

1.5 Элементы генетического алгоритма

Оказывается, что нет строго определения «генетического алгоритма», принимаемого всем сообществом эволюционных вычислений, которое отделяет генетические алгоритмы от других эволюционных методов вычислений. Однако же, можно сказать, что большинство методов, называемых «ГА» имеют по крайней мере следующие элементы в целом: популяцию хромосом, выбор в соответствие с пригодностью, скрещивание для производства новых потомков и произвольные мутации новых потомков. Инверсия – четвертый элемент ГА по Холланду – редко используется в современных реализациях и его польза, если это имеет место, не является хорошо обоснованной. (Инверсия будет детально рассмотрена в разделе 5)

Хромосомы в популяции генетического алгоритма типично принимают вид битовых строк. Каждая позиция в хромосоме имеет два возможных аллеля: 0 и 1. Каждая хромосома может представляться как точка в пространстве поиска возможных решений. ГА обрабатывает популяции хромосом, последовательно заменяя одну популяцию другой. Наиболее часто ГА требует фитнес-функции, которая назначает значение пригодности каждой хромосоме в текущей популяции. Пригодность хромосомы зависит от того, как хорошо хромосома решает имеющуюся проблему.

Примеры фитнес-функций

Одним из наиболее распространенных применений ГА является задача оптимизации, где целью является найти множество значений параметров, которые максимизируют, приближено, сложную многопараметрическую функцию.

$$f(y) = y + |\sin(32y)|, \quad 0 \leq y < \pi$$

(Riolo 1992). Здесь возможным решением являются значения y , которые могут быть закодированы как битовые строки, представляющие вещественные числа. Вычисление пригодности переводит заданную битовую строку x в вещественное число y и затем вычисляется значение функции для этого значения. Пригодность строки – это значение функции в этой точке.

В качестве нечислового примера, рассмотрим проблему нахождения последовательности из 50 аминокислот, которые сворачиваются в желаемую трехмерную протеиновую структуру. ГА может быть применен к этой проблеме в нахождении популяции возможных решений, каждое из которых кодируется как 50-буквенная строка, к примеру

INCCVASASDMIKPVFTVASYLKNWTKAKGPNFEICISGRTPYWDNFPGI

где каждая буква представляет одну из 20-ти возможных аминокислот. Единственный путь определить пригодность последовательности – это отрицательная величина потенциальной энергии последовательности, связанная с требуемой структурой. Потенциальная энергия показывает, как много физического сопротивления должно быть приложено для принудительного размещения в желаемой структуре – минимум потенциальной энергии, максимум пригодности. Конечно, никто не хочет физически складывать каждую последовательность в популяции в желаемую структуру и измерять ее сопротивление – это было бы очень сложно, если не невозможно. Взамен, заданная последовательность и желаемая структура (и знание некоторых из основ биофизики), позволяют оценить потенциальную энергию вычислением

некоторых сил, действующих на каждую аминокислоту, так что подсчет пригодности в целом может быть произведено численно.

Эти примеры показывают две разных ситуации, в которых возможное решение проблемы кодируется как абстрактная хромосома, закодированная как строка символов, с функцией пригодности, определенной на результирующем пространстве строк. Генетический алгоритм – это метод поиска такой фитнес-поверхности для наиболее пригодных строк.

Операторы генетического алгоритма

Наиболее простая форма генетического алгоритма включает в себя три типа операторов: отбор, скрещивание (одноточечное) и мутацию

Отбор Этот оператор выбирает хромосомы в популяции для репродукции. Чем пригодней хромосома, тем большее число раз она может выбираться для репродукции.

Скрещивание Этот оператор произвольно выбирает позицию и меняет последовательности до и после позиции между двумя хромосомами для создания двух потомков. К примеру, строки 10000100 и 11111111 могут быть скрещены после третьей позиции в каждой для производства двух потомков 10011111 и 11100100. Оператор скрещивания приблизительно копирует биологическую рекомбинацию между двумя одинарно-хромосомными (гаплоидными) организмами.

Мутация Этот оператор произвольно меняет несколько бит в хромосоме. К примеру, строка 00000100 может быть мутирована в ее второй позиции в 01000100. Мутация может происходить в каждой позиции строки с некоторой вероятностью, обычно очень малой (например, 0.001).

1.6 Простой генетический алгоритм

Задана ясно определенная проблема и битовое представление строк для возможных решений, тогда простой ГА работает следующим образом:

1. Начать с произвольно сгенерированной популяцией из n l -битных хромосом (возможных решений задачи)

2. Вычисляется пригодность $f(x)$ каждой хромосомы x в популяции.

3. Повторяются следующие шаги, пока n потомков будет создано:

- a. Выбирается пара родителей-хромосом из текущей популяции, вероятность выбора является возрастающей функцией пригодности. Выбор делается «с возвратом», означая, что та же хромосома может быть выбрана более одного раза в качестве родителя.

- b. С вероятностью p_c («вероятность скрещивания» или «степень скрещивания»), скрещиваются в паре в произвольно выбранной точке (выбранной с равномерной вероятностью) для формирования двух потомков. Если скрещивание не происходит, форма потомков в точности копирует их соответствующих родителей. (Следует обратить внимание, что здесь степень скрещивания определена как вероятность что два родителя скрещиваются в одной точке. Также имеются «многоточечные» версии скрещивания в ГА, где степень скрещивания для пары родителей является номером точки, в которой происходит скрещивание).

с. Мутация двух потомков в каждой позиции с вероятностью p_m (вероятность мутации или степень мутации) и размещение результирующей хромосомы в новой популяции.

Если n нечетное, один новый член популяции может быть произвольно отброшен.

4. Замена текущей популяции новой популяцией.

5. Переход к шагу 2.

Каждая итерация этого процесса называется генерацией. ГА обычно итерируется в пределах от 50 до 500 генераций. Полный набор генераций называется последовательностью. В конце последовательности часто имеется одна или более высоко пригодных хромосом в популяции. Так как случайность играет большую роль в каждой последовательности, две последовательности с разным произвольным началом, как правило, продемонстрируют обстоятельно разное поведение. Исследователи ГА обычно приводят статистику (такую как найденное наиболее пригодное решение в последовательности и генерацию, в которой наиболее пригодный представитель был найден) усредненную на многих различных последовательностях решения ГА той же задачи.

Простая процедура описывает только основы для большинства применений ГА. Имеется определенное число деталей, таких как размер популяции, а также вероятности скрещивания и мутации, и успех алгоритма часто сильно зависит от этих деталей. Также существует много усложненных версий ГА (к примеру, ГА, которые работают с представлениями, иными, нежели строковые, или ГА, которые имеют различные типы операторов скрещивания и мутации). Множество примеров будет дано в следующих разделах.

В качестве более подробного примера простого ГА, допустим, что l (длина строки) равняется 8, что $f(x)$ эквивалентно количеству единиц в битовой строке x (наиболее простой вариант фитнес функции, использованный в качестве наглядного примера), что n (размер популяции) равен 4, что $p_c = 0.7$, and что $p_m = 0.001$. (Как и фитнес-функция, эти значения l и n были выбраны для упрощения. Более типичные значения l и n находятся в пределах 50–1000. Значения, заданные для p_c и p_m в некоторой мере типичны.)

Начальная (случайно сгенерированная) популяция может выглядеть как эта:

Название	Строка	Приг
A	00000110	2
B	11101110	6
C	00100000	1
D	00110100	3

Распространенным методом выбора в ГА является фитнес-пропорциональный выбор, в котором ожидаемое число репродукций особи эквивалентно ее пригодности, деленной на среднее значение пригодности в популяции. (Это эквивалентно тому, что биологи называют «естественным отбором»).

Простой метод, осуществляющий фитнес-пропорциональный выбор – это «колесо рулетки» (Goldberg 1989a), который идейно эквивалентен присвоению каждой особи сектора круглого рулеточного колеса, эквивалентного по площади пригодности особи. Колесо рулетки вращается, шар останавливается на одном клиновидном секторе и соответствующая особь выбирается. В примере выше $n = 4$, колесо рулетки вращается 4 раза, первые два вращения могут выбрать хромосомы B и D в качестве родителей, и вторые два вращения могут выбрать хромосомы B и C в качестве родителей. (Факт, что A может не быть выбрана – это

просто удачный пример. Если колесо рулетки будет вращаться много раз, средние результаты будут ближе к ожидаемым значениям.)

Однажды выбранная пара родителей, с вероятностью p_c они скрещиваются для формирования двух потомков. Если они не скрещиваются, тогда потомки в точности копируют каждого родителя. Предположим, что в приведенном примере, родители В и D скрещиваются после первой битовой позиции и формируют потомков $E = 10110100$ и $F = 01101110$, и родители В и С не скрещиваются, взамен формируя потомков, которые в точности копируют В и С. Далее, каждый потомок становится объектом мутации в каждой позиции с вероятностью p_m . К примеру, предполагается, что потомок E мутирует в шестой позиции и образует $E' = 10110000$, потомки F и С не мутируют вовсе, и потомок В мутирует в первой позиции и образует $B' = 01101110$. Тогда новая популяция будет следующей:

Название	Строка	Приг
E'	10110000	3
F	01101110	5
C	00100000	1
B'	01101110	5

Следует отметить, что в новой популяции, несмотря на то, что лучшая строка была потеряна (с пригодностью 6), средняя пригодность выросла от 12/4 до 14/4. Итерация этой процедуры, в конечном счете, приведет к строке со всеми единицами.