СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДА ВЫБОРА ВСПОМОГАТЕЛЬНЫХ КРИТЕРИЕВ И МЕТОДА СПУСКА СО СЛУЧАЙНЫМИ МУТАЦИЯМИ

М. В. Буздалов

ассистент кафедры компьютерных технологий Университета ИТМО

E-mail: mbuzdalov@gmail.com

А. С. Буздалова

студентка кафедры компьютерных технологий Университета ИТМО

E-mail: abuzdalova@gmail.com

Аннотация. Рассматривается ранее предложенный метод выбора вспомогательных критериев оптимизации EA+RL, предназначенный для повышения эффективности эволюционных алгоритмов и основанный на применении обучения с подкреплением. Формулируется модельная задача, на примере которой сравнивается время работы этого метода и метода спуска со случайными мутациями. Приводится схема доказательства, показывающего, что отношение времен работы рассматриваемых методов выражается как экспонента от параметра модельной задачи, что подтверждает эффективность метода EA+RL.

Введение

Эффективность однокритериальной оптимизации в некоторых случаях может быть повышена путем введения вспомогательных критериев [1]. Насколько известно авторам, в последнее время ведутся активные исследования по применению вспомогательных критериев в задачах дискретной оптимизации, решаемых с помощью эволюционных алгоритмов [2–6].

Использование вспомогательных критериев позволяет избежать остановку процесса оптимизации в локальных оптимумах целевого критерия [4, 5], а также расширяет исследуемую область пространства поиска [2], за счет чего оптимальное значение целевого критерия может быть найдено за меньшее число итераций эволюционного алгоритма.

Вспомогательные критерии, как правило, формулируются в ходе анализа задачи. Например, может проводиться декомпозиция целевого критерия на вспомогательные [6, 7]. Также существует пример автоматической генерации вспомогательных критериев для задачи о генерации тестов против решений олимпиадных задач [8].

Обычно об эффективности вспомогательных критериев довольно сложно судить заранее. Более того, один и тот же вспомогательный критерий на различных этапах процесса оптимизации может как ускорять поиск оптимального значения целевого критерия, так и замедлять его [7]. В связи с этим воз-

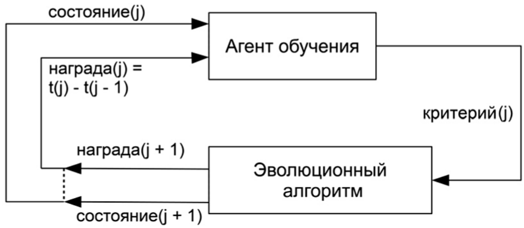

Рис. 1. Метод выбора критериев EA+RL,
t — целевой критерий, *j* — номер итерации

никает задача автоматического выбора вспомогательного критерия, наиболее эффективного на данном этапе оптимизации, из заранее подготовленного набора критериев.

Автоматический выбор вспомогательных критериев может производиться с помощью ранее предложенного метода EA+RL [9], основанного на выборе критериев оптимизации для эволюционного алгоритма с помощью обучения с подкреплением (см. рис. 1) [10, 11]. Агент обучения на каждой итерации эволюционного алгоритма выбирает критерий оптимизации из списка, состоящего из вспомогательных критериев и целевого. Выбранный критерий используется при формировании очередного поколения эволюционного алгоритма. Затем формируется некоторое представление состояния эволюционного алгоритма, а также награда, зависящая от роста целевого критерия. Награда используется для обновления оценки ожидаемой награды в данном состоянии. Выбирается критерий, максимизирующий оценку ожидаемой награды. В случае, когда оценка одинакова, критерии выбираются равновероятно.

Эффективность метода EA+RL была подтверждена экспериментально на примере решения ряда модельных задач, а также практической задачи генерации тестов [8]. Также существует теоретический результат, показывающий на примере одной модельной задачи, что метод EA+RL позволяет игнорировать неэффективный вспомогательный критерий [12]. Цель данной работы — показать теоретически, что применение метода EA+RL может повысить эффективность оптимизации целевого критерия.

Постановка модельной задачи

Опишем модельную задачу *XdivK* с одним вспомогательным критерием. Пространство поиска — битовые строки длины n. Пусть x — число единиц в битовой строке. Требуется максимизировать целевой критерий:

$$t = \left\lfloor \frac{x(s)}{k} \right\rfloor,$$

где k — целочисленный параметр, $n \bmod k = 0$. В качестве вспомогательного критерия будем использовать значение x.

Заметим, что оптимизация по вспомогательному критерию позволяет достичь оптимума целевого критерия за меньшее число итераций. Пусть есть две битовые строки a и b с числом единиц y и z соответственно, $y < z$ (см. рис. 2). Пусть также с точки зрения целевого критерия эти строки не различаются. Однако использование строки b с большим числом единиц позволяет улучшить значение целевого критерия с большей вероятностью, чем использование строки a. Вспомогательный критерий ускоряет процесс оптимизации целевого критерия за счет того, что он позволяет выбрать строку b.

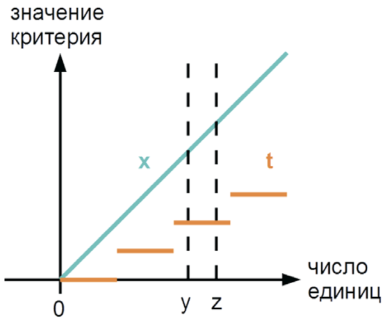

Рис. 2. Целевой и вспомогательный критерии в задаче *XdivK*

Далее на примере описанной модельной задачи будет сравнено время работы метода EA+RL и эволюционного алгоритма, не использующего вспомогательный критерий. Требуется показать, что метод EA+RL позволит использовать преимущества вспомогательного критерия и найти оптимальное значение целевого критерия за меньшее число итераций.

В качестве эволюционного алгоритма будем рассматривать метод спуска со случайными мутациями. В качестве обучения с подкреплением используется алгоритм Q-обучения [10, 11], поддерживающий текущую оценку ожидаемой награды, которая обновляется в соответствии с формулой: $Q(s, f) = Q(s, f) + \alpha(r + \gamma \max_{f'} Q(s', f') — Q(f, a))$, где f — выбранный критерий, состояние s — значение целевого критерия, награда r — разность значений целевого критерия на двух последовательных итерациях. Изначально Q инициализируется нулями.

Анализ времени работы метода EA+RL и метода спуска со случайными мутациями

В данном разделе представлена общая схема доказательства оценки времени работы метода EA+RL при решении рассматриваемой задачи. Полную версию доказательства можно прочитать в статье [13].

Представим процесс оптимизации как цепь Маркова (см. рис. 3). Общий вид цепи одинаков как для метода EA+RL, так и для метода спуска со случайными мутациями. Вершины цепи соответствуют числу единиц в строке. Цветом выделены вершины, в которых x нацело делится на k. Заметим, что из этих вершин отсутствуют переходы в вершины с меньшим значением x, так как эволюционный алгоритм выбирает строки с большим или равным значением текущего критерия.

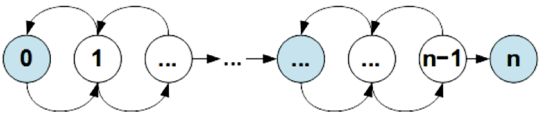

Рис. 3. Процесс оптимизации в виде цепи Маркова

Рассмотрим число итераций, необходимое для перехода из вершины x в вершину $x+1$. Пусть $Z_E(x)$ — математическое ожидание числа итераций метода спуска, $Z_R(x)$ — математическое ожидание числа итераций метода EA+RL. На рис. 4 для метода спуска и метода EA+RL рядом с каждым переходом цепи Маркова подписаны выбранный критерий, мутация и вероятность соответствующего перехода.

Оценим вероятности переходов в случае использования метода спуска (см. рис. 4, слева). Для состояний, в которых x делится нацело на k ($x \bmod k = 0$), выполняется соотношение:

$$Z_E(x) = \frac{n-x}{n} + \frac{x}{n}(1 + Z_E(x)) = \frac{n}{n-x},$$

в то время как для остальных состояний выполняется:

$$Z_E(x) = \frac{n-x}{n} + \frac{x}{n}(1 + Z_E(x-1) + Z_E(x)) = \frac{n}{n-x} + Z_E(x-1)\frac{x}{n-x}.$$

Рассмотрим теперь вероятности переходов в случае использования метода EA+RL, выбирающего на каждой итерации вспомогательный или целевой

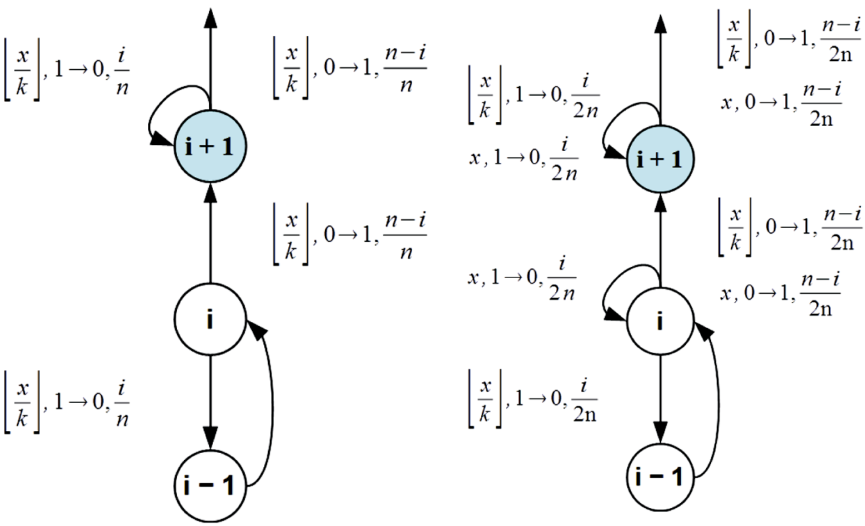

Рис. 4. Вероятности переходов в методе спуска (слева) и в методе EA+RL (справа)

критерий и передающий этот критерий в метод спуска со случайными мутациями (см. рис. 4, справа). В этом случае вероятность перехода в состояние с меньшим числом единиц ниже, так как при выборе вспомогательного критерия переход в строку с меньшим числом единиц осуществляться не будет.

Для состояний, в которых x делится нацело на k ($x \bmod k = 0$), выполняется соотношение:

$$Z_R(x) = \frac{n-x}{n} + \frac{x}{n}(1 + Z_R(x)) = \frac{n}{n-x},$$

для остальных состояний выполняется:

$$Z_R(x) = \frac{n-x}{n} + \frac{x}{2n}(1 + Z_R(x)) + \frac{x}{2n}(1 + Z_R(x-1) + Z_R(x)),$$

$$Z_R(x) = \frac{n}{n-x} + Z_R(x-1)\frac{x}{2(n-x)}.$$

Приведенные выражения представляются как:

$$Z_E(x) = \sum_{i=0}^{x \bmod k} \frac{\binom{n}{x-i}}{\binom{n-1}{x}}, \qquad Z_R(x) = \sum_{i=0}^{x \bmod k} 2^{-i}\frac{\binom{n}{x-i}}{\binom{n-1}{x}},$$

что можно доказать по индукции.

Для вычисления общего числа итераций, необходимого для достижения максимального значения целевого критерия, следует просуммировать полученные выражения вдоль рассмотренной цепи Маркова:

$$T_E(x) = \sum_{x=0}^{n-1} Z_E(x) \text{ и } T_R(x) = \sum_{x=0}^{n-1} Z_R(x).$$

Можно показать, что $T_E(n, k) = T_R(n, k) = \Omega(n^k) = O(n^{k+1})$, где k — константа. Также можно показать, что если $n \to \infty$ при фиксированном k, то

$$\frac{T_E(n, k)}{T_R(n, k)} \geq 2^{k-2}(1 - O(1)).$$

Последнее выражение означает, что метод EA+RL позволяет решить модельную задачу *XdivK* не менее, чем в 2^{k-2} раза быстрее, чем метод спуска со случайными мутациями. Был проведен численный эксперимент, который показал, что данную оценку можно попытаться улучшить до 2^{k-1}.

Заключение

Проведен сравнительный анализ времени работы метода выбора вспомогательных критериев оптимизации EA+RL и метода спуска со случайными мутациями. Показано, что метод EA+RL позволяет решить рассмотренную модельную задачу экспоненциально быстрее в зависимости от параметра задачи. В сформулированной задаче присутствует один вспомогательный критерий, эффективный на протяжении всего процесса оптимизации. В рамках дальнейших исследований целесообразно провести анализ времени работы метода EA+RL на примере задачи, где эффективность вспомогательных критериев зависит от этапа оптимизации.

Литература

1. *Евтушенко Ю. Г., Жадан В. Г.* Точные вспомогательные функции в задачах оптимизации // Журнал вычислительной математики и математической физики. № 1. Т. 30. 1990. С. 43–57.
2. *Neumann F., Wegener I.* Can single-objective optimization profit from multiobjective optimization? // Knowles J., Corne D., Deb K., Chair D. (Eds.), Multiobjective Problem Solving from Nature, Natural Computing Series, Springer Berlin Heidelberg. 2008. P. 115–130.
3. *Segura C., Coello C. A. C., Miranda G., Léon C.* Using multi-objective evolutionary algorithms for single-objective optimization / 4OR 11 (3) . 2013. P. 201–228.
4. *Knowles J. D., Watson R. A., Corne D.* Reducing local optima in single-objective problems by multi-objectivization // Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, EMO'01, Springer-Verlag, London, UK. 2001. P. 269–283.
5. *D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, E. Zitzler*. On the effects of adding objectives to plateau functions // IEEE Trans. Evolutionary Computation 13 (3). 2009. P. 591–603.
6. *Handl J., Lovell S., Knowles J.* Multiobjectivization by decomposition of scalar cost functions // Rudolph G., Jansen T., Lucas S., Poloni C., Beume N. (Eds.), Parallel Problem Solving from Nature PPSN X, Vol. 5199 of Lecture Notes in Computer Science, Springer Berlin Heidelberg. 2008. P. 31–40.
7. *Jensen M. T.* Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation: Evolutionary computation combinatorial optimization // Journal of Mathematical Modelling and Algorithms 3 (4). 2004. P. 323–347.
8. *Буздалов М. В*. Генерация тестов для олимпиадных задач по программированию с использованием генетических алгоритмов // Научно-технический вестник СПбГУ ИТМО. 2011. № 2 (72). С. 72–77.
9. *Buzdalova A., Buzdalov M.* Increasing Efficiency of Evolutionary Algorithms by Choosing between Auxiliary Fitness Functions with Reinforcement Learning // Proceedings of the Eleventh International Conference on Machine Learning and Applications, ICMLA 2012. Boca Raton: IEEE Computer Society. Vol. 1. 2012. P. 150–155.
10. *Sutton R. S., Barto A. G.* Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 1998. 322 p.

11. *Николенко С. И., Тулупьев А. Л.* Самообучающиеся системы. М.: МЦНМО, 2009. 288 с.
12. *Buzdalov M., Buzdalova A., Shalyto A.* First Step towards the Runtime Analysis of Evolutionary Algorithm Adjusted with Reinforcement Learning // Proceedings of the Twelve International Conference on Machine Learning and Applications, ICMLA 2013. Boca Raton: IEEE Computer Society, 2013. Vol. 1. P. 203–208.
13. *Buzdalov M., Buzdalova A.* OneMax Helps Optimizing XdivK: Theoretical Runtime Analysis for RMHC and EA+RL // Proceedings of the Genetic and Evolutionary Computation Conference «GECCO-2014». Vol. 2. 2014. P. 201–202.