

Дальневосточный Государственный Университет
Институт Математики и Компьютерных Наук
Кафедра Информатики

Курсовой проект

Тема:
“Генетические Алгоритмы”

Исполнил – *Студент 3-го курса*
Несов Роман Геннадьевич

Руководитель – *Ассистент кафедры информатики*
Кленин Александр Сергеевич

Владивосток 1999 г.

Содержание:

| | |
|---|----|
| 1. Естественный отбор в природе. | 3 |
| 2. Что такое генетический алгоритм. | 4 |
| 3. Подробное описание генетического алгоритма. | 6 |
| 4. Влияние параметров генетического алгоритма на эффективность поиска. | 7 |
| 5. Особенности генетических алгоритмов. | 9 |
| 6. Список литературы и ссылки | 11 |

Генетические алгоритмы - это аналитические технологии, созданные и выверенные самой природой за миллионы лет ее существования. Они позволяют решать задачи прогнозирования, классификации, поиска оптимальных вариантов, и совершенно незаменимы в тех случаях, когда в обычных условиях решение задачи основано на интуиции или опыте, а не на строгом (в математическом смысле) ее описании.

Цель данного проекта – это обзор выше упомянутой темы, для того чтоб в дальнейшем разработать систему генерирующей решение с помощью **генетических алгоритмов**. Ниже будет подробно освещена эта тема и затронуты наиболее важные аспекты этой задачи. Вначале заглянем в источник этих алгоритмов.

1

Естественный отбор в природе

Эволюционная теория утверждает, что каждый биологический вид целенаправленно развивается и изменяется для того, чтобы наилучшим образом приспособиться к окружающей среде. В процессе эволюции многие виды насекомых и рыб приобрели защитную окраску, еж стал неуязвимым благодаря иглам, человек стал обладателем сложнейшей нервной системы. Можно сказать, что эволюция - это процесс оптимизации всех живых организмов. Рассмотрим, какими же средствами природа решает эту задачу оптимизации.

Основной механизм эволюции - это естественный отбор. Его суть состоит в том, что более приспособленные особи имеют больше возможностей для выживания и размножения и, следовательно, приносят больше потомства, чем плохо приспособленные особи. При этом благодаря передаче генетической информации (генетическому наследованию) потомки наследуют от родителей основные их качества. Таким образом, потомки сильных индивидуумов также будут относительно хорошо приспособленными, а их доля в общей массе особей будет возрастать. После смены нескольких десятков или сотен поколений средняя приспособленность особей данного вида заметно возрастает.

Чтобы сделать понятными принципы работы генетических алгоритмов, поясним также, как устроены механизмы генетического наследования в природе. В каждой клетке любого животного содержится вся генетическая информация этой особи. Эта информация записана в виде набора очень длинных молекул ДНК (Дезоксирибонуклеиновая Кислота). Каждая молекула ДНК - это цепочка, состоящая из молекул нуклеотидов четырех типов, обозначаемых А, Т, С и G. Собственно, информацию несет порядок следования нуклеотидов в ДНК. Таким образом, генетический код индивидуума - это просто очень длинная строка символов, где используются всего 4 буквы. В животной клетке каждая молекула ДНК окружена оболочкой - такое образование называется хромосомой.

Каждое врожденное качество особи (цвет глаз, наследственные болезни, тип волос и т.д.) кодируется определенной частью хромосомы, которая называется геном этого свойства. Например, ген цвета глаз содержит информацию, кодирующую определенный цвет глаз. Различные значения гена называются его аллелями.

При размножении животных происходит слияние двух родительских половых клеток и их ДНК взаимодействуют, образуя ДНК потомка. Основной способ взаимодействия -

кроссовер (cross-over, скрещивание). При кроссовере ДНК предков делятся на две части, а затем обмениваются своими половинками.

При наследовании возможны мутации из-за радиоактивности или других влияний, в результате которых могут измениться некоторые гены в половых клетках одного из родителей. Измененные гены передаются потомку и придают ему новые свойства. Если эти новые свойства полезны, они, скорее всего, сохранятся в данном виде - при этом произойдет скачкообразное повышение приспособленности вида.

2 Что такое генетический алгоритм

Пусть дана некоторая сложная функция (целевая функция), зависящая от нескольких переменных, и требуется найти такие значения переменных, при которых значение функции максимально. Задачи такого рода называются задачами оптимизации и встречаются на практике очень часто.

Один из наиболее наглядных примеров - задача распределения инвестиций. В этой задаче переменными являются объемы инвестиций в каждый проект, а функцией, которую нужно максимизировать - суммарный доход инвестора. Также даны значения минимального и максимального объема вложения в каждый из проектов, которые задают область изменения каждой из переменных.

Попытаемся решить эту задачу, применяя известные нам природные способы оптимизации. Будем рассматривать каждый вариант инвестирования (набор значений переменных) как индивидуума, а доходность этого варианта - как приспособленность этого индивидуума. Тогда в процессе эволюции (если мы сумеем его организовать) приспособленность индивидуумов будет возрастать, а значит, будут появляться все более и более доходные варианты инвестирования. Остановив эволюцию в некоторый момент и выбрав самого лучшего индивидуума, мы получим достаточно хорошее решение задачи. Генетический алгоритм - это простая модель эволюции в природе, реализованная в виде компьютерной программы. В нем используются как аналог механизма генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде.

Вот как моделируется генетическое наследование:

| | |
|-------------------------------|---|
| Хромосома | Вектор (последовательность) из нулей и единиц. Каждая позиция (бит) называется геном. |
| Индивидуум = генетический код | Набор хромосом = вариант решения задачи. |
| Кроссовер | Операция, при которой две хромосомы обмениваются своими частями. |
| Мутация | Случайное изменение одной или нескольких позиций в хромосоме. |

Чтобы смоделировать эволюционный процесс, сгенерируем вначале случайную популяцию - несколько индивидуумов со случайным набором хромосом (числовых векторов). Генетический алгоритм имитирует эволюцию этой популяции как циклический процесс скрещивания индивидуумов и смены поколений.

Жизненный цикл популяции - это несколько случайных скрещиваний (посредством кроссовера) и мутаций, в результате которых к популяции добавляется какое-то количество новых индивидуумов.



Отбор в генетическом алгоритме - это процесс формирования новой популяции из старой, после чего старая популяция погибает. После отбора к новой популяции опять применяются операции кроссовера и мутации, затем опять происходит отбор, и так далее. Отбор в генетическом алгоритме тесно связан с принципами естественного отбора в природе следующим образом:

| | |
|------------------------------------|--|
| Приспособленность индивидуума | Значение целевой функции на этом индивидууме. |
| Выживание наиболее приспособленных | Популяция следующего поколения формируется в соответствии с целевой функцией. Чем приспособленнее индивидуум, тем больше вероятность его участия в кроссовере, т.е. размножении. |

Таким образом, модель отбора определяет, каким образом следует строить популяцию следующего поколения. Как правило, вероятность участия индивидуума в скрещивании берется пропорциональной его приспособленности. Часто используется так называемая стратегия элитизма, при которой несколько лучших индивидуумов переходят в следующее поколение без изменений, не участвуя в кроссовере и отборе. В любом случае каждое следующее поколение будет в среднем лучше предыдущего. Когда приспособленность индивидуумов перестает заметно увеличиваться, процесс останавливают и в качестве решения задачи оптимизации берут наилучшего из найденных индивидуумов.

Возвращаясь к задаче оптимального распределения инвестиций, поясним особенности реализации генетического алгоритма в этом случае.

- Индивидуум = вариант решения задачи = набор из 10 хромосом X_j
- Хромосома X_j = объем вложения в проект $j = 16$ -разрядная запись этого числа
- Так как объемы вложений ограничены, не все значения хромосом являются допустимыми. Это учитывается при генерации популяций.
- Так как суммарный объем инвестиций фиксирован, то реально варьируются только 9 хромосом, а значение 10-ой определяется по ним однозначно.

3 Подробное описание генетического алгоритма

1. Создание структуры решения искомой задачи в виде массива $a[i]$, $i = 1, \dots, n$, где n - максимальное число компонент структуры. Пример: поиск функции $y=f(x)$ наилучшего в классе полиномов приближения экспериментальных точек $\{x_i, y_i\}$, $j=1, \dots, m$. Структура определяется битовым массивом, где каждому элементу массива сопоставлен простейший многочлен типа x^i , $i=1, \dots, n$, где n - максимальная степень полинома.

2. Создание показателя эффективности структуры, заполненной конкретными значениями. Пример: Показателем эффективности для нашего примера будет невязка определенная методом наименьших квадратов $J_a = I_1 + I_2 + \dots + I_m$, где $I_j = (y_j - f_a(x_j))^2$, где $f_a(x)$ есть сумма всех элементов вида $a_i x^i$, где $a_i = 0$ или 1

3. Задание некоторого массива различных структур S_k , $k=1, \dots, N$, размерностью N , большей, чем число компонент n в структуре. Данный массив можно сгенерировать случайно, задав нули и единицы в каждой структуре.

4. Расчет показателей эффективности J_k для каждой структуры S_k . По формуле заданной в пункте 2.

5. Естественный отбор структур по некоторому правилу выбора наилучших структур среди заданного массива структур. Пример: можно по правилу вида $J_0 = M(J_k)$ - среднее значение J_k , если $J_k < J_0$, то структура остается, иначе умирает. Если разница между предыдущим J_0 и новым J_0 меньше какого-то малого числа, то **конец расчета**.

6. Замена выбывших структур на новые, рожденные от наиболее приспособленных структур с помощью генетических операторов

а.) мутация - замена в структуре одного из значений случайно выбранной компоненты
Пример: из (1, 1, 0, 1, 0, 0, 1, 0) получится (1, 1, 0, 1, 1, 0, 1, 0).

б.) инверсия - перестановка в структуре некоторой ее части наоборот
Пример: из (1, 1, 0, 1, 0, 0, 1, 0) получится (1, 1, 0, 0, 1, 0, 1, 0).

в.) кроссинговер - создание структуры, основанной на двух структурах - заменой одной части первой структуры на ту же область во второй.
Пример: из (A, B, C, D, E) и (a, b, c, d, e) получится (A, B, c, d, E).

7. Переход к этапу 4.

Операторы кроссовера и мутации

Наиболее традиционным подходом является отход от традиционной схемы "размножения", используемой в большинстве реализованных ГА-мах и повторяющих классическую схему. Классическая схема предполагает ограничение численности потомков путем использования так называемой вероятности кроссовера. Такая модель придает величине, соответствующей численности потомков, вообще говоря, недетерминированный характер. Есть метод предлагающий отойти от вероятности кроссовера и использовать фиксированное число брачных пар на каждом поколении, при этом каждая брачная пара "дает" двух потомков. Такой подход хорош тем, что делает процесс поиска более управляемым и предсказуемым в смысле вычислительных затрат.

В качестве генетических операторов получения новых генотипов "потомков", используя генетическую информацию хромосомных наборов родителей мы применяются два типа кроссоверов - одно- и двухточечный. Вычислительные эксперименты показали, что даже для простых функций нельзя говорить о преимуществе того или иного оператора. Более того было показано, что использование механизма случайного выбора одно- или двухточечного кроссовера для каждой конкретной брачной пары подчас оказывается более эффективным, чем детерминированный подход к выбору кроссоверов, поскольку достаточно трудно определить который из двух операторов более подходит для каждого конкретного ландшафта приспособленности. Использование же случайного выбора преследовало целью прежде всего сгладить различия этих двух подходов и улучшить показатели среднего ожидаемого результата. Для всех представленных тестовых функций так и произошло, - случайного выбор оказался эффективнее худшего.

Повышение эффективности поиска при использовании случайного выбора операторов кроссовера повлияло на то, чтобы применить аналогичный подход при реализации процесса мутагенеза новых особей, однако в этом случае преимущество перед детерминированным подходом не так очевидно в силу традиционно малой вероятности мутации. В основном вероятность мутации составляет 0.001 - 0.01.

Выбор родительской пары

Первый подход самый простой - это случайный выбор родительской пары ("*панмиксия*"), когда обе особи, которые составят родительскую пару, случайным образом выбираются из всей популяции, причем любая особь может стать членом нескольких пар. Несмотря на простоту, такой подход универсален для решения различных классов задач. Однако он достаточно критичен к численности популяции, поскольку эффективность алгоритма, реализующего такой подход, снижается с ростом численности популяции. Второй способ выбора особей в родительскую пару - так называемый *селективный*. Его суть состоит в том, что "родителями" могут стать только те особи, значение приспособленности которых не меньше среднего значения приспособленности по популяции, при равной вероятности таких кандидатов составить брачную пару. Такой подход обеспечивает более быструю сходимость алгоритма. Однако из-за быстрой сходимости селективный выбор родительской пары не подходит тогда, когда ставится задача определения нескольких экстремумов, поскольку для таких задач алгоритм, как правило, быстро сходится к одному из решений. Кроме того, для некоторого класса задач

со сложным ландшафтом приспособленности быстрая сходимость может превратиться в преждевременную сходимость к квазиоптимальному решению. Этот недостаток может быть отчасти компенсирован использованием подходящего механизма отбора (о чем будет сказано ниже), который бы "тормозил" слишком быструю сходимость алгоритма. Другие два способа формирования родительской пары, на которые хотелось бы обратить внимание, это *инбридинг* и *аутбридинг*. Оба эти метода построены на формировании пары на основе близкого и дальнего "родства" соответственно. Под "родством" здесь понимается расстояние между членами популяции как в смысле геометрического расстояния особей в пространстве параметров. В связи с этим будем различать генотипный и фенотипный (или географический) инбридинг и аутбридинг. Под инбридингом понимается такой метод, когда первый член пары выбирается случайно, а вторым с большей вероятностью будет максимально близкая к нему особь. Аутбридинг же, наоборот, формирует брачные пары из максимально далеких особей. Использование генетических инбридинга и аутбридинга оказалось более эффективным по сравнению с географическим для всех тестовых функций при различных параметрах алгоритма. Наиболее полезно применение обоих представленных методов для многоэкстремальных задач. Однако два этих способа по-разному влияют на поведение генетического алгоритма. Так инбридинг можно охарактеризовать свойством концентрации поиска в локальных узлах, что фактически приводит к разбиению популяции на отдельные локальные группы вокруг подозрительных на экстремум участков ландшафта, напротив аутбридинг как раз направлен на предупреждение сходимости алгоритма к уже найденным решениям, заставляя алгоритм просматривать новые, неисследованные области.

Механизм отбора

Обсуждение вопроса о влиянии метода создания родительских пар на поведение генетического алгоритма невозможно вести в отрыве от реализуемого механизма отбора при формировании нового поколения. Наиболее эффективные два механизма отбора элитный и отбор с вытеснением.

Идея элитного отбора, в общем, не нова, этот метод основан на построении новой популяции только из лучших особей репродукционной группы, объединяющей в себе родителей, их потомков и мутантов. В основном это объясняют потенциальной опасностью преждевременной сходимости, отдавая предпочтение пропорциональному отбору. Быстрая сходимость, обеспечиваемая элитным отбором, может быть, когда это необходимо, с успехом компенсирована подходящим методом выбора родительских пар, например аутбридингом. Именно такая комбинация "аутбридинг - элитный отбор" является одной из наиболее эффективных.

Второй метод, на котором хотелось бы остановиться, это отбор с вытеснением. Будет ли особь из репродукционной группы заноситься в популяцию нового поколения, определяется не только величиной ее приспособленности, но и тем, есть ли уже в формируемой популяции следующего поколения особь с аналогичным хромосомным набором. Из всех особей с одинаковыми генотипами предпочтение сначала, конечно же, отдается тем, чья приспособленность выше. Таким образом, достигаются две цели: во-первых, не теряются лучшие найденные решения, обладающие различными хромосомными наборами, а во-вторых, в популяции постоянно поддерживается достаточное генетическое разнообразие. Вытеснение в данном случае формирует новую популяцию скорее из далеко расположенных особей, вместо особей, группирующихся около текущего найденного решения. Этот метод особенно хорошо себя показал при

решении многоэкстремальных задач, при этом помимо определения глобальных экстремумов появляется возможность выделить и те локальные максимумы, значения которых близки к глобальным.

5 Особенности генетических алгоритмов

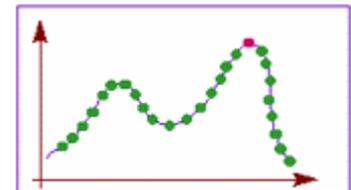
Генетический алгоритм - новейший, но не единственно возможный способ решения задач оптимизации. С давних пор известны два основных пути решения таких задач - переборный и локально-градиентный. У этих методов свои достоинства и недостатки, и в каждом конкретном случае следует подумать, какой из них выбрать.

Рассмотрим достоинства и недостатки стандартных и генетических методов на примере классической задачи коммивояжера. Суть задачи состоит в том, чтобы найти кратчайший замкнутый путь обхода нескольких городов, заданных своими координатами. Оказывается, что уже для 30 городов поиск оптимального пути представляет собой сложную задачу, побудившую развитие различных новых методов (в том числе нейросетей и генетических алгоритмов).

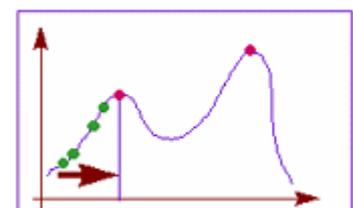


Каждый вариант решения (для 30 городов) - это числовая строка, где на j -ом месте стоит номер j -ого по порядку обхода города. Таким образом, в этой задаче 30 параметров, причем не все комбинации значений допустимы. Естественно, первой идеей является полный перебор всех вариантов обхода.

Переборный метод наиболее прост по своей сути и тривиален в программировании. Для поиска оптимального решения (точки максимума целевой функции) требуется последовательно вычислить значения целевой функции во всех возможных точках, запоминая максимальное из них. Недостатком этого метода является большая вычислительная стоимость. В частности, в задаче коммивояжера потребуется просчитать длины более 10^{30} вариантов путей, что совершенно нереально. Однако, если перебор всех вариантов за разумное время возможен, то можно быть абсолютно уверенным в том, что найденное решение действительно оптимально.



Второй популярный способ основан на методе градиентного спуска. При этом вначале выбираются некоторые случайные значения параметров, а затем эти значения постепенно изменяют, добиваясь наибольшей скорости роста целевой функции. Достигнув локального максимума, такой алгоритм останавливается, поэтому для поиска глобального оптимума потребуются дополнительные усилия.



Градиентные методы работают очень быстро, но не гарантируют оптимальности найденного решения. Они идеальны для применения в так называемых *униmodalных* задачах, где целевая функция имеет единственный локальный максимум (он же - глобальный). Легко видеть, что задача коммивояжера *униmodalной* не является.



Типичная практическая задача, как правило, мультимодальна и многомерна, то есть содержит много параметров. Для таких задач не существует ни одного универсального метода, который позволял бы достаточно быстро найти абсолютно точное решение.

Однако, комбинируя переборный и градиентный методы, можно надеяться получить хотя бы приближенное решение, точность которого будет возрастать при увеличении времени расчета.

Генетический алгоритм представляет собой именно такой комбинированный метод. Механизмы скрещивания и мутации в каком-то смысле реализуют переборную часть метода, а отбор лучших решений - градиентный спуск.

На рисунке показано, что такая комбинация позволяет обеспечить устойчиво хорошую



эффективность генетического поиска для любых типов задач.

Итак, если на некотором множестве задана сложная функция от нескольких переменных, то генетический алгоритм - это программа, которая за разумное время находит точку, где значение функции достаточно близко к максимально возможному. Выбирая приемлемое

время расчета, мы получим одно из лучших решений, которые вообще возможно получить за это время.

Литература и Ссылки:

- Генетические алгоритмы по-русски <http://www.chat.ru/~saisa>
- НейроПроект.** Аналитические технологии XXI века <http://www.neuroproject.ru>
- Научное издательство ТВП <http://www.tvp.ru/mathem3.htm>
- Факультет вычислительной математики и кибернетики МГУ (ВМиК)
http://cmc.cs.msu.su/labs/lvk/materials/tez_sapr99_1.html
- Neural Bench Development <http://www.neuralbench.ru/rus/theory/genetic.htm>
- Журнал "Автоматизация Проектирования" <http://www.opensystems.ru/ap/1999/01/08.htm>
- (EHIPS) Генетические алгоритмы <http://www.iki.rssi.ru/ehips/genetic.htm>
- SENN Генетические Алгоритмы http://fdmhi.mega.ru/ru/senn_ga.htm
- Генетические алгоритмы и машинное обучение
http://www.math.tsu.ru/russian/center/ai_group/ai_collection/docs/faqs/ai/part5/faq3.html
- Компьютерра | 11/1999 | Генетические алгоритмы: почему они работают?
http://www.vspu.ru/public_html/cterra/20.html
- Лекции по нейронным сетям и генетическим алгоритмам
<http://infoart.baku.az/inews/30000007.htm>
- @lgorithms: Catalogue of sources. <http://algo.ekaboka.com/algo-rus/index.htm>