

Теоретический анализ метода выбора вспомогательных критериев на задачах XdivK и Generalized OneMax ¹

Петрова И. А., аспирант кафедры компьютерных технологий Университета
ИТМО, irenepetrova@yandex.com

Буздalова А. С., аспирант кафедры компьютерных технологий
Университета ИТМО, abuzdalova@gmail.com

Аннотация

Эффективность однокритериального эволюционного алгоритма может быть повышена путем введения вспомогательных критериев. Существует метод EA+RL, основанный на обучении с подкреплением, позволяющий выбирать оптимизируемый критерий на каждом шаге эволюционного алгоритма. Однако на практике вспомогательные критерии могут быть не только полезными, но и мешающими на некоторых этапах оптимизации. Ранее было доказано, что в случае использования таких критериев метод EA+RL не справляется с решением задачи Generalized OneMax, а время его работы на задаче XdivK намного больше времени работы метода спуска со случайными мутациями. В данной работе предлагается модификация метода EA+RL и проводится ее теоретический анализ на задачах XdivK и Generalized OneMax. Доказывается, что предложенная модификация справляется с решением данных задач, в отличие от метода EA+RL. Приводятся асимптотические оценки времени работы предложенной модификации.

Введение

Рассмотрим задачу однокритериальной оптимизации целевого критерия при помощи эволюционного алгоритма (ЭА) [11]. Одной из наиболее распространенных оценок эффективности ЭА, используемой в данной работе, является число вычислений функции приспособленности (ФП), необходимое для достижения оптимума.

Эффективность оптимизации целевого критерия может быть повышена путем введения вспомогательных критериев [1, 7, 9]. Одним из методов, позволяющим выбирать оптимизируемый критерий на каждом шаге оптими-

¹Выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-31-00380 мол_а

зации является метод EA+RL [4], основанный на обучении с подкреплением [10]. В методе EA+RL агент обучения выбирает оптимизируемый критерий — целевой или один из вспомогательных и применяет его к среде в качестве которой выступает ЭА. ЭА создает новое поколение с использованием данного критерия и возвращает агенту численную награду, равную разнице лучших значений целевого критерия в новом и предыдущем поколениях. Целью агента является максимизация суммарной награды.

На практике вспомогательные критерии зачастую создаются автоматически и могут быть эффективными на одних этапах оптимизации и мешающими на других [2, 6]. Будем называть такие критерии *переключающимися*.

Эффективность метода EA+RL была теоретически доказана в случае использования стационарных вспомогательных критериев [3], то есть таких критериев, которые не меняют своей эффективности в процессе оптимизации. Однако в случае использования переключающихся вспомогательных критериев при решении задачи XdivK, EA+RL не справляется с игнорированием критерия, являющегося мешающим на данном шаге оптимизации [8]. Также при решении задачи Generalized OneMax с переключающимися вспомогательными критериями, EA+RL застревает в локальном оптимуме [5].

В существующих теоретических исследованиях метода EA+RL в качестве ЭА использовался метод спуска со случайными мутациями (Random Local Search, RLS), а в качестве алгоритма обучения с подкреплением использовался Q-learning. Данная конфигурация метода EA+RL обозначается как RLS+Q-learning. Поколение состояло из одной особи [8]. Особи представлялись битовыми строками, использовалась однобитовая мутация. В случае если значения оптимизируемого критерия новой и старой особи совпадали, в качестве нового поколения выбиралась новая особь. Состояние ЭА определялось как значение целевого критерия в текущем поколении. Псевдокод метода RLS+Q-learning представлен в листинге 1. В алгоритме Q-learning эффективность выбора критерия h в состоянии s определяется значением $Q(s, a)$ которое обновляется на каждом шаге алгоритма в строке 12 псевдокода, где α — скорость обучения, а γ — дисконтный фактор.

Модификация метода EA+RL

В методе EA+RL в случае когда новая особь лучше текущей по выбранному критерию, в следующее поколение выбирается новая особь. Однако в случае когда выбранный критерий является мешающим, новая особь может быть хуже текущей по целевому критерию. В таком случае ЭА теряет особь с лучшим значением целевого критерия.

В предложенной модификации, в случае когда новая особь лучше текущей по выбранному критерию, однако хуже по целевому, в следующее

поколение выбирается текущая особь. Таким образом, в отличие от метода EA+RL, в предложенной модификации лучшее найденное решение никогда не теряется. В теоретических исследованиях предложенной модификации использовалась описанная выше конфигурация метода EA+RL, использовавшаяся в существующих теоретических исследованиях.

Листинг 1 Алгоритм RLS + Q-learning

```

1: Особь  $y \leftarrow$  случайная битовая строка
2: Множество  $H$  состоит из вспомогательных и целевого критерия
3:  $Q(s, h) \leftarrow 0$  для каждого состояния  $s$  и критерия  $h \in H$ 
4: while (Оптимум целевого критерия  $t$  не найден) do
5:   Текущее состояние  $s \leftarrow t(y)$ 
6:   Особь  $y' \leftarrow$  результат мутации особи  $y$  (инверсия случайного бита)
7:   Критерий  $h: Q(s, h) = \max_{h' \in H} Q(s, h')$  {Если значения  $Q$  одинаковы, критерии выбираются равновероятно}
8:   if  $h(y') \geq h(y)$  then
9:      $y \leftarrow y'$ 
10:  Новое состояние  $s' \leftarrow t(y)$ 
11:  Награда  $r \leftarrow s' - s$ 
12:   $Q(s, h) \leftarrow Q(s, h) + \alpha(r + \max_{h' \in H} Q(s', h') - Q(s, h))$ 

```

Модельные задачи

Теоретический анализ модификации метода EA+RL проводился на задачах XdivK и Generalized OneMax, ранее использовавшихся при анализе метода EA+RL [5, 8]. В обеих рассмотренных задачах особью является битовая строка длины n .

Значение целевого критерия задачи Generalized OneMax, далее обозначаемый как OM_d , вычисляется как число бит в особи, совпадающих с заданной битовой маской. Битовая маска содержит d нулей и $n - d$ единиц.

Значение целевого критерия задачи XdivK вычисляется как $\lfloor \frac{x}{k} \rfloor$, где x число единиц, а k — некоторая константа, такая что n делится нацело на k .

В обеих задачах были использованы два переключающихся вспомогательных критерия. Для описания вспомогательных критериев вводятся две функции OneMax и ZeroMax. Значениями функций OneMax и ZeroMax является число единичных и нулевых бит в особи соответственно. Вспомогательные критерии h_1 и h_2 , задающиеся формулой 1, могут быть равны как OneMax (OM) так и ZeroMax (ZM) в зависимости от этапа оптимизации. Па-

параметр p называется *точкой переключения*.

$$h_1(x) = \begin{cases} \text{OM}, x \leq p \\ \text{ZM}, p < x \leq n \end{cases} \quad h_2(x) = \begin{cases} \text{ZM}, x \leq p \\ \text{OM}, p < x \leq n \end{cases} \quad (1)$$

В задаче XdivK критерий, на текущем шаге оптимизации равный OneMax, позволяет различить особи с одинаковым значением целевого критерия и выбрать из них ту, в которой содержится большее число единиц. Такая особь с большей вероятностью породит особь с более высоким значением целевого критерия. Так как целью задачи XdivK является получение особи состоящей только из единиц, критерий, на текущем шаге оптимизации равный ZeroMax, является мешающим.

В задаче OM_d , вспомогательные критерии могут быть мешающими или совпадающими с целевым. Рассмотрим пример, показывающий почему это так. Пусть i -й бит в битовой маске равен единице и оператор мутации инвертирует i -й бит особи из 0 в 1. В таком случае критерий равный OneMax совпадает с целевым, а критерий равный ZeroMax является мешающим, так как новая особь с лучшим значением целевого критерия не будет принята.

Анализ модификации EA+RL на Generalized OneMax

Для оценки времени работы предложенной модификации строится марковская цепь, представляющая соответствующий процесс оптимизации. [3, 8]. Состояние обучения с подкреплением и марковское состояние определяются как значение целевого критерия, а именно как число совпавших с маской бит в особи. Отметим, что в данной задаче как только алгоритм находит особь с более высоким значением целевого критерия, агент переходит в новое состояние обучения, в котором у него нет опыта. В случае, когда найденная особь имеет худшее или такое же значение целевого критерия агент остается в текущем состоянии и получает нулевую награду. Таким образом, оптимизируемый критерий всегда выбирается равновероятно.

Марковская цепь для предложенной модификации представлена на рис. 1. На переходах отмечены их вероятности, которые вычислялись следующим образом. Пусть в состоянии i из i совпавших с маской бит k единичных. Обозначим число единичных бит из несовпавших $n - i$ бит как t . В случае, когда выбранный критерий равен OneMax, вероятность перехода в состояние $i + 1$ равна вероятности инверсии нулевого несовпадающего бита, а именно $\frac{n-i-t}{n}$. Вероятность остаться в состоянии i равна сумме вероятностей инверсии единичного несовпадающего бита и инверсии совпадающего бита, а именно $\frac{t}{n} + \frac{i}{n}$. С помощью аналогичных рассуждений получаем, что выборе критерия равного ZeroMax вероятность перехода в состояние $i + 1$

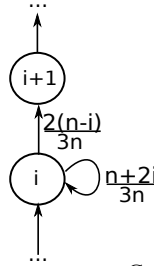


Рис. 1: Марковская цепь для Generalized OneMax

равна $\frac{t}{n}$, а вероятность остаться в состоянии i равна $\frac{n-i}{n}$. При выборе целевого критерия вероятности перейти в состояние $i+1$ и остаться в состоянии i равны $\frac{n-i}{n}$ и $\frac{i}{n}$ соответственно. Таким образом получаем, что вероятность перейти из состояния i в состояние $i+1$ равна $\frac{1}{3} \cdot (\frac{n-i-t}{n} + \frac{t}{n} + \frac{n-i}{n}) = \frac{2}{3} \cdot \frac{n-i}{n}$. Вероятность остаться в состоянии i соответственно равна $\frac{1}{3} \cdot \frac{n+2i}{n}$.

Математическое ожидание $T(n)$ времени работы предложенной модификации равно числу вычислений ФП, необходимому для достижения марковского состояния n из нулевого состояния. Каждый переход в марковской цепи соответствует вычислению ФП особи, полученной в результате мутации. Поэтому математическое ожидание времени работы алгоритма равно числу сделанных алгоритмом переходов в марковской цепи. $T(n)$ вычисляется как $\sum_{i=0}^{n-1} E(i \rightarrow i+1)$, где $E(i \rightarrow i+1)$ математическое ожидание числа переходов, необходимое для достижения марковского состояния $i+1$ из i .

Из вида марковской цепи получаем, что $E(i \rightarrow i+1) = 1 \cdot \frac{2}{3} \cdot \frac{n-i}{n} + (1 + E(i \rightarrow i+1)) \cdot \frac{1}{3} \cdot \frac{n+2i}{n}$. Отсюда получаем, что $E(i \rightarrow i+1) = \frac{3}{2} \cdot (1 + \frac{i}{n-i})$. Подставляя полученное выражение для $E(i \rightarrow i+1)$ в формулу для $T(n)$, получаем $T(n) = \frac{3}{2} \cdot \sum_{i=0}^{n-1} (1 + \frac{i}{n-i}) = O(n \log n)$. Таким образом, время работы предложенной модификации и RLS на задаче Generalized OneMax асимптотически одинаково.

Анализ модификации EA+RL на XdivK

Для оценки времени работы предложенной модификации на задаче XdivK строится марковская цепь. Марковское состояние соответствует числу единичных бит в особи. Состояние обучения с подкреплением определяется как значение целевого критерия. Таким образом, каждое состояние обучения с подкреплением включает k марковских состояний.

Марковская цепь для предложенной модификации представлена на рис. 2а. Для анализа времени работы предложенной модификации также необходимо построить марковскую цепь для алгоритма RLS без вспомогательных критериев (рис. 2б). Данная марковская цепь строится аналогично

марковской цепи для предложенной модификации. Метки на переходах имеют вид $\langle F, M \rangle$, где F — критерий, который может быть выбран для осуществления данного перехода, M — соответствующий результат мутации.

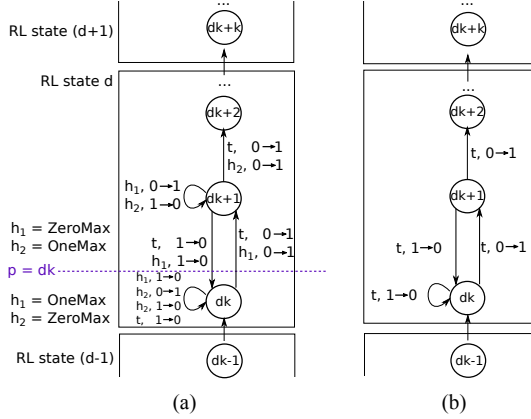


Рис. 2: Марковские цепи для XdivK: модификация EA+RL (a), RLS (b)

Математическое ожидание $T(n)$ вычисляется как и при анализе задачи Generalized OneMax по формуле $T(n) = \sum_{i=0}^{n-1} E(i \rightarrow i+1)$, где $E(i \rightarrow i+1)$ математическое ожидание числа переходов, необходимое для достижения марковского состояния $i+1$ из состояния i .

Значение $E(i \rightarrow i+1)$ зависит от номера состояния i . Рассмотрим два случая. В первом случае $i = dk$, где d некоторая константа. Математическое ожидание числа переходов необходимое для достижения состояния $dk+1$ из состояния dk вычисляется как $z_{dk} = E(dk \rightarrow dk+1) = \frac{2}{3} \cdot \frac{(n-dk)}{n} \cdot 1 + (\frac{2}{3} \cdot \frac{dk}{n} + \frac{1}{3}) \cdot (1 + z_{dk})$. Отсюда получаем, что $z_{dk} = \frac{3n}{2(n-dk)}$.

Во втором случае $i = dk+t$, где $1 \leq t \leq k-1$. Математическое ожидание числа переходов необходимое для достижения состояния $dk+t+1$ из состояния $dk+t$ вычисляется как $z_{dk+t} = E(dk+t \rightarrow dk+t+1) = \frac{2(n-dk-t)}{3n} + \frac{2(dk+t)}{3n} \cdot (1 + z_{dk+t-1} + z_{dk+t}) + (\frac{dk+t}{3n} + \frac{n-dk-t}{3n}) \cdot (1 + z_{dk+t})$. Отсюда получаем, что $z_{dk+t} = z_{dk+t-1} \cdot \frac{dk+t}{n-dk-t} + \frac{3n}{2(n-dk-t)}$.

Для получения оценки на время работы предложенной модификации вычислим математическое ожидание времени работы RLS на задаче XdivK. Метод вычисления аналогичен методу, использованному выше для предложенной модификации. Математическое ожидание общего времени работы вычисляется как $T(n)$. Вновь рассматриваются два случая на номер состо-

яния i : $i = dk$ and $i = dk + t$. Математическое ожидание числа переходов необходимое для достижения состояния $dk + 1$ из dk вычисляется как $a_{dk} = E(dk \rightarrow dk + 1) = \frac{(n-dk)}{n} \cdot 1 + \frac{dk}{n} \cdot (1 + a_{dk})$. Отсюда получаем, что $a_{dk} = \frac{n}{(n-dk)}$. Математическое ожидание числа переходов необходимое для достижения состояния $dk + t + 1$ из $dk + t$ вычисляется как $a_{dk+t} = E(dk+t \rightarrow dk+t+1) = \frac{(n-dk-t)}{n} \cdot 1 + \frac{dk+t}{n} \cdot (1 + a_{dk+t-1} + a_{dk+t})$. Отсюда получаем, что $a_{dk+t} = a_{dk+t-1} \cdot \frac{dk+t}{n-dk-t} + \frac{n}{(n-dk-t)}$.

Из выражений для z_{dk} и a_{dk} получаем, что $z_{dk} = \frac{3}{2}a_{dk}$. Из выражений для $T(n)$, z_{dk+t} , a_{dk+t} с помощью математической индукции получаем, что математическое ожидание времени работы предложенной модификации с переключающимися вспомогательными критериями на задаче XdivK в 1.5 раза больше времени работы RLS. Таким образом, нижняя и верхняя асимптотические оценки времени работы предложенной модификации и RLS совпадают и равны $\Omega(n^k)$ и $O(n^{k+1})$ соответственно [3]. Таким образом, предложенная модификация способна работать в случае использования переключающихся вспомогательных критериев в отличие от метода EA+RL.

Заключение

В работе предложена модификация метода EA+RL, сохраняющее лучшее найденное решение. Проведен теоретический анализ времени работы предложенной модификации на задачах Generalized OneMax и XdivK с переключающимися вспомогательными критериями. Доказано, что асимптотическая оценка времени работы предложенной модификации на задаче Generalized OneMax составляет $\Theta(n \log n)$, а на задаче XdivK нижняя и верхняя оценки времени работы равны $\Omega(n^k)$ и $O(n^{k+1})$ соответственно. Данные оценки совпадают с оценками времени работы RLS. Таким образом, предложенная модификация позволяет распространить применимость метода EA+RL на случай использования переключающихся вспомогательных критериев.

Литература

- [1] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. On the effects of adding objectives to plateau functions // IEEE Transactions on Evolutionary Computation, 13(3):591–603, 2009.
- [2] M. Buzdalov and A. Buzdalova. Adaptive selection of helper-objectives for test case generation // 2013 IEEE Congress on Evolutionary Computation, volume 1, pages 2245–2250, 2013.
- [3] M. Buzdalov and A. Buzdalova. OneMax helps optimizing XdivK: Theoretical runtime analysis for RLS and EA+RL // Proceedings of Genetic

and Evolutionary Computation Conference Companion, pages 201–202. ACM, 2014.

- [4] A. Buzdalova and M. Buzdalov. Increasing efficiency of evolutionary algorithms by choosing between auxiliary fitness functions with reinforcement learning // Proceedings of the International Conference on Machine Learning and Applications, volume 1, pages 150–155, 2012.
- [5] A. Buzdalova, I. Petrova, and M. Buzdalov. Runtime analysis of different approaches to select conflicting auxiliary objectives in the generalized onemax problem // Proceedings of IEEE Symposium Series on Computational Intelligence, pages 280–286, 2016.
- [6] D. F. Lochtefeld and F. W. Ciarallo. Helper-objective optimization strategies for the Job-Shop scheduling problem // Applied Soft Computing, 11(6):4161–4174, 2011.
- [7] F. Neumann and I. Wegener. Can single-objective optimization profit from multiobjective optimization? // Multiobjective Problem Solving from Nature, Natural Computing Series, pages 115–130. Springer Berlin Heidelberg, 2008.
- [8] I. Petrova, A. Buzdalova, and G. Korneev. Runtime analysis of random local search with reinforcement based selection of non-stationary auxiliary objectives: initial study // Proceedings of 22nd International Conference on Soft Computing MENDEL 2016, pages 95–102, Czech Republic, 2016.
- [9] C. Segura, C. A. C. Coello, G. Miranda, and C. Léon. Using multi-objective evolutionary algorithms for single-objective optimization // 4OR, 3(11):201–228, 2013.
- [10] R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction // MIT Press, Cambridge, MA, USA, 1998.
- [11] Ю.А. Скобцов. Основы эволюционных вычислений // ДонНТУ, Донецк, 2008.