УДК 51

СРАВНЕНИЕ МЕТОДОВ САМОНАСТРОЙКИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Становов В. В.
научный руководитель д-р техн. наук Семенкин Е. С.
Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева

Генетические алгоритмы являются мощным оптимизационным средством, которое в наши дни часто используется при решении сложных оптимизационных задач с множеством переменных, таких как задачи настройки весов нейронных сетей, настройки параметров нечетких контроллеров и другие. Несмотря на широкую распространенность генетических алгоритмов, в большинстве случаев исследователи используют какую-либо стандартную схему с одним типом селекции, одним типом скрещивания и фиксированной вероятностью мутации. Такой подход может быть достаточно эффективен, однако эта эффективность напрямую зависит от решаемой задачи.

Как было показано в нескольких предыдущих работах, некоторые из генетических операторов показывают себя лучше на функциях определенного вида. То есть, к примеру, для некоторых задач оптимальными могут быть пропорциональная селекция, равномерное скрещивание и сильная мутация, а для других – другие. В сущности, изменение операторов меняет поведение алгоритма, то есть, например, сильная мутация дает гораздо больший разброс точек, чем слабая, а турнирная селекция позволяет добиться значительного селективного давления. При этом установка одной неизменной комбинации параметров, которая хорошо показывает себя на конкретной задаче, может быть не лучшим выбором, так как для наиболее эффективного и быстрого решения задачи может потребоваться изменение поисковых характеристик в процессе работы алгоритма.

Например, если в начале работы алгоритма требуется значительный разброс точек (большая вероятность мутации) для выделения наиболее перспективных областей поиска, то на поздних этапах необходима конфигурация, которая бы позволила быстро сойтись к локальным оптимумам. Таким образом, необходимо изменять конфигурацию в процессе работы алгоритма, то есть метод, который бы мог изменять применяемые операторы по ходу поиска.

Существует несколько подходов к решению данной задачи, среди которых можно выделить коэволюцию, методы самонастройки и самоадаптации. Коэволюционный подход подразумевает конкуренцию нескольких конфигураций алгоритмов, работающих параллельно за вычислительные ресурсы. При этом конфигурации выбираются так, чтобы среди них были и направленные на глобальный, и на локальный поиск. Методы самонастройки подразумевают введение некоторых вероятностей применения каждого оператора на основании его успешности. То есть, в одном алгоритме могут быть применены все имеющиеся типы операторов, но каким-то из них будет отдано предпочтение, если они лучше покажут себя в процессе поиска.

В данной работе будут рассмотрены два метода самонастройки, взятые из научной литературы,метод PDP (Population-levelDynamicProbabilities), основанный на подсчете числа улучшений данным оператором,и метод, основанный на значениях пригодностей потомков. Данные алгоритмы универсальны и могут быть применены не только к генетическому алгоритму, но также и к любому другому эволюционному алгоритму, в котором требуется настройка параметров. Рассмотрим каждый из этих методов подробнее.

Первый метод назначает каждому из генетических операторов минимальное значение вероятности его применения $p_{all} = \frac{0.2}{n}$, где n–число различных операторов данного типа. Данные пороговые значения вероятностей необходимы для того, чтобы ни одна из вероятностей применения операторов не стала равной нулю в процессе работы

алгоритма, так как в этом случае, если на каком-то этапе потребуется именно этот оператор, он не будет иметь возможности повысить свою вероятность применения.

Далее, вводятся значения $r_i = \frac{success_i{}^2}{used_i}$для каждого оператора i, где $used_i$–число применений данного оператора, $success_i$–число успешных применений оператора, то есть случаев, когда после использования данного оператора пригодность потомка превысила пригодность соответствующего родителя. Для каждого оператора рассчитывается вероятность его применения p_iпо следующей формуле:

$$p_i = p_{all} + \left[r_i \frac{1.0 - np_{all}}{scale}\right],$$

$$scale = \sum_{j=1}^{n} r_j$$

Значения $success$возводятся в квадрат из-за очень низкой степени успешности генетических операторов. Значения $scale$ позволяют нормировать значения вероятностей так, что их сумма всегда равна единице.

Второй метод самонастройки основывается на поощрении того оператора, который доставил наибольшую суммарную пригодность на данном поколении. Пусть, как и раньше, z – число различных операторов -го типа. Начальные значения вероятностей устанавливаются $p_i = \frac{1}{z}$.Оценка эффективности каждого параметра каждого типа производится на основании усредненных значений пригодности:

$$AvgFit_i = \frac{\sum_{j=1}^{n_i} f_{ij}}{\sum_{j=1}^{n_i} 1}, i = 1,2 \dots, z$$

где n_i – количество потомков, сформированных i-м типом оператора, f_{ij} – пригодность j-го потомка, построенного i-м оператором, $AvgFit_i$ – средняя пригодность решений, построенных при помощи i-го оператора.

После этого вероятность применения оператора, чье значение $AvgFit_i$ является наибольшим среди всех операторов такого типа, увеличивается на $\frac{(z-1)K}{zN}$, а вероятности применения остальных операторов уменьшаются на $\frac{K}{zN}$, где N–число поколений генетического алгоритма, K–константа, чье значение устанавливалось равным 0.5.

Стоит отметить, что, как и в случае алгоритма PDP, следует устанавливать минимальную вероятность применения операторов. В данной работе минимальная вероятность устанавливалась равной 0.2. Таким образом, в случае, если вероятность одного из операторов падает до минимальной, то изменение вероятностей прекращается. Как только оператор с минимальной вероятностью получает наибольшее значение $AvgFit_i$, его вероятность возрастает, а вероятности остальных операторов уменьшаются.

В данной работе в генетическом алгоритме использовались по три типа операторов селекции, скрещивания и мутации, а именно: пропорциональная, ранговая и турнирная селекция с размером турнира, равным трем; одноточечное, двухточечное и равномерное скрещивание; слабая, средняя и сильная мутации.

В качестве тестовых функций были выбраны 15 функций используемых в качестве тестовых в ходе открытого сравнения оптимизаторов BlackBoxOptimizationBenchmark, в частности, функции с номерами 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 17, 18. Среди тестовых функций были функции с множеством экстремумов, функции со значительным отличием значения функции в экстремуме от остальных значений, плохо обусловленные функции и другими особенностями, которые представляют сложности при поиске экстремума. Размерность решаемой задачи также варьировалась в ходе работы алгоритма от 5 до 10 измерений.

Для определения качества работы алгоритма замерялось два значения – надежность и среднее число поколений. Надежность представляет собой отношение числа раз, когда алгоритм нашел решение с заданной точностью к общему числу запусков, которое

равнялось 100 для каждой функции. Среднее число поколений – это усредненный номер поколения, на котором было найдено решение.

Для каждой функции устанавливался свой объем вычислительных ресурсов так, чтобы надежность не была равна ни единице, ни нулю, так как в этом случае невозможно будет определить разницу между алгоритмами. Заданное число поколений и индивидов для каждой функции представлено в таблице ниже.

№ F	1	2	3	4	5	6	7	8	11	12	13	14	15	17	18
Инд.	30	50	100	100	30	100	100	100	150	100	100	100	150	75	100
Пок.	30	50	100	100	30	100	100	100	150	100	100	100	150	75	100

Для сравнения двух алгоритмов по надежности использовался непараметрическийU-критерий Манна-Уитни. Он позволяет выявлять различия в значениях параметра между двумя малыми выборками. То есть, если значение критерия меньше ли равно табличному значению (с уровнем значимости 0.05 в данной работе), то признается существенная разница между выборками, а, следовательно, и алгоритмами. Значения надежности алгоритма замерялись 10 раз для каждой функции, таким образом, каждая из выборок имела 10 измерений.Также был протестирован стандартный генетический алгоритм.

Для наглядного представления результатов тестирования в таблицах ниже было введено значение T, при этом 1 означает, что первый из алгоритмов лучше второго, 0 – алгоритмы неразличимы на данной функции, -1 – второй алгоритм лучше первого.

В таблице ниже представлены результаты сравнения второго алгоритма самонастройки с методом PDP при размерности 5.

Таблица 1

№F	1	2	3	4	5	6	7	8	11	12	13	14	15	17	18	Сумм.
T	0	-1	0	0	-1	1	0	1	0	1	1	0	1	1	1	7-2=5

Как можно видеть из таблицы, только на второй и пятой функции метод PDPпоказал лучшие результаты. На семи функциях второй метод продемонстрировал лучшие значения пригодностей. При размерности 10:

Таблица 2

№F	1	2	3	4	5	6	7	8	11	12	13	14	15	17	18	Сумм.
T	-1	-1	0	0	-1	0	1	0	0	1	0	0	1	1	1	5-3=2

Аналогичные тесты были произведены для сравнения второго метода самонастройки, который показал себя лучше, со стандартным генетическим алгоритмом на всех возможных 27 вариантах настроек. В данном случае для каждой конфигурации настроек (№Kв таблице) рассчитывалось сумма значений тестов по всем функциям.

Таблица 3

№K	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\sum T$	14	12	13	14	11	13	14	10	13	15	15	15	15	15
№K	15	16	17	18	19	20	21	22	23	24	25	26	27	Сумм
$\sum T$	15	15	15	15	11	11	1	11	11	-1	11	11	-1	234

Таким образом, в подавляющем большинстве случаев алгоритм с самонастройкой показал лучшие результаты. Стоит отметить, что стандартный алгоритм немного превзошел алгоритм с самонастройкой в комбинациях турнирной селекцией, особенно вместе с сильной мутацией.

По результатам проведенной работы можно заключить, что второй алгоритм самонастройки показал наилучшие результаты как в сравнении со стандартным алгоритмом, так и в сравнении с алгоритмом PDP, несмотря на схожую основную идею. С ростом размерности решаемой задачи оба метода самонастройки начинают показывать схожие результаты, хотя метод PDPлучше проявляет себя на более простых функциях. Таким образом, можно сделать вывод, что учет суммарного увеличения пригодности является более предпочтительным, чем подсчет числа улучшений функции пригодности. Дальнейшее повышение качества работы второго можно также добиться, варьируя

параметры, то есть минимальное значение вероятности и шаг изменения параметров. Также в дальнейшем возможна разработка схожих подходов, основанных на той же идее.