УДК 519.67

Решение экстремальных задач с помощью генетических алгоритмов

© М. Б. Беляева[1]

Аннотация. В работе рассматриваются способы решения оптимизационных задач генетическими алгоритмами. Описываются правила кодирования и определения необходимой длины хромосомы. Приведен вычислительный эксперимент, подтверждающий работоспособность ГА.
Ключевые слова: генетический алгоритм, задачи оптимизации.

1. Введение

Основной механизм эволюции - это естественный отбор. Он представляет собой процесс, приводящий к выживанию и преимущественному размножению наиболее приспособленных к данным условиям среды особей, которые приносят больше потомства, чем плохо приспособленные. А так как потомки наследуют генетическую информацию своих родителей, то потомки более сильных индивидуумов так же будут более приспособленными. Таким образом, увеличивается количество особей, обладающих наиболее полезными для выживания в данных условиях признаками, а значит, приспособленность вида в целом будет с течением времени возрастать[1] [2]. Именно на таких генетических процессах биологических организмов основаны генетические алгоритмы, предназначенные для решения задач функциональной оптимизации. Генетический алгоритм (ГА) представляет собой эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию [4].

Генетические алгоритмы являются хорошо известными на данный момент представителями эволюционных алгоритмов и по своей сути являются алгоритмами для нахождения глобального экстремума многоэкстремальной функции. ГА заключается в параллельной обработке множества альтернативных решений. При этом поиск концентрируется на наиболее перспективных из них. Это говорит о возможности использования генетических алгоритмов при решении любых задач искусственного интеллекта, оптимизации, принятия решений. Откуда следует широкое практическое применение ГА в менеджменте и управлении для решения задач поиска оптимальных решений, формирования моделей и прогнозирования значений различных показателей [3].

2. Возможности генетических алгоритмов для решения задач оптимизации

Как уже было отмечено выше, эволюция - это процесс постоянной оптимизации биологических видов. Задачи оптимизации - наиболее распространенный и важный для практики класс задач. Конечно, математики издавна занимались подобными задачами и разработали множество методов их решения. В случае, если целевая функция достаточно гладкая и имеет только один экстремум (унимодальна), то оптимальное решение можно

[1]Доцент кафедры математического моделирования, Стерлитамакская государственная педагогическая академия, г. Стерлитамак; Beljaeva_mb@rambler.ru.

получить, например, методом градиентного спуска. Недостатком градиентного алгоритма являются слишком высокие требования к функции - на практике унимодальность встречается крайне редко, а для «неправильной» функции градиентный метод часто приводит к неоптимальному ответу. Аналогичные проблемы возникают и с применением других математических методов.

Таким образом, возникает необходимость в каком-либо новом методе оптимизации, пригодном для практики. Одним из решений данной проблемы является применение механизмов эволюционного процесса. Фактически ГА организуют искусственную эволюцию в специально построенном мире.

Стандартный генетический алгоритм начинает свою работу с формирования начальной популяции $I_0 = \{i_1, i_2, \ldots, i_s\}$ - конечного набора допустимых решений задачи. Эти решения могут быть выбраны случайным образом или получены с помощью вероятностных жадных алгоритмов. Выбор начальной популяции не имеет значения для сходимости процесса в асимптотике, однако формирование «хорошей» начальной популяции (например из множества локальных оптимумов) может заметно сократить время достижения глобального оптимума.

На каждом шаге эволюции с помощью вероятностного оператора селекции выбираются два решения, родители i_1, i_2. Оператор скрещивания по решениям i_1, i_2 строит новое решение i', которое затем подвергается небольшим случайным модификациям, которые принято называть мутациями. Затем решение добавляется в популяцию, а решение с наихудшим значением целевой функции удаляется из популяции.

Среди операторов селекции наиболее распространенными являются два вероятностных оператора случайной и турнирной селекции. Турнирная селекция имеет определенные преимущества перед случайной, так как не теряет своей избирательности, когда в ходе эволюции все элементы популяции становятся примерно равными по значению целевой функции. Операторы селекции строятся таким образом, чтобы с ненулевой вероятностью любой элемент популяции мог бы быть выбран в качестве одного из родителей. Более того, допускается ситуация, когда оба родителя представлены одним и тем же элементом популяции. Как только два решения выбраны, к ним применяется вероятностный оператор скрещивания (crossover).

Существует много различных версий этого оператора [7], среди которых простейшим, по видимому, является однородный оператор. По решениям i_1, i_2 он строит решение i', присваивая каждой координате этого вектора с вероятностью 0,5 соответствующее значение одного из родителей. Если вектора i_1, i_2 совпадали скажем по первой координате, то вектор i' «унаследует» это значение. Геометрически оператор скрещивания случайным образом выбирает в гиперкубе вершину i', которая принадлежит минимальной грани, содержащей вершины i_1, i_2. Можно сказать, что оператор скрещивания старается выбрать новое решение i' где-то между i_1, i_2 полагаясь на удачу. Более аккуратная процедура скрещивания состоит в выборе такого i', которое является оптимальным решение исходной задачи на соответствующей грани гиперкуба. Отметим, что модификация решения i' может состоять не только в случайной мутации, но и в частичной перестройке решения алгоритмами локального поиска.

Применение локального спуска позволяет генетическому алгоритму сосредоточиться только на локальных оптимумах. Множество локальных оптимумов может оказаться экспоненциально большим, и на первый взгляд кажется, что такой вариант алгоритма не будет иметь больших преимуществ. Однако экспериментальные исследования распределения локальных оптимумов свидетельствуют о высокой концентрации их в непосредственной близости от глобального оптимума [6], [8]. Это наблюдение известно как гипотеза о существовании «большой долины» для задач на минимум или «центрального горного массива» для задач на максимум. Основное предположение заключается в том, что в среднем локальные оптимумы расположены гораздо ближе к глобальному оптимуму, чем к случайно выбранной точке. Их распределение в области допустимых решений не является равномерным. Они концентрируются в районе глобального оптимума, занимая область небольшого диаметра.

Эта гипотеза отчасти объясняет работоспособность генетических алгоритмов. Если в популяции собираются локальные оптимумы, которые согласно гипотезе сконцентрированы в одном месте, и очередное решение i' выбирается где-то между двумя произвольными локальными оптимумами, то такой процесс имеет много шансов найти глобальный оптимум. Аналогичные рассуждения объясняют работоспособность и других локальных алгоритмов. В связи с этим проверка и теоретическое обоснование данной гипотезы представляет несомненный интерес.

3. Постановка задачи

Рассмотрим пример применения генетического алгоритма для нахождения оптимального значения нелинейной целевой функции с ограничениями:

$$f(x_1, x_2) = (-2 * x_2^3 + 6 * x_2^2 + 6 * x_2 + 10) * \sin(\ln(x_1) * \exp(x_1^2)) \to \max, \tag{3.1}$$

$$0.5 \leq x_1 \leq 1.1; 1 \leq x_2 \leq 4.6; \tag{3.2}$$

Точность вычислений $\varepsilon = 10^{-5}$.

Поверхность целевой функции представлена на рисунке 3.1. Про анализе поведения функции видно, что ее можно отнести к классу овражных функций, для которых градиентные методы, как правило, приходят в точки локальных экстремумов.

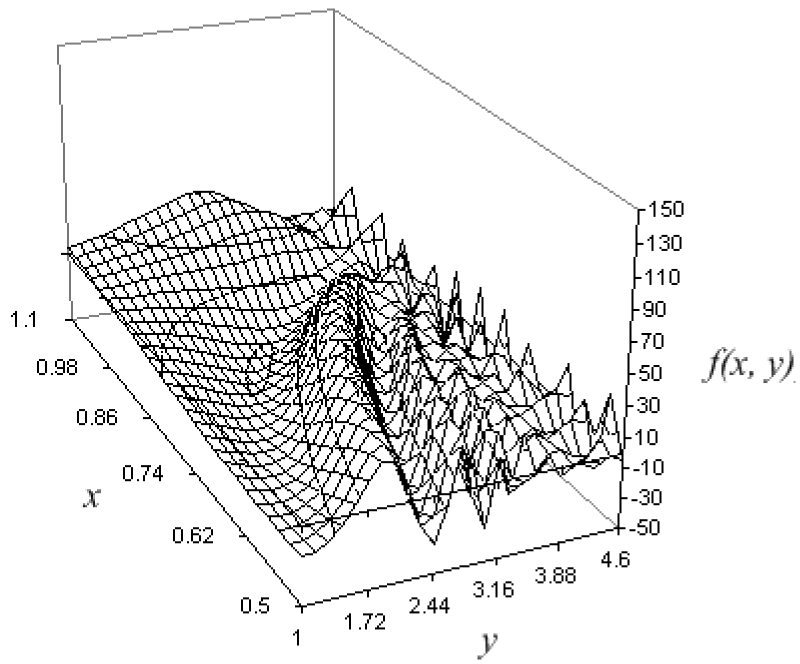

Рисунок **3.1**

Поверхность целевой функции

Максимальное значение целевая функция принимает в точке:
$x_1^* = 0.657208; x_2^* = 2.418399; \max f(x_1, x_2) = 31.31355$.

Для реализации генетического алгоритма необходимо закодировать оптимизируемые параметры в двоичные строки. Длина строки зависит от требуемой точности. Например, пусть переменная x_j имеет интервал изменения $[a_j; b_j]$. В этом случае интервал изменения переменной должен быть разбит как минимум на $(b_j - a_j) \cdot 10^5$ квантов. Требуемое число битов находится по формуле: $2^{m_{i-1}} \leq (b_j - a_j) \cdot 10^5 \leq 2^{m_i} - 1$. Обратное преобразование строки битов в действительное значение переменной x_j выполняется по формуле:

$$x_j = a_j + \textit{десятичное число (строка битов j)} \cdot \frac{b_j - a_j}{2^{m_i} - 1},$$

где *десятичное число*(*строка битов j*) представляет собой десятичное значение, закодированное в бинарной строке.

Найдем число битов, необходимых для кодирования переменных x_1 и x_2 задачи (3.1)–(3.2).

$$(1.1 - 0.5) * 100000 = 60000; 2^{15} < 60000 < 2^{16} - 1; \Rightarrow m_1 = 16;$$
$$(4.6 - 1) * 100000 = 360000; 2^{18} < 360000 < 2^{19} - 1; \Rightarrow m_2 = 19;$$

$$m = m_1 + m_2 = 16 + 19 = 35.$$

Суммарная длина хромосомы составляет 35 битов.

4. Реализация алгоритма

Исходная популяция генерируется случайно из n-го количества битовых строк.

Оценка функции соответствия хромосомы выполняется в три шага [5]:

1. Преобразовать генотип хромосомы в фенотип. В данной задаче это означает преобразование двоичной строчки в соответствующее действительное значение $x^k = (x_1^k, x_2^k)$, $k = 1, 2, .., n,$ где n - число вариантов в исходной популяции.

2. Вычислить целевую функцию $f(x^k)$.

3. Преобразовать целевую функцию в значение функции соответствия. Для решаемой задачи оптимизации функция соответствия эквивалентна целевой функции. $eval(v^k) = f(x^k), k = 1, 2, ..., n$.

Функция соответствия играет роль среды и оценивает хромосомы по степени их приспособленности к выполнению критерия оптимизации. Далее из сформированной популяции производится случайный или турнирный отбор родительских пар для скрещивания, так как новые члены популяции образуются в результате скрещивания и мутации начального набора значений. Для скрещивания хромосом используем метод с одной точкой обмена. Мутация состоит в изменении одного или большего числа генов с вероятностью, равной коэффициенту мутации. Проделав p подобных итераций, находим наилучшее решение.

Для реализации алгоритма был составлен программный комплекс средствами визуального программирования Delphi 7 для решения различных оптимизационных задач методом ГА с применением случайного и турнирного отборов.

Вычислительный эксперимент проводился при следующих параметрах:

–размер начальной популяции: 50;

–количество скрещиваний: 100;

–процент мутаций: 0,1;

–длина хромосомы: 35.

Результат работы программы приведен в таблице 1

Таблица 1. Результаты вычислений.

Параметры	ГА случайны отбор		ГА турнирный отбор	
Номер опыта	1	2	1	2
Вероятность совпадений, %:	95	100	95	100
Количество итераций	22	61	5	33
Время счета	0 с 40 мс	0 с 111 мс	0 с 130 мс	0 с 691 мс
Полученное решение:	x=1,09985 y=2,71380 f(x,y)=30,22124	x=0,68047 y=2,50293 f(x,y)=31,24433	x=0,63828 y=2,34954 f(x,y)=31,27805	x=0,64530 y=2,37812 f(x,y)=31,30063
Относительная ошибка,%:	3,49	0,22	0,11	0,04

Анализируя данные таблицы 1, можно сделать следующие выводы:

–за 100 итераций оба метода получили приближенное решение задачи (3.1)–(3.2);

–ГА турнирный находит более точное решение за 61 итерацию, затрачивая при этом 0с. 691 мс. с относительной ошибкой 0,04%;

– ГА случайный находит решение за 33 итерации, время счета - 111 мс., допустив при этом 0,22% относительной ошибки.

Таким образом можно отметить, что в данном случае ГА с турнирным отбором находит решение задачи оптимизации более эффективно.

Генетические алгоритмы являются достаточно мощным средством и могут с успехом применяться для широкого класса прикладных задач, включая те, которые трудно, а иногда и вовсе невозможно, решить другими методам. Однако, генетические алгоритмы, как и другие методы эволюционных вычислений, не гарантирует обнаружения глобального решения за полиномиальное время.

Список литературы

1. Андрейчиков, А. В. Интеллектуальные информационные системы / А. В. Андрейчиков. – Москва. – 2006. – 422 с.

2. Батищев, Д.И. Генетические алгоритмы. Решения экстремальных задач.: учеб. пособие //Воронеж, 1995. – 65 с.

3. Береснев, В. Л. Экстремальные задачи стандартизации / В. Л. Береснев, Э. Х. Гимади, В. Т. Дементьев. – Новосибирск: Наука. – 1998. – 382 с.

4. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский [и др.]. – Томск: ТГУ, 1997. – 373 с.

5. Еремеев А. В. Разработка и анализ генетических и гибридных алгоритмов для решения задач дискретной оптимизации // Омск. – 2000. – 240 с.

6. Boese K. D., Kahng A. B., Muddu S. A new adaptive multi-start technique for combinatorial global optimizations. Oper. Res. Lett. v16 (1994), N2, pp 101-114.

7. Goldberg D. E. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley. 1989.

8. Kirkpatrick S., Toulouse G. Configuration space analysis of traveling salesman problems. J. de Phys. v46 (1985) , pp 1277-1292.

The decision of extreme problems by means of genetic algorithms

© **M. B. Beljaeva**[2]

Abstract. In work ways of the decision of optimising problems are considered by genetic algorithms. Rules of coding and definition of necessary length are described Chromosomes. Computing experiment confirming is resulted Working capacity of GA.

Key Words: genetic algorithm, optimisation problems.

[2]The senior lecturer of chair of mathematical modelling, Sterlitamaksky state pedagogical academy, Sterlitamak; Beljaeva_mb@rambler.ru