оценка [15]. Пусть есть разбиение пространства A_i, $1 \leqslant i \leqslant m$ и, пусть известна вероятность p_i перейти с уровня A_i на уровень A_i, $j > i$. Тогда математическое ожидание достижения уровня A_m не превышает

for $i \in \{1, \ldots, \mu\}$ **do**
  $x_i \leftarrow$ случайная особь из $\mathcal{D}$, посчитать $f(x_i)$
end for
$X \leftarrow \{x_1, \ldots, x_\mu\}$ ▹ Множество родительских особей
while true do
  for $i \in \{1, \ldots, \lambda\}$ **do**
    $j \leftarrow$ случайное число, выбранное равновероятно из $\{1, \ldots, \mu\}$
    $\ell \leftarrow$ случайное число, выбранное из $\mathcal{M}$
    $y_i \leftarrow$ изменить случайные ℓ бит в x_j, посчитать $f(y_i)$
  end for
  $Y \leftarrow \{y_1, \ldots, y_\lambda\}$ ▹ Множество дочерних особей
  $X \leftarrow \mu$ лучших особей из $X \cup Y$ предпочитая особь из Y в случае особей с одинаковой приспособленностью
end while

Рисунок. $(\mu + \lambda)$-ЭА с $\mathcal{M}$ и $\mathcal{D}$ в качестве параметров

$$\sum_{i=1}^{m-1} P[\text{алгоритм стартует с } A_i] \sum_{j=i}^{m-1} \frac{1}{p_i} \le \sum_{j=1}^{m-1} \frac{1}{p_i}.$$

Теорема 2. Метод уровней приспособленности, верхняя оценка [16]. Пусть есть разбиение пространства A_i, $1 \le i \le m$, и пусть вероятность перейти с уровня A_i на уровень A_j не меньше, чем $s_i\gamma_{i,j}$, где s_i — это вероятность перейти с A_i на любой уровень выше, $\gamma_{i,j}$ — это вероятность перейти с A_i на A_j при условии, что точно происходит переход с A_i, и $\sum_{j=i+1}^{m} \gamma_{i,j} = 1$. Пусть также есть некоторое χ, $0 \le \chi \le 1$, и $\gamma_{i,j} \le \chi \sum_{k=i}^{m} \gamma_{i,j}$ для всех $1 \le i < j < m$, а также $(1-\chi)s_i \le s_{i+1}$ для всех $1 \le i \le m-2$. Тогда математическое ожидание достижения уровня A_m не превышает

$$\sum_{i=1}^{m-1} P[\text{алгоритм стартует с } A_i]\left(\frac{1}{s_i} + \chi \sum_{j=i+1}^{m-1} \frac{1}{s_i}\right).$$

Данные теоремы нацелены на поиск времени до оптимума, который принадлежит уровню m, но для получения результатов FT-анализа нужен переход к промежуточным значениям.

Теорема 3. Метод уровней приспособленности, переход от верхних границ из теорем 1, 2 к FT-анализу. Пусть есть разбиение пространства A_i, $1 \le i \le m$, целевое значение $k \in A_{m'}$ и $m' < m$. Тогда, если объединить уровни $A_{m'} \ldots A_m$ в один уровень $A_{m'}$, то теоремы 1 и 2 все еще соблюдаются при соответствующем изменении величин, используемых в них.

Доказательство.

При объединении уровней $A_{m'} \ldots A_m$ уровень $A_{m'}$ будет содержать все оптимальные особи с приспособленностью $k' \ge k$. В случае теоремы 1 вероятность перейти на этот объединенный уровень будет равна $\sum_{i=m'}^{m} p_i$; в случае теоремы 2 объединяются значения $\gamma'_{i,m'} \leftarrow \sum_{k=m'}^{m} \gamma_{i,k}$, $\forall i < m'$, а величины s_i и χ не изменяются. Тогда легко заметить, что обе теоремы все так же соблюдаются, если вместо числа уровней приспособленности m подставить новое число уровней m'.

Таким образом, применение представленных теорем совсем не изменяется, и многие другие теоремы о методе уровней могут быть подобным способом адаптированы для FT-анализа. Отсюда следует, что результаты FT-анализа часто могут быть получены без особых усилий из существующих оценок, для которых применялся метод уровней приспособленности. Далее приводятся несколько новых результатов для различных алгоритмов с применением приведенных теорем.

Вариации (1 + 1)-ЭА для задачи LeadingOnes

Теорема 4. Время достижения особи с приспособленностью не меньше k при оптимизации функции LeadingOnes семейством (1 + 1)-ЭА составляет:

— $\dfrac{kn}{2}$ для RLS;

— $\dfrac{(1-p)^{1-k} - (1-p)}{2p^2}$ для (1 + 1)-ЭА;

— $\dfrac{(1-p)^{1-k} - (1-p)}{2p^2}(1-(1-p)^n)$ для $(1 + 1)_{>0}$-ЭА;

— $\dfrac{1}{2}\sum_{i=0}^{k-1} \dfrac{1}{p(1-p)^i + \frac{1}{n}(1-p)^n}$ для $(1 + 1)_{0\to1}$-ЭА.

Доказательство.

Для использования теорем 2 и 3 нужно определить промежуточные величины: учитывая, что целевой уровень — это k, получаем $\gamma_{i,j} = 2^{i-j}$ для всех $j < k$, $\gamma_{i,k} = 2^{i-k+1}$, иначе, и $\chi = 1/2$ [16]. Тогда можно применить теорему 2:

$$\sum_{i=0}^{k-1} P[\text{начальная особь имеет приспособленность } i] \times$$
$$\times\left(\frac{1}{q_i} + \frac{1}{2}\sum_{j=i+1}^{k-1}\frac{1}{q_i}\right) = \frac{1}{2}\sum_{i=0}^{k-1}\frac{1}{q_i}.$$

Правая половина получается из первой при раскрытии суммы и перестановке членов [12], а также, если заметить, что начальная особь имеет приспособленность i с вероятностью $P = 2^{-i-1}$.

Завершается доказательство поиском вероятности q изменить ровно один самый левый нулевой бит и не изменить ни одного бита до него с учетом того, что текущая приспособленность i:

— RLS: $q_i = 1/n$;

— (1 + 1)-ЭА: $q_i = (1-p)^i p$;

— $(1 + 1)_{0\to1}$-ЭА: $q_i = (1-p)^i p + 1/n(1-p)^n$;

— $(1 + 1)_{>0}$-ЭА: $q_i = (1-p)^i p(1-(1-p)^n)$.

Такие же результаты для (1 + 1)-ЭА и $(1 + 1)_{>0}$-ЭА уже были получены ранее [11], но в представленном здесь виде не фиксируется оптимизирующий алгоритм, поэтому конечная формула представляет собой более общий вид из-за наличия вероятности q.

Вариации (1 + 1)-ЭА на задаче OneMax

На таких функциях как OneMax, для которых более вероятен длинный переход на последующие уровни приспособленности, исследования затрудняются, например, появляется нижняя граница на целевую приспособленность, которую нужно достичь.

Теорема 5. Время достижения особи с приспособленностью не ниже $k \ge n/2 + \sqrt{n}\ln n$ при оптимизации OneMax семейством (1 + 1)-ЭА не больше:

$\dfrac{1}{q}(H_n - H_{n-k})$, если алгоритм начинает свою работу с самой дальней особи;

$\dfrac{1}{q}\left(H_{n/2} - H_{n-k} - \dfrac{1-o(1)}{2n}\right)$, если алгоритм начинает свою работу со случайной особи, где вероятность q — это вероятность изменить ровно один бит в особи, которая для рассматриваемых алгоритмов равна:

— RLS: $q = 1/n$;

— (1 + 1)-ЭА: $q = p(1-p)^{n-1}$;

— $(1 + 1)_{0\to1}$-ЭА: $q = p(1-p)^{n-1} + \dfrac{(1-p)^n}{n}$;

— $(1 + 1)_{>0}$-ЭА: $q = \dfrac{p(1-p)^{n-1}}{1-(1-p)^n}$.

Доказательство.

Пусть алгоритм начинает свою работу с особи с приспособленностью i с вероятностью s_i, и известна вероятность q изменить ровно один бит. Тогда вероятность p_i перейти на уровень выше $p_i \geq q(n-i)$, и, используя теоремы 1 и 3, получаем:

$$\sum_{i=0}^{k-1} s_i \sum_{j=i}^{k-1} \frac{1}{q(n-j)} = \frac{1}{q}\sum_{i=0}^{k-1} s_i (H_{n-i} - H_{n-k}) \leq \frac{H_n - H_{n-k}}{q}.$$

Данное выражение уже подходит для верхней границы с инициализацией худшей особи. Для случайной же особи вероятность $s_i = \binom{n}{i}/2^n$, и тогда

$$\sum_{i=0}^{k-1} \frac{\binom{n}{i} H_{n-i} - H_{n-k}}{q2^n} \leq \sum_{i=0}^{n} \frac{\binom{n}{i} H_{n-i}}{q2^n} - \sum_{i=0}^{n} \frac{\binom{n}{i} H_{n-k}}{q2^n} +$$

$$+ \frac{1}{q}\sum_{i=k}^{n} \frac{\binom{n}{i} H_{n-k} - H_{n-i}}{2^n} = \frac{1}{q}\left(H_{n/2} - H_{n-k} - \frac{(1-o(1)}{2^n}\right) +$$

$$+ \frac{1}{q}\sum_{i=k}^{n} \frac{\binom{n}{i} H_{n-k} - H_{n-i}}{2^n} = \frac{1}{q}\left(H_{n/2} - H_{n-k} - \frac{(1-o(1)}{2^n}\right) +$$

$$+ \frac{1}{q}\frac{O(\ln(n))}{n^{\Omega(\ln n)}},$$

где третье выражение берется из [17, 18]:

$$\sum_{i=0}^{n} \frac{\binom{n}{i} H_i}{2^n} = H_n - \sum_{k=1}^{n} \frac{1}{k2^k} = H_n - \ln 2 + O(2^{-n}) =$$

$$= H_{n/2} - \frac{1-o(1)}{2n},$$

а последнее выражение получается, если заметить, что $H_{n-k} - H_{n-i} = O(\ln n)$, и для всех $k \geq n/2 + \sqrt{n}\ln n$, $i \geq k$ соблюдается $\frac{1}{2^n}\sum_{i=k}^{n}\binom{n}{i} = n^{-\Omega(\ln n)}$, что следует из границ Чернова. Таким образом, самое последнее слагаемое является экспоненциально малым, и, в частности, составляет $o(1/n)$, что полностью исчезает в выражении $\frac{1-o(1)}{2n}$.

(μ + 1)-ЭА на задаче OneMax

Последним полученным результатом является верхняя граница для функции OneMax на (μ + 1)-ЭА, адаптированная из существующего доказательства на время оптимизации.

Теорема 6. Пусть $\mu = poly(n)$ и $b = \lfloor n(1-1/\mu)\rfloor$. Тогда время достижения особи с приспособленностью $k > 0$ при оптимизации функции OneMax на (μ + 1)-ЭА будет не больше

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

$$+ \frac{\mu}{p(1-p)^{n-1}} \begin{cases} \frac{k}{n}, & k < b+2 \\ \frac{b+1}{n} + \frac{1}{\mu}(H_{n-b-1} - H_{n-k}), & \text{в ином случае} \end{cases}.$$

Доказательство.

Это доказательство следует утверждениям, аналогичным [8]: предполагается, что на каждом уровне L алгоритм создает $R = \min\{\mu, n/(n-L)\}$ одинаковых особей (реплик) лучшей особи и только затем пытается увеличить ее приспособленность. Также предполагается, что не происходит увеличения приспособленности больше, чем на один.

Пусть на какой-то итерации родительская популяция содержит $i < R$ лучших особей, тогда вероятность создать еще одну реплику равна $(1-p)^n i/\mu$, и время до создания R копий равно

$$\frac{\mu}{(1-p)^n}\sum_{i=1}^{R-1}\frac{1}{i} \leq \frac{\mu}{(1-p)^n}\sum_{i=1}^{\frac{n}{n-L}-1}\frac{1}{i} \leq \frac{\mu}{(1-p)^n}\ln\frac{en}{n-L},$$

тогда общее время, которое (μ + 1)-ЭА проведет за созданием копий

$$T_r \leq \sum_{L=0}^{k-1}\frac{\mu}{(1-p)^n}\ln\frac{en}{n-L} = \frac{\mu}{(1-p)^n}\left(k\ln en + \sum_{x=n-k+1}^{n}\ln\frac{1}{x}\right) \leq$$

$$\leq \frac{\mu}{(1-p)^n}\left(k\ln en + k - 1 + (n-k)\ln(n-k+1) - n\ln n\right) =$$

$$= \frac{\mu}{(1-p)^n}\left(2k - 1 - (n-k)\ln\frac{n}{n-k+1}\right),$$

где $\sum_{x=a}^{b}\ln\frac{1}{x} \leq \ln\frac{1}{a} + \int_a^b \ln\frac{1}{x}dx = \ln\frac{1}{a} + \left[x + x\ln\frac{1}{x}\right]_a^b$.

Что касательно увеличения приспособленности лучшей особи, то вероятность будучи на уровне L создать особь на уровне $L + 1$ из одной из R существующих реплик не меньше

$$R(n-L)p(1-p)^{n-1}/\mu \geqslant \min\{\mu(n-L), n\}p(1-p)^{n-1}/\mu,$$

и тогда можно применить теоремы 1 и 3 для расчета времени, потраченного на переходы с одного уровня приспособленности на следующий

$$T_f \leq \sum_{i=0}^{k-1}\frac{1}{p_i} = \frac{\mu}{p(1-p)^{n-1}}\sum_{i=0}^{k-1}\frac{1}{\min\{\mu(n-i), n\}}.$$

В отличие от доказательства [8], здесь рассматриваются разные случаи минимизации выражения в знаменателе последней дроби. В зависимости от того, как целевое значение k соотносится к числу $b = \lfloor n(1-1/\mu)\rfloor$, время T_f станет

$$T_f \leq \begin{cases} \frac{\mu}{np(1-p)^{n-1}}k, & k < b+2 \\ \frac{\mu}{np(1-p)^{n-1}}(b+1) + \frac{H_{n-b-1} - H_{n-k}}{p(1-p)^{n-1}}, & \text{в ином случае} \end{cases}.$$

Это завершает доказательство, если заметить, что общее время работы с учетом инициализации родительских μ особей $T_k = \mu + T_r + T_f$.

Заключение

В данной работе показана проблема необходимости более полного, чем поиск времени оптимизации, анализа времени работы эволюционных алгоритмов. Для этого выбран анализ времени работы, необходимого для достижения решения с заданной приспособленностью (FT-анализ), и показано, что он может быть проведен с использованием метода уровней приспособленности без особых проблем. Представлен ряд новых результатов FT-анализа с использованием этого метода, а также продемонстрировано получение оценок FT-анализа из существующих доказательств на время оптимизации, которые использовали метод уровней приспособленности. В дальнейшем планируется похожим образом исследовать дрейфовый анализ в контексте FT-анализа.

Литература

1. Lengler J. Drift analysis // Theory of Evolutionary Computation: Recent Developments in Discrete Optimization. Springer, 2020. P. 89–131. doi: 10.1007/978-3-030-29414-4_2
2. Doerr B. A tight runtime analysis for the cGA on jump functions: EDAs can cross fitness valleys at no extra cost // Proc. Genetic and Evolutionary Computation Conference (GECCO). 2019. P. 1488–1496. doi: 10.1145/3321707.3321747
3. Hansen N., Auger A., Ros R., Mersmann O., Tušar T., Brockhoff D. COCO: A platform for comparing continuous optimizers in a black-box setting // Optimization Methods and Software. 2020. in press. doi: 10.1080/10556788.2020.1808977
4. Doerr C., Wang H., Ye F., van Rijn S., Bäck T. IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics [Электронный ресурс]. URL: https://arxiv.org/abs/1810.05281 (дата обращения: 27.06.2020).
5. Jansen T., Zarges C. Performance analysis of randomised search heuristics operating with a fixed budget // Theoretical Computer Science. 2014. V. 545. P. 39–58. doi: 10.1016/j.tcs.2013.06.007
6. Doerr B., Jansen T., Witt C., Zarges C. A method to derive fixed budget results from expected optimisation times // Proc. 15th Genetic and Evolutionary Computation Conference (GECCO 2013). 2013. P. 1581–1588. doi: 10.1145/2463372.2463565
7. Lengler J., Spooner N. Fixed budget performance of the (1+1) EA on linear functions // Proc. 13th Foundations of Genetic Algorithms (FOGA). 2015. P. 52–61. doi: 10.1145/2725494.2725506
8. Witt C. Runtime analysis of the (μ+1) EA on simple pseudo-Boolean functions // Evolutionary Computation. 2006. V. 14. N 1. P. 65–86. doi: 10.1162/106365606776022751
9. Pinto E.C., Doerr C. Towards a more practice-aware runtime analysis of evolutionary algorithms [Электронный ресурс]. URL: https://arxiv.org/abs/1812.00493 (дата обращения: 27.06.2020).
10. Vinokurov D., Buzdalov M., Buzdalova A., Doerr B., Doerr C. Fixed-target runtime analysis of the (1+1) EA with resampling // Proc. Genetic and Evolutionary Computation Conference Companion (GECCO'19). 2019. P. 2068–2071. doi: 10.1145/3319619.3326906
11. Doerr B. Analyzing randomized search heuristics via stochastic domination // Theoretical Computer Science. 2019. V. 773. P. 115–137. doi: 10.1016/j.tcs.2018.09.024
12. Böttcher S., Doerr B., Neumann F. Optimal fixed and adaptive mutation rates for the LeadingOnes problem // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2010. V. 6238. P. 1–10. doi: 10.1007/978-3-642-15844-5_1
13. Badkobeh G., Lehre P.K., Sudholt D. Unbiased black-box complexity of parallel search // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2014. V. 8672. P. 892–901. doi: 10.1007/978-3-319-10762-2_88
14. Doerr B., Doerr C, Ebel F. From black-box complexity to designing new genetic algorithms // Theoretical Computer Science. 2015. V. 567. P. 87–104. doi: 10.1016/j.tcs.2014.11.028
15. Wegener I. Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions // International Series in Operations Research & Management Science. 2003. V. 48. P. 349–369. doi: 10.1007/0-306-48041-7_14
16. Sudholt D. A new method for lower bounds on the running time of evolutionary algorithms // IEEE Transactions on Evolutionary Computation. 2013. V. 17. N 3. P. 418–435. doi: 10.1109/TEVC.2012.2202241
17. Doerr B., Doerr C. The impact of random initialization on the runtime of randomized search heuristics // Algorithmica. 2016. V. 75. N 3. P. 529–553. doi: 10.1007/s00453-015-0019-5

References

1. Lengler J. Drift analysis. *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*. Springer, 2020, pp. 89–131. doi: 10.1007/978-3-030-29414-4_2
2. Doerr B. A tight runtime analysis for the cGA on jump functions: EDAs can cross fitness valleys at no extra cost. *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, 2019, pp. 1488–1496. doi: 10.1145/3321707.3321747
3. Hansen N., Auger A., Ros R., Mersmann O., Tušar T., Brockhoff D. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 2020, in press. doi: 10.1080/10556788.2020.1808977
4. Doerr C., Wang H., Ye F., van Rijn S., Bäck T. *IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics.* Available at: https://arxiv.org/abs/1810.05281 (accessed: 27.06.2020).
5. Jansen T., Zarges C. Performance analysis of randomised search heuristics operating with a fixed budget. *Theoretical Computer Science*, 2014, vol. 545, pp. 39–58. doi: 10.1016/j.tcs.2013.06.007
6. Doerr B., Jansen T., Witt C., Zarges C. A method to derive fixed budget results from expected optimisation times. *Proc. 15th Genetic and Evolutionary Computation Conference (GECCO 2013)*, 2013, pp. 1581–1588. doi: 10.1145/2463372.2463565
7. Lengler J., Spooner N. Fixed budget performance of the (1+1) EA on linear functions. *Proc. 13th Foundations of Genetic Algorithms (FOGA)*, 2015, pp. 52–61. doi: 10.1145/2725494.2725506
8. Witt C. Runtime analysis of the (μ+1) EA on simple pseudo-Boolean functions. *Evolutionary Computation*, 2006, vol. 14, no. 1, pp. 65–86. doi: 10.1162/106365606776022751
9. Pinto E.C., Doerr C. *Towards a more practice-aware runtime analysis of evolutionary algorithms*. Available at: https://arxiv.org/abs/1812.00493 (accessed: 27.06.2020).
10. Vinokurov D., Buzdalov M., Buzdalova A., Doerr B., Doerr C. Fixed-target runtime analysis of the (1+1) EA with resampling. *Proc. Genetic and Evolutionary Computation Conference Companion (GECCO'19)*, 2019, pp. 2068–2071. doi: 10.1145/3319619.3326906
11. Doerr B. Analyzing randomized search heuristics via stochastic domination. *Theoretical Computer Science*, 2019, vol. 773, pp. 115–137. doi: 10.1016/j.tcs.2018.09.024
12. Böttcher S., Doerr B., Neumann F. Optimal fixed and adaptive mutation rates for the LeadingOnes problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6238, pp. 1–10. doi: 10.1007/978-3-642-15844-5_1
13. Badkobeh G., Lehre P.K., Sudholt D. Unbiased black-box complexity of parallel search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8672, pp. 892–901. doi: 10.1007/978-3-319-10762-2_88
14. Doerr B., Doerr C, Ebel F. From black-box complexity to designing new genetic algorithms. *Theoretical Computer Science*, 2015, vol. 567, pp. 87–104. doi: 10.1016/j.tcs.2014.11.028
15. Wegener I. Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. *International Series in Operations Research & Management Science*, 2003, vol. 48, pp. 349–369. doi: 10.1007/0-306-48041-7_14
16. Sudholt D. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2013, vol. 17, no. 3, pp. 418–435. doi: 10.1109/TEVC.2012.2202241
17. Doerr B., Doerr C. The impact of random initialization on the runtime of randomized search heuristics. *Algorithmica*, 2016, vol. 75, no. 3, pp. 529–553. doi: 10.1007/s00453-015-0019-5

18. Spivey M.Z. Combinatorial sums and finite differences // Discrete Mathematics. 2007. V. 307. N 24. P. 3130–3146. doi: 10.1016/j.disc.2007.03.052

18. Spivey M.Z. Combinatorial sums and finite differences. *Discrete Mathematics*, 2007, vol. 307, no. 24, pp. 3130–3146. doi: 10.1016/j.disc.2007.03.052

Авторы

Буздалов Максим Викторович — кандидат технических наук, научный сотрудник, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 48160963700, ORCID ID: 0000-0002-7120-8824, mbuzdalov@gmail.com

Винокуров Дмитрий Владимирович — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 57215125737, ORCID ID: 0000-0001-9074-3884, dimaoq@gmail.com

Authors

Maxim V. Buzdalov — PhD, Scientific Researcher, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 48160963700, ORCID ID: 0000-0002-7120-8824, mbuzdalov@gmail.com

Dmitry V. Vinokurov — Postgraduate, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 57215125737, ORCID ID: 0000-0001-9074-3884, dimaoq@gmail.com