

Эффективность генетических алгоритмов и возможность их использования в прогнозных моделях производственных и природных процессов

Юшкевич О.Е

- **Генетический алгоритм** — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе.
- Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и кроссинговер.
- Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

Описание алгоритма

- Задача формализуется таким образом, чтобы её решение могло быть закодировано в виде вектора («генотипа») генов, где **каждый ген может быть битом**, числом или неким другим объектом. В классических реализациях генетического алгоритма (ГА) предполагается, что генотип имеет фиксированную длину. Однако существуют вариации ГА, свободные от этого ограничения.
- Некоторым, обычно случайным, образом создаётся множество **генотипов** начальной популяции. Они оцениваются с использованием «**функции приспособленности**», в результате чего с каждым генотипом ассоциируется определённое значение («**приспособленность**»), которое определяет насколько хорошо **фенотип**, им описываемый, решает поставленную задачу

- **Генотип** — совокупность **генов** данного организма, которая, в отличие от понятия генофонд, характеризует **особь**, а не вид.
- **Фенотип** — совокупность характеристик, присущих индивиду на определённой стадии развития. Фенотип формируется на основе генотипа.

- Из полученного множества решений («поколения») с учётом значения «приспособленности» выбираются решения (обычно лучшие особи имеют большую вероятность быть выбранными), к которым применяются «генетические операторы» (в большинстве случаев «**скрещивание**» — **crossover** и «**мутация**» — **mutation**), результатом чего является получение новых решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение.

Этот набор действий повторяется **итеративно**, так моделируется «**эволюционный процесс**», продолжающийся несколько жизненных циклов (*поколений*), пока не будет выполнен **критерий остановки алгоритма**. Таким критерием может быть:

- Нахождение глобального решения;
- Исчерпание числа поколений, отпущенных на эволюцию;
- Исчерпание времени, отпущенного на эволюцию.

Таким образом, можно выделить следующие этапы генетического алгоритма:

1. Задать целевую функцию (приспособленности) для особей популяции
 2. Создать начальную популяцию
- (Начало цикла)
 1. Размножение (скрещивание)
 2. Мутирование
 3. Вычислить значение целевой функции для всех особей
 4. Формирование нового поколения (селекция)
 5. Если выполняются условия остановки, то (конец цикла), иначе (начало цикла).

Создание начальной популяции

- Перед первым шагом нужно **случайным образом** создать начальную популяцию; даже если она окажется совершенно неконкурентоспособной, вероятно, что генетический алгоритм всё равно достаточно быстро переведёт её в жизнеспособную популяцию. Таким образом, на первом шаге **можно особенно не стараться сделать слишком уж приспособленных особей**, достаточно, чтобы они соответствовали формату особей популяции, и на них можно было подсчитать функцию приспособленности (Fitness). Итогом первого шага является популяция N , состоящая из N особей.

Отбор (селекция)

- На этапе отбора нужно из всей популяции выбрать определённую её долю, которая останется «в живых» на этом этапе эволюции. Есть разные способы проводить отбор. Вероятность выживания особи h должна зависеть от значения функции приспособленности $Fitness(h)$. Сама доля выживших s **обычно является параметром** генетического алгоритма, и её просто задают заранее. По итогам отбора из N особей популяции N должны остаться sN особей, которые войдут в итоговую популяцию N' . Остальные особи погибают.

- Турнирная селекция - сначала случайно выбирается установленное количество особей (обычно две), а затем из них выбирается особь с лучшим значением функции приспособленности

- Метод рулетки - вероятность выбора особи тем вероятнее, чем лучше её значение функции приспособленности

$$p_i = \frac{f_i}{\sum_{i=1}^N f_i}$$

где p_i - вероятность выбора i особи,

f_i - значение функции приспособленности для i особи,

N - количество особей в популяции.

- Метод ранжирования - вероятность выбора зависит от места в списке особей отсортированном по значению функции приспособленности

$$p_i = \frac{1}{N} \left(a - (a - b) \frac{i - 1}{N - 1} \right),$$

где $a \in [1, 2]$, $b = 2 - a$,

i - порядковый номер особи в списке особей отсортированном по значению функции приспособленности.

- Равномерное ранжирование - вероятность выбора особи определяется выражением:

$$p_i = \begin{cases} \frac{1}{\mu}, & \text{if } 1 \leq i \leq \mu \\ 0, & \text{if } \mu \leq i \leq N \end{cases}$$

где $\mu \leq N$ параметр метода.

Выбор родителей

Размножение в генетических алгоритмах обычно половое — чтобы произвести потомка, нужны несколько родителей, обычно два.

Можно выделить несколько операторов выбора родителей:

- Панмиксия - оба родителя выбираются случайно, каждая особь популяции имеет равные шансы быть выбранной
- Инбридинг - первый родитель выбирается случайно, а вторым выбирается такой, который наиболее похож на первого родителя
- Аутбридинг - первый родитель выбирается случайно, а вторым выбирается такой, который наиболее не похож на первого родителя

Инбридинг и аутбридинг бывают в двух формах: фенотипной и генотипной. В случае фенотипной формы похожесть измеряется в зависимости от значения функции приспособленности (чем ближе значения целевой функции, тем особи похожее), а в случае генотипной формы похожесть измеряется в зависимости от представления генотипа (чем меньше отличий между генотипами особей, тем особи похожее).

Существует несколько поводов для критики насчёт использования генетического алгоритма по сравнению с другими методами оптимизации:

- Повторная оценка **функции приспособленности** для сложных проблем, часто является фактором, ограничивающим использование алгоритмов искусственной эволюции. Поиск оптимального решения для сложной задачи высокой размерности зачастую требует **очень затратной оценки функции приспособленности**. В реальных задачах, таких как задачи структурной оптимизации, единственный запуск функциональной оценки требует от **нескольких часов до нескольких дней** для произведения необходимых вычислений.

- Генетические алгоритмы **плохо масштабируемы** под сложность решаемой проблемы. Это значит, что число элементов, подверженных мутации очень велико, если велик размер области поиска решений.
- Это делает использование данной вычислительной техники чрезвычайно сложным при решении таких проблем, как, например, проектирование двигателя, дома или самолёта. Для того чтобы сделать так, чтобы такие проблемы поддавались эволюционным алгоритмам, **они должны быть разделены на простейшие представления данных проблем.**
- Таким образом, эволюционные вычисления используются, например, при разработке **формы лопастей**, вместо всего двигателя, формы здания, вместо подробного строительного проекта и формы фюзеляжа, вместо разработки вида всего самолёта.

Пример

1. Сначала пропорциональный отбор назначает каждой особи вероятность, которая равна отношению её приспособляемости к суммарной приспособляемости популяции:

$$p_i = \frac{f_i}{\sum_{i=1}^N f_i}$$

Потом происходит отбор (с замещением) всех n особей для дальнейшей генетической обработки, согласно величины P_i .

При таком отборе члены популяции с высокой приспособляемостью будут выбираться с большей вероятностью, чем особи с низкой приспособляемостью.

После отбора, n отобранных особей случайным способом разбиваются на $n/2$ пары. Для каждой пары с вероятностью P_c может выполняться кроссинговер. С вероятностью $1-P_c$ кроссинговер может не произойти и неизмененные особи переходят в стадию мутации. Если кроссинговер происходит, полученные потомки заменяют собой родителей и переходят к мутации.

2. Кросинговер это операция, при которой с двух хромосом порождается одна или несколько новых хромосом. Одноточечный кросинговер работает так: сначала, случайным образом выбирается одна с I-1 точек разрыва (участок между соседними битами в ряду).

Обе родительские структуры разрываются на два сегмента в этой точке. Потом, соответствующие сегменты разных родителей склеиваются и выходят генотипы потомков.

Например, допустим, один родитель состоит с 10 нулей, а другой с 10 единиц. Пусть с 9 возможных точек разрыва выбрана точка 3. Родители и их потомки показаны ниже.

Родитель1 0000000000 **000~0000000**--> 111~0000000 **1110000000** Потомок1

Родитель2 1111111111 **111~1111111**--> 000~1111111 **0001111111** Потомок2

После того как заканчивается стадия кросинговера, выполняются операции мутации.

3. Мутация – это преобразование хромосомы, которое случайно изменяет одну или несколько её позиций (генов). Распространенный вид мутаций - это случайная замена только одного гена с хромосомы.

В каждом рядке, который подвергается мутации, случайный бит с вероятностью P_m изменяется на противоположный.

4. Популяция, полученная после мутации заменяет старую и цикл одного поколения оканчивается. Следующие поколения обрабатываются подобным образом: отбор, кросинговер и мутация.

Пример

Рассмотрим функцию $f(x) = 2x^2 + 1$ (1.1) и предположим что x принимает целое значение с интервала от 0 до 15. Задача оптимизации этой функции состоит в перемещении в пространстве с 16 точками со значениями 0, ..., 15 для определения точки, в которой функция принимает максимальное (или минимальное) значение.

В таком случае в роли параметра задачи выступает переменная x . Множество {0, ..., 15} составляет пространство поиска и одновременно – множество потенциальных решений задачи. Каждое из 16 чисел, которые принадлежат этому множеству, называется точкой пространства поиска, решением, значение параметра, фенотипом. Решение, которое оптимизирует функцию, называется наилучшим или оптимальным решением. Значение параметра x можно закодировать следующим способом:

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

Это широко известный способ двоичного кодирования, связанный с записью десятичных чисел в двоичной системе. Представленные кодовые комбинации также называются цепочками или хромосомами. В рассматриваемом примере они выступают и в качестве генотипов. Каждая из хромосом состоит из 4х генов. Значение гена в конкретной позиции называется алелью, которая принимает в данном случае значение 0 или 1.

Популяция состоит из особей, которые выбираются из этих 16 хромосом. Примером популяции с численностью 6 может быть, например, множество хромосом {0010, 0101, 0111, 1001, 1100, 1110}, которые представляют собой закодированную форму таких фенотипов: {2, 5, 7, 9, 12, 14}. Функция приспособляемости задается таким выражением:

$$\underline{f}(x) = 2x^2 + 1.$$

Приспособляемость отдельных хромосом в популяции определяется значением этой функции для значений x , которые соответствуют этим хромосомам, то есть для фенотипов, которые соответствуют определенным генотипам.