

ТЕМА 16

16.1 Генетический алгоритм

Развитие природных систем на протяжении многих веков привлекало внимание ученых. И только изучение эволюционных принципов и генетических основ наследственности позволило им разработать как модели молекулярной эволюции, описывающие динамику изменения молекулярных последовательностей, так и макроэволюционные модели, используемые в экологии, истории и социологии для исследования экосистем и сообществ организмов. Круг задач, решаемых с помощью генетического алгоритма (ГА), очень широк. Ниже перечислены некоторые задачи, для решения которых используются ГА:

- задачи численной оптимизации;
- задачи о кратчайшем пути;
- задачи компоновки;
- составление расписаний;
- аппроксимация функций;
- отбор (фильтрация) данных;
- настройка и обучение искусственной нейронной сети;
- биоинформатика;
- игровые стратегии;
- нелинейная фильтрация;
- развивающиеся агенты/машины.

На рисунке 1 приведена общая схема генетического алгоритма.

Генетический алгоритм работает с популяцией особей, в хромосоме (генотип) каждой из которых закодировано возможное решение задачи (фенотип). В начале работы алгоритма популяция формируется случайным образом (блок «Формирование начальной популяции» на рисунке 1). Для того чтобы оценить качество закодированных решений, используют функцию приспособленности, которая необходима для вычисления приспособленности каждой особи (блок «Оценивание популяции» на рисунке 1). По результатам оценивания особей наиболее приспособленные из них выбираются (блок «Селекция» на рисунке 1) для скрещивания. В результате скрещивания выбранных особей посредством применения генетического оператора кроссинговера создается потомство, генетическая информация которого формируется в результате обмена хромосомной информацией между родительскими особями (блок «Скрещивание» на рисунке 1). Созданные потомки формируют новую популяцию, причем часть потомков мутирует (используется генетический оператор мутации), что выражается в случайном изменении их генотипов (блок «Мутация» на рисунке 1). Этап, включающий в себя последовательность «Оценивание популяции» – «Селекция» – «Скрещивание» – «Мутация», называется поколением. Эволюция популяции состоит из последовательности таких поколений.

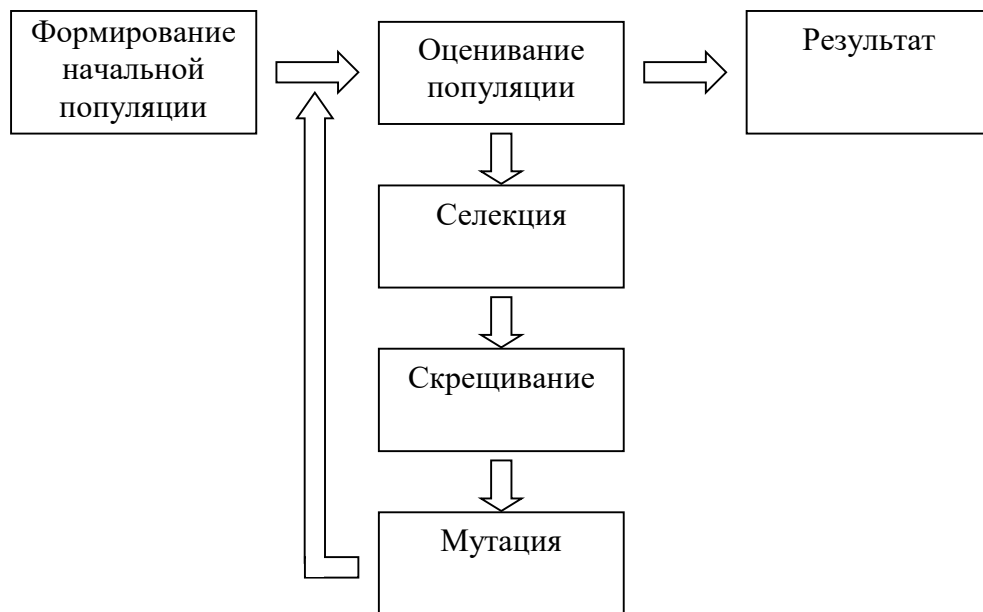


Рисунок 1 – Схема генетического алгоритма

Длительность эволюции может определяться следующими факторами:

- нахождение решения в результате эволюционного поиска;
- ограниченность количества поколений;
- ограниченность количества вычислений функции приспособленности (целевой функции);
- вырождение популяции, когда степень разнородности хромосом в популяции становится меньше допустимого значения.

Рассмотрим параметры и этапы генетического алгоритма.

16.2 Кодирование информации и формирование популяции

Выбор способа кодирования является одним из важнейших этапов при использовании эволюционных алгоритмов. В частности, должно выполняться следующее условие: возможность закодировать (с допустимой погрешностью) в хромосоме любую точку из рассматриваемой области пространства поиска. Невыполнение этого условия может привести как к увеличению времени эволюционного поиска, так и к невозможности найти решение поставленной задачи. Как правило, в хромосоме кодируются численные параметры решения. Для этого возможно использование целочисленного и вещественного кодирования.

В классическом генетическом алгоритме хромосома представляет собой битовую строку, в которой закодированы параметры решения поставленной задачи. На рисунке 2 показан пример кодирования четырех 10-разрядных параметров в 40-разрядной хромосоме.

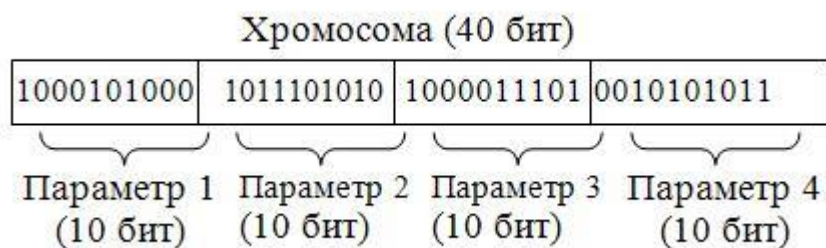


Рисунок 2 – Пример целочисленного кодирования

Обычно считают, что каждому параметру соответствует свой ген. Таким образом, хромосома на рисунке 5 состоит из четырех 10-разрядных генов. Несмотря на то, что каждый параметр закодирован в хромосоме целым числом (в виде двоичной последовательности), ему могут быть поставлены в соответствие и вещественные числа. Ниже представлен один из вариантов прямого и обратного преобразования «целочисленный ген → вещественное число».

Часто бывает удобнее кодировать в гене не целое число, а вещественное. Это позволяет избавиться от операций кодирования/декодирования, используемых в целочисленном кодировании, а также увеличить точность найденного решения. Пример вещественного кодирования представлен на рисунке 3.

Как правило, начальная популяция формируется случайным образом. При этом гены инициализируются случайными значениями.

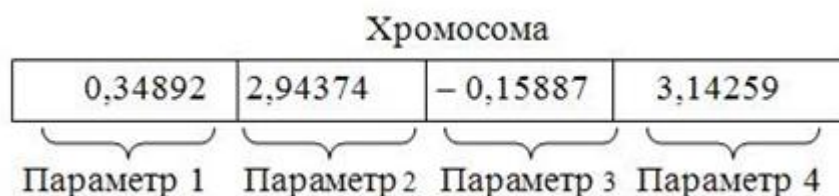


Рисунок 3 – Пример вещественного кодирования

Оценивание популяции необходимо для того, чтобы выявить в ней более приспособленные и менее приспособленные особи. Для подсчета приспособленности каждой особи используется функция приспособленности (целевая функция)

$$f_i = f(G_i),$$

где $(G_i = \{g_{ik} : k = 1, 2, \dots, N\})$ – хромосома i -й особи, g_{ik} – значение k -го гена i -й особи, N – количество генов в хромосоме.

В случае использования целочисленного кодирования для вычисления значения функции приспособленности часто бывает необходимо преобразовать закодированные в хромосоме целочисленные значения к вещественным числам. Другими словами

$$f_i = f(X_i),$$

где $(X_i = \{x_{ik} : k = 1, 2, \dots, N\})$ – вектор вещественных чисел, соответствующих генам i -й хромосомы.

Как правило, использование эволюционного алгоритма подразумевает решение задачи максимизации (минимизации) целевой функции, когда необходимо найти такие значения параметров функции f , при которых значение функции максимально (минимально). В соответствии с этим, если решается задача минимизации и $f(G_i) < f(G_j)$, то считают, что i -я особь лучше (приспособленнее) j -й особи.

В случае задачи максимизации, наоборот, если $f(G_i) > f(G_j)$, то i -я особь считается более приспособленной, чем j -я особь.

16.3 Селекция

Селекция (отбор) необходима, чтобы выбрать более приспособленных особей для скрещивания. Существует множество вариантов селекции, рассмотрим наиболее известные из них.

Рулеточная селекция. В данном варианте селекции вероятность i -й особи принять участие в скрещивании p_i пропорциональна значению ее приспособленности f_i и равна $p_i = \frac{f_i}{\sum_j f_j}$.

Процесс отбора особей для скрещивания напоминает игру в «рулетку». Рулеточный круг делится на сектора, причем площадь i -го сектора пропорциональна значению p_i . После этого n раз «вращается» рулетка, где n – размер популяции, и по сектору, на котором останавливается рулетка, определяется особь, выбранная для скрещивания.

Селекция усечением. При отборе усечением после вычисления значений приспособленности для скрещивания выбираются ln лучших особей, где l – «порог отсечения», $0 < l < 1$, n – размер популяции. Чем меньше значение l , тем сильнее давление селекции, т. е. меньше шансы на выживание у плохо приспособленных особей. Обычно выбирают l в интервале от 0,3 до 0,7.

Турнирный отбор. В случае использования турнирного отбора для скрещивания, как и при рулеточной селекции, отбираются n особей. Для этого из популяции случайно выбираются t особей, и самая приспособленная из них допускается к скрещиванию. Считается, что формируется турнир из t особей, t – размер турнира. Эта операция повторяется n раз. Чем больше значение t , тем больше давление селекции.

Вариант турнирного отбора, когда $t = 2$, называют бинарным турниром. Типичные значения размера турнира t равны 2, 3, 4, 5.

16.4 Скрещивание и формирование нового поколения

Отобранные в результате селекции особи, называемые родительскими, скрещиваются и дают потомство. Хромосомы потомков формируются в процессе обмена генетической информацией между родительскими особями. Для этого применяется оператор кроссинговера. Созданные таким образом потомки составляют популяцию следующего поколения. Будем рассматривать случай, когда из множества родительских особей случайным образом выбираются две особи и скрещиваются с вероятностью P_C , в результате чего создаются два потомка. Этот процесс повторяется до тех пор, пока не будет создано n потомков. Вероятность скрещивания P_C является одним из ключевых параметров генетического алгоритма и в большинстве случаев ее значение находится в диапазоне от 0,6 до 1.

Для кодирования часто используются 1-точечный, 2-точечный и однородный операторы кроссинговера. 1-точечный кроссинговер работает аналогично операции перекреста для хромосом при скрещивании биологических организмов. Для этого выбирается произвольная точка разрыва и для создания потомков производится обмен частями родительских хромосом. Пример работы 1-точечного кроссинговера представлен на рисунке 4,а. Для оператора 2-точечного кроссинговера выбираются две случайные точки разрыва, после чего для создания потомков родительские хромосомы обмениваются участками, лежащими между точками разрыва (рисунок 4,б). Для 2-точечного оператора кроссинговера начало и конец хромосомы считаются «склеенными», в результате чего одна из точек разрыва может попасть в начало/конец хромосом. В таком случае результат работы 2-точечного кроссинговера будет совпадать с результатом работы 1-точечного кроссинговера. На рисунке 4,б точка разрыва в месте склеивания хромосом показана пунктирными стрелками.

При использовании однородного оператора кроссинговера разряды родительских хромосом наследуются независимо друг от друга. Для этого определяют вероятность p_0 , что i -й разряд хромосомы первого родителя попадет к первому потомку, а второго родителя – ко второму потомку. Вероятность противоположного события равна $(1 - p_0)$. Каждый разряд родительских хромосом «разыгрывается» в соответствии со значением p_0 между хромосомами потомков. Чаще всего вероятность обоих событий одинакова, т. е. $p_0 = 0,5$.

Операторы кроссинговера характеризуются способностью к разрушению родительских хромосом. Кроссинговер для целочисленного кодирования считается более разрушительным, если в результате его применения различие между получившимися хромосомами потомков и хромосомами родителей велико. Это означает, что способность целочисленного кроссинговера к разрушению зависит от того, насколько сильно он «перемешивает» (рекомбинирует) содержимое родительских хромосом. Так, 1-точечный кроссинговер считается слаборазрушающим, а однородный кроссинговер в большинстве случаев является сильно

разрушающим оператором. Двухточечный кроссинговер по разрушающей способности занимает промежуточную позицию по отношению к 1-точечному и однородному операторам кроссинговера.

Одновременно со способностью к разрушению оценивается способность к созданию кроссинговером новых особей. Получается, что, разрушая хромосомы родительских особей, кроссинговер может создать новые хромосомы, не встречавшиеся ранее в процессе эволюционного поиска.

В результате скрещивания создаются потомки, которые формируют популяцию следующего поколения. Обновленная таким образом популяция не обязательно должна включать в себя одних только особей-потомков. Пусть доля обновляемых особей равна T , $0 < T < 1$, тогда в новое поколение попадает Tn потомков, где n – размер популяции, а $(1 - T)n$ особей в новой популяции являются наиболее приспособленными родительскими особями, так называемыми элитными особями. Параметр T отражает разрыв поколений. Использование элитных особей позволяет увеличить скорость сходимости генетического алгоритма.

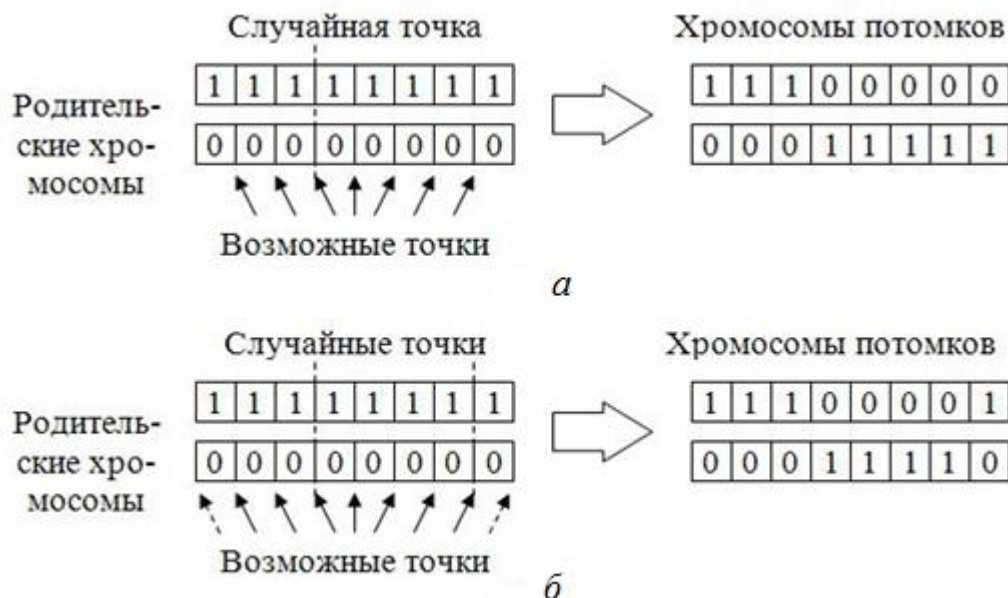


Рисунок 4 – Примеры работы кроссинговера
а) 1-точечного; б) 2-точечного

16.5 Мутация

Оператор мутации используется для внесения случайных изменений в хромосомы особей. Это позволяет «выбираться» из локальных экстремумов и тем самым эффективнее исследовать пространство поиска. Аналогично оператору кроссинговера, работа оператора мутации зависит от вероятности применения мутации P_M . Рассмотрим базовые варианты оператора мутации в зависимости от способа представления генетической информации.

Целочисленное кодирование. Одним из основных операторов мутации для целочисленного кодирования является битовая мутация. Она изменяет

отдельные разряды в хромосоме. Для этого каждый разряд инвертируется с вероятностью P_M . В силу того, что применение мутации разыгрывается столько раз, сколько разрядов содержится в хромосоме, значение P_M выбирают небольшим, чтобы сильно не разрушать найденные хорошие хромосомы. Один из типичных вариантов: $P_M = L^{-1}$, где L – длина хромосомы в битах. В этом случае каждая хромосома мутирует в среднем один раз.

Вещественное кодирование. Оператор мутации для вещественного кодирования изменяет содержимое каждого гена с вероятностью P_M , а величина изменения выбирается случайно в некотором диапазоне $[-\xi; +\xi]$, например, $[-0,5; 0,5]$, и может иметь как равномерное, так и любое другое распределение. Для того чтобы избежать сильных изменений содержимого хромосомы в результате мутации, значение вероятности P_M выбирается небольшим. Например, $P_M = N^{-1}$, где N – количество генов в хромосоме. Также возможна адаптивная настройка величины диапазона 2ξ изменения значения гена в результате мутации.

16.6 Настройка параметров генетического алгоритма

Результат работы ГА сильно зависит от того, каким образом настроены его параметры. Основными параметрами ГА являются:

- длительность эволюции (количество поколений);
- размер популяции;
- интенсивность (давление) селекции;
- тип оператора кроссинговера;
- вероятность кроссинговера P_C ;
- тип оператора мутации;
- вероятность мутации P_M ;
- величина разрыва поколений T .

Различные параметры влияют на разные аспекты эволюционного поиска, среди которых можно выделить два наиболее общих:

- 1) исследование пространства поиска;
- 2) использование найденных «хороших» решений.

Первый аспект отвечает за способности ГА к эффективному поиску решения и характеризует способности алгоритма избегать локальных экстремумов. Второй аспект важен для постепенного улучшения имеющихся результатов от поколения к поколению на основе уже найденных «промежуточных» решений.