«Генетический алгоритм»

**РЕФЕРАТ**

Выпускная квалификационная работа: 42 с., 5 главы, 17 рис., 11 табл., 13 слайдов, 18 источников литературы.

Объект исследования: генетический алгоритм.

Цель работы: изучение генетического алгоритма, основания его появления, связи биологических примеров и математических обоснований, применение в решении задач.

В выпускной работе проводится изучение генетического алгоритма, история его формирования. А также развития алгоритма, его основные операторы достоинства и недостатки. В работе приводятся примеры использования данного алгоритма.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ, ЗАДАЧА ОПТИМИЗАЦИИ, ГЕНЕТИЧЕСКИЙ ОПЕРАТОР, МУТАЦИЯ, КРОССИНГОВЕР, МЕХАНИЗМ ОТБОРА, ДИАФАНТОВО УРАВНЕНИЕ, ЗАДАЧА КОММЕВОЯЖЕРА.

Оглавление

[Введение 4](#_Toc484096781)

[1. Эволюция в природе 6](#_Toc484096782)

[1.1 Естественный отбор 6](#_Toc484096783)

[1.2 Задача оптимизации 7](#_Toc484096784)

[2. Генетические алгоритмы 9](#_Toc484096785)

[2.1 История развития генетических алгоритмов 9](#_Toc484096786)

[2.2 Общий вид генетического алгоритма 10](#_Toc484096787)

[2.3 Генетические операторы 12](#_Toc484096788)

[2.4 Достоинства и недостатки стандартных и генетических методов 17](#_Toc484096789)

[3. Влияние параметров генетического алгоритма на эффективность поиска 23](#_Toc484096790)

[3.1 Операторы кроссовера и мутации 23](#_Toc484096791)

[3.2 Выбор родительской пары 23](#_Toc484096792)

[3.3 Механизм отбора 25](#_Toc484096793)

[4. Наиболее актуальные проблемы генетического алгоритма и интересные модификации 26](#_Toc484096794)

[5. Пример. 28](#_Toc484096795)

[5.1 Решение Диофантова уравнения 28](#_Toc484096796)

[5.2 Вариация коммивояжёра 30](#_Toc484096797)

[5.3 Простой пример генетического алгоритма 33](#_Toc484096798)

[Заключение 38](#_Toc484096799)

[Список использованной литературы 39](#_Toc484096800)

[Приложение 1 40](#_Toc484096801)

[Приложение 2 41](#_Toc484096802)

# Введение

Смысл генетических алгоритмов подсмотрен у окружающей нас живой природы и заключается в организации эволюционного процесса, суть которого заключена в нахождении оптимального решения в непростой комбинаторной задаче. Разработчик, выступающий в качестве "создателя", генетических алгоритмов должен точнее установить правила эволюции, чтобы получить требующийся результат, добиться поставленной цели, как можно быстрее.

Генетическими алгоритмами называют методы оптимизации многоэкстремальных функций в виду того, что качество решения обычно оценивается некоторой оценочной функцией. В решаемой генетическим алгоритмом задаче дополнительной информации, кроме самой задачи больше нет. В эволюционном процессе популяция сама вырабатывает необходимые для выживания и приспособления качества, которые одновременно являются оптимальным решением.

В решении большинства NP-трудных задач, особенно где математические модели имеют сложную структуру и применение стандартных методов типа ветвей и границ, динамического или линейного программирования крайне затруднено генетические алгоритмы, на сегодняшний день, доказали свою конкурентоспособность. 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.

Простая модель эволюции в природе, реализованная в виде компьютерной программы, называется генетическим алгоритмом. В нем используются как аналог естественного отбора, так и аналог механизма генетического наследования. При этом в упрошенном виде сохраняется биологическая терминология. Представим себе искусственный мир, населенный множеством существ (особей), причем каждое существо — это некоторое решение нашей задачи. Более приспособленной особью будем читать ту особь, которая лучше соответствует решения (чем большее значение целевой функции оно дает). Тогда задача максимизации целевой функции сводится к поиску наиболее приспособленного существа. Естественно, мы не можем поселить в нашем выдуманном (виртуальном) мире все существа сразу, поскольку их очень много. Вместо этого мы будем изучать несколько поколений, сменяющихся друг другом. Тогда, если мы путем естественного отбора и генетического наследования получим мир, который будет подчинен законом эволюции. Заметим, что создание наилучших решений, в соответствии с нашим определением приспособленности, является целью этой искусственной эволюции. Очевидно, эволюция — бесконечный процесс, в ходе которого приспособленность особей постепенно повышается. Принудительно остановив этот процесс через достаточно долгое время после его начала и выбрав наиболее приспособленную особь в текущем поколении, мы получим не абсолютно точный, но близкий к оптимальному ответ. Такова, вкратце, идея генетического алгоритма.

1. **Эволюция в природе**

**1.1 Естественный отбор**

Возможно заявить, эволюция - это процесс оптимизации всех живых организмов. Главную роль в теории эволюции представляет естественный отбор. Его сущность заключается в том, что более адаптированные особи лучше выносят все тяготы жизни и приносят значительно больше потомства, нежели наименее адаптированные. Подметим, собственно сам по себя естественный отбор еще не гарантирует становления биологического вида. Несомненно, при предположить, что всегда потомки рождаются приблизительно неизменными, то разные поколения будут отличаться лишь численностью, однако не по приспособленности. Следовательно, весьма значительно изучить, каким способом происходит наследование, т. е. каким образом свойства потомка зависят от свойств предка. Главной закон наследования интуитивно ясен каждому — он состоит в том, что потомки похожи на предков. А именно, отпрыски более приспособленных предков будут наиболее приспособленными в своем поколении. Дабы понять, на чем основана эта похожесть, нам потребуется малость углубиться в устройство животной клетки — в гены и хромосомы. Примерно в каждой клетке любого животного имеется набор хромосом, несущих информацию относительно этого животного. Главная доля хромосомы — фибра ДНК (молекула дезоксирибонуклеиновой кислоты), которая состоит из четырех видов специальных соединений — нуклеотидов, идущих в известной последовательности. Нуклеотиды обозначаются рунами A, T, C и G, и именно последовательность их расположения всегда кодирует генетические свойства данного организма. Глася более правильно, ДНК предопределяет, какие химические реакции будут протекать в данной клетке, как она будет жить и какие функции исполнять. Ген — это часть цепи ДНК, определяющая качество особи, например, окраска глаз, вид волос, окраска кожи и т.д. Вся сумма генетических признаков человека кодируется через приблизительно 60 тыс. генов, суммарная длина которых - более 90 млн. нуклеотидов. Отличают два типа клеток: половые (такие, как сперматозоид и яйцеклетка) и соматические. В каждой соматической клетке человека содержится 46 хромосом. Эти 46 хромосом — это 23 пары, причем в каждой паре одна из хромосом — отцовская, а вторая — материнская. Парные хромосомы соответствуют одному признаку — предположим, отцовская хромосома может иметь ген черной окраски глазища, а парная ей материнская — ген голубоглазой окраски. Есть определенные законы, определяющие участие всех генов в организме, которые учувствуют в его развитии. Подробнее, в нашем примере потомок станет черноглазым, потому что ген голубых глаз является «слабым» (рецессивным) и подавляется геном любого другого цвета.

23 хромосомы хранятся в половых клетках, и они непарные. При оплодотворении происходит слияние женских и мужских половых клеток и образуется новая клетка (ее называют зародышем), которая уже содержит 46 хромосом. Какие отцовские и материнские свойства получит клетка, зависит от того, какие клетки учувствовали при оплодотворении. Дело в том, что процесс мейоза (выработки половых клеток в организме) случаен, благодаря этому потомки отличаются от предков. При выработке половых клеток организме происходит следующее: парные хромосомы соматической клетки сближаются вплотную, затем их нити ДНК разрываются в нескольких случайных местах и хромосомы обмениваются своими частями.

Благодаря данному процессу происходит появление новых разновидностей хромосом, называемое «кроссинговер». Любая из вновь появившихся таким образом хромосом оказывается затем внутри одной из половых клеток, и ее генетическая информация может проявиться в потомках рассматриваемой особи. Второй важный фактор, влияющий на наследственность, — это мутации, которые выражаются в изменении некоторых участков ДНК. Мутации также случайны и могут быть вызваны различными внешними факторами, например, радиоактивное облучение.

**1.2 Задача оптимизации**

Как уже было оговорено ранее, эволюция – это процесс постоянной оптимизации биологического вида. Теперь мы знаем, как это происходит. Естественный отбор гарантирует, что сильнейшие особи дадут большее количество потомства, а благодаря генетическому наследованию, можно быть уверенным, что некоторые потомки не только сохранят лучшие качества родителей, но и будут обладать новыми свойствами. Если эти свойства окажутся необходимыми для выживания, то вероятность того, что они передадутся следующему поколению, будет весьма велика. Подобным способом происходит улучшение приспособленности вида к окружающей среде и накопление полезных качеств вида. Понимая, как происходит оптимизация вида в природе, мы можем применить схожий метод решения различных оптимизационных задач.

Задачи оптимизации, как самые распространенные, приходится решать каждому, например, в быту, распределение времени между различными делами, на работе, добиться максимального дохода компании или максимальной скорости работы программного кода. Среди таких задач имеются задачи, решение которых просто, но встречаются и такие, решение которых найти крайне затруднительно. Приведем несколько примеров и введем пару обозначений. Как правило, в задаче оптимизации мы можем распоряжаться несколькими параметрами (обозначим их значения через , , ..., ), а нашей целью является максимизация (или минимизация) некоторой функции, f(, , ..., ), зависящей от этих параметров. Функция f называется целевой функцией. Например, если требуется максимизировать целевую функцию «доход компании», то изменяемыми параметрами будут затраты на рекламу, объем производства, число сотрудников компании, цены на конечные продукты и т. д. Стоит сказать, что данные параметры связаны между друг с другом — в частности, при уменьшении числа сотрудников скорее всего упадет и объем производства. Естественно, математики давно занимается подобными задачами и разработала несколько методов решения. В случае если целевая функция достаточно гладкая и имеет только один локальный максимум (унимодальная), то оптимальное решение можно получить методом градиентного спуска. Суть данного метода заключается в том, что оптимальное решение получается итерациями. Берется случайная начальная точка, а затем на малый шаг происходит ее сдвиг, причем шаг делается в том направлении, в котором целевая функция растет быстрее всего, затем процесс повторяется снова. 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. Похожие проблемы возникают и с применением других математических методов. Во многих важных задачах параметры могут принимать лишь определенные значения, причем во всех остальных точках целевая функция не определена. Безусловно, в данном случае речи о гладкости функции быть не может, и необходимы принципиально иные методы решения. Таким образом, появляется необходимость в создании принципиально нового метода оптимизации решения, пригодного для практического применения. Примером такого метода могут послужить генетические алгоритмы.

1. **Генетические алгоритмы**

**2.1 История развития генетических алгоритмов**

Впервые генетический алгоритм подробно описал и исследовал в своей работе де Джонг, сейчас это явление носит название стандартного генетического алгоритма. В своем исследовании де Джонг проанализировал схему кодирования генов с помощью битовых строк фиксированной длины и генетических операторов, выполнил большое количество численных экспериментов, сравнил результаты с оценками, предсказываемыми теоремой Холланда. Важность этой работы определяется прежде всего ее строгим и системным характером, ясностью изложения и представления результатов, выбранным уровнем абстракции и упрощения, позволяющим избежать ненужной детализации. Помимо этого, в работе подчеркнута необходимость дальнейшего исследования более сложный представлений генотипов и операций над ними. Последующие работы де Джонга продемонстрировали его интерес к проблемам, которые требовали менее структурированных схем представления гена.

Другой отправной точкой развития исследований по генетическим алгоритмам была публикация в 1975 году книги Холланда "Adaptatiоn in Natural and Artificial Systems". Данная книга послужила теоретический фундамент для исследования де Джонга и всех следующих его работ. В книге Холланда математически обоснована идея близких подмножеств внутри генотипа, процесса гибели и селективного воспроизведения. Цели этой книги были настолько широки, что послужили толчком дальнейшего изучения этого вопроса. Хотя, в работе, доказательства были основаны на генах фиксированной длины и соответствующих простых операторах, это не ограничило дальнейшее представление подхода на случаи более богатые по возможностям. Например, Холланд описал возможность применения операторов внутри хромосомной дупликации, удаления и сегрегации участков хромосомы.

Тем не менее, следующие исследователи восприняли теоретические результаты, содержащиеся в этих работах слишком буквально, и разрабатывали различные модификации стандартного генетического алгоритма де Джонга.

Исследование Кавичио, нацеленное на разработку детектора распознавания образов было одним из первых, использующих гены переменной длины. Расположенные на плоскости двоичные пиксели, способные различать свет и темноту, нумеровались и объединялись в подмножества, образующие разные детекторы, которые затем использовались некоторым априорно заданным алгоритмом для распознавания образов. При этом не возникало использование одного и того же пикселя дважды или не использования пикселя.

Смит использовал правила, оперирующие строками переменной длины, создавая модель машинного игрока в покер. Его LS-1 система использовала модифицированный кроссинговер, который располагал точки кроссинговера как на границах строк, кодирующих правила, так и внутри них. Он также включил в алгоритм оператор инверсии, надеясь, что это поможет более эффективно распределять правила внутри строки.

Несколько ранее Смита, Франц рассмотрел возможность смещения отдельных подмножеств битов внутри строки в своей работе по нелинейной оптимизации. К сожалению, из-за неудачного выбора вида функции оптимизации (которая легко оптимизировалась и простым генетическим алгоритмом без использования инверсии) не было показано реальных преимуществ нового алгоритма по сравнению с общепринятым. Тем не менее, эту работу можно считать одной из первых, начавших применение более гибких схем генетических алгоритмов.

Другие исследователи предложили различные модификации схем генетического переупорядочивания внутри строки, когда используется кроссинговер. Одновременно был проведен ряд исследований, развивавших теорию Холланда, и продемонстрированы новые генетические операторы. Однако вопрос об их преимуществах и тщательном сравнении со стандартным генетическим алгоритмом продолжает оставаться открытым.

**2.2 Общий вид генетического алгоритма**

Вначале необходимо сгенерировать определенное количество возможных решений (чаще всего они определяются случайным образом, для определения первой популяции). Так же необходимо задать функцию оптимальности f(x), определяющую эффективность каждого найденного решения, для отсеивания решений, вышедших за допустимую область. Каждое решение кодируется как вектор x, называемый хромосомой. Аллели – это гены (элементы хромосом), изменяющиеся в определенных позициях. Генотип – значение генома.

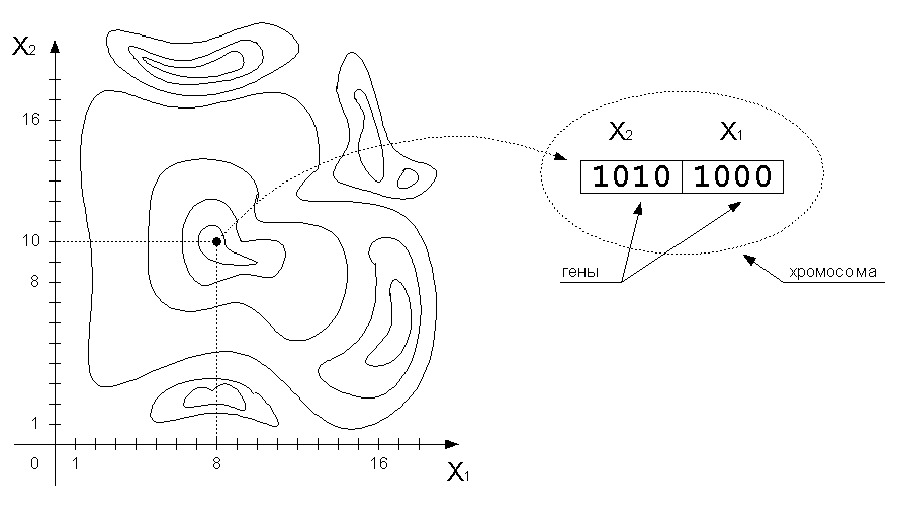


рис.1 структура хромосомы

Обычно хромосомы представляются в двоичном целочисленном виде или двоичном с плавающей запятой. В некоторых случаях более эффективно будет использование кода Грея (табл. 1), в котором изменение одного бита в любой позиции приводит к изменению значения на единицу. Для некоторых задач двоичное представление естественно, для других задач может оказаться полезным отказаться от двоичного представления. В общем случае выбор способа представления параметров задачи в виде хромосомы влияет на эффективность решения.

Таблица 1. Код Грея

|  |  |  |
| --- | --- | --- |
| Целое | Двоичный код | Код Грея |
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |

Популяция – совокупность решений на конкретной итерации, количество хромосом в популяции задаётся изначально и в процессе обычно не изменяется. Для генетических алгоритмов пока не существует четких математических основ, поэтому при их реализации возможны разные варианты решения.

1. Начальная популяция — конечный набор допустимых решений задачи. Эти решения могут быть выбраны случайным образом или получены с помощью вероятностных жадных алгоритмов. Как мы увидим ниже, выбор начальной популяции не имеет значения для сходимости процесса, однако формирование "хорошей" начальной популяции (например, из множества локальных оптимумов) может заметно сократить время достижения глобального оптимума. Это отличается от стандартных методов, когда начальное состояние всегда одно и то же.
2. Каждая хромосома популяции оценивается функцией эффективности, и ей в соответствии с этой оценкой присваивается вероятность воспроизведения .
3. В соответствии с вероятностями воспроизведения генерируется новая популяция хромосом, причём с большей вероятностью воспроизводятся наиболее эффективные элементы. Популяция следующего поколения в большинстве реализаций генетических алгоритмов содержит столько же особей, сколько начальная, но в силу отбора приспособленность в ней в среднем выше. Воспроизведение осуществляется при помощи генетических операторов кроссинговера (скрещивание) и мутации.
4. Если найдено удовлетворительное решение, процесс останавливается, иначе продолжается с шага 2.

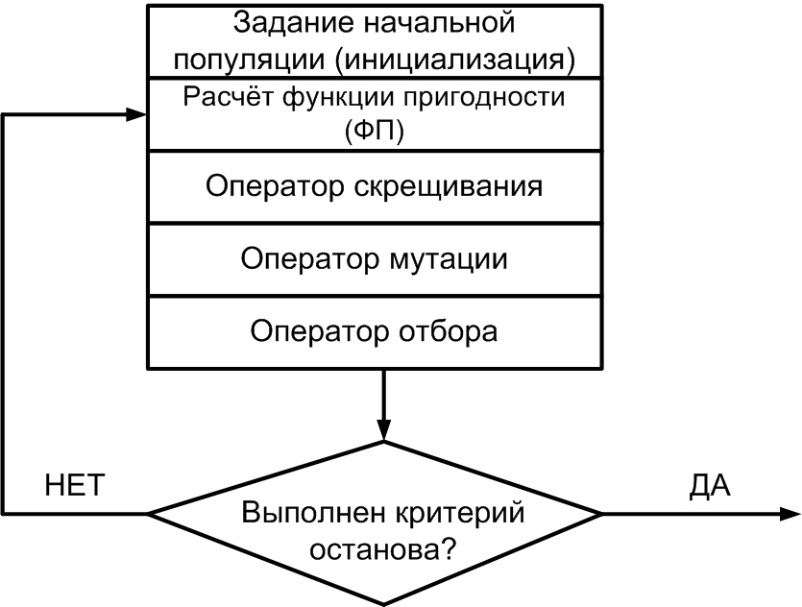


рис.2 блок-схема алгоритма.

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

Работа генетического алгоритма представляет собой итерационный процесс, который продолжается до тех пор, пока не выполнятся заданное число поколений или какой-либо иной критерий останова. На каждом поколении ГА реализуется отбор пропорционально приспособленности, кроссовер и мутация. Блок-схема генетического алгоритма достаточно проста (рис.2).

**2.3 Генетические операторы**

Стандартные операторы для генетических алгоритмов:

* селекция
* скрещивание
* мутация.

Оператор отбора [селекция]. Служит для создания промежуточной популяции . В которую копируются хромосомы из текущей популяции в соответствии с их вероятностью воспроизведения .

Существуют различные схемы отбора, самая популярная из них – пропорциональный отбор: .

Самые популярные типы отбора селекции:

* рулетка
* турнир.

Рулетка, пожалуй, самый простейший способ пропорционального отбора, отбирает особей путем «запуска» рулетки раз. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер -ого сектора пропорционален соответствующей величине , которая вычисляется по формуле . При приведенном отборе с большей вероятностью будут реже выбираться особи с низкой приспособленностью, чем особи с более высокой.

При турнирной селекции формируется случайное подмножество из элементов популяции и среди них выбирается один элемент с наибольшим значением целевой функции. Турнирный отбор реализует турниров, чтобы выбрать особей. Каждый турнир построен на выборке элементов из популяции, и выбора лучшей особи среди них. Наиболее распространен турнирный отбор с .

Турнирная селекция имеет определенные преимущества перед пропорциональной, так как не теряет своей избирательности, когда в ходе эволюции все элементы популяции становятся примерно равными по значению целевой функции. Пропорциональный отбор не гарантирует сохранности лучших результатов, достигнутых в какой-либо популяции, и для преодоления такого явления используется элитный отбор - несколько лучших индивидуумов переходят в следующее поколение без изменений, не участвуя в кроссинговерe и отборе.

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.

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.

Отбор в генетическом алгоритме и принцип естественного отбора в природе тесно связаны между собой следующим образом:

Таблица 2

|  |  |
| --- | --- |
| 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. |

Оператор кроссинговера (в некоторых источниках по генетическим алгоритмам также использыется название кроссовер или скрещивание) – операция, в ходе которой осуществляется обмен частями хромосом между двумя (может быть и больше) хромосомами в популяции. Кроссинговер смешивает "генетический материал" двух родителей, причем можно ожидать, что приспособленность родителей выше средней в предыдущем поколении, так как они только что прошли очередной раунд борьбы за выживание. Это аналогично соперничеству настоящих живых существ, где лишь сильнейшим удается передать свои (предположительно хорошие) гены следующему поколению. Важно, что кроссинговер может порождать новые хромосомы, ранее не встречавшиеся в популяции. Скрещивание может быть одноточечным или многоточечным.

Простейший одноточечный кроссинговер (рис. 3) производит обмен частями, на которые хромосома разбивается точкой кроссинговера. Одноточечный кроссовер работает следующим образом. Сперва, случайным образом выбирается одна точна разрыва. Точка разрыва – участок между соседними битами в строке. Затем, обе родительские структуры разрываются в этой точке на два куска. После, соответствующие сегменты разных родителей склеиваются и получаются два генотипа, которые могут встретиться в геноме потомков.

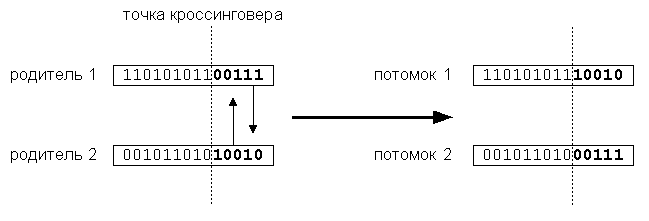


рис.3 кроссинговер

Многоточечный кроссинговер является классическим обобщением одноточечного кроссинговера. Для двухточечного потомки наследуют фрагменты хромосом родителей между двумя случайно выбранными точками скрещивания, как на рисунке 4.

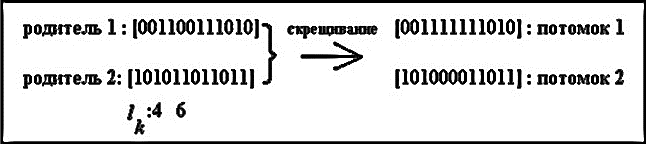


рис.4 пример двухточечного кроссинговера

Случай большего числа точек скрещивания. Например, трехточечный кроссинговер представлен на рисунке 5.

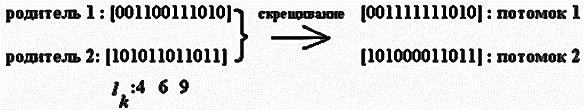


рис.5 Пример трехточечного кроссинговера

Многоточечный кроссинговер с большим четным количеством точек скрещивания графически удобно представить для хромосом в виде «колец», что показано на рис. 6. При этом хромосома рассматривается как замкнутое кольцо, а точки скрещивания выбираются с равной вероятностью по всей его окружности.

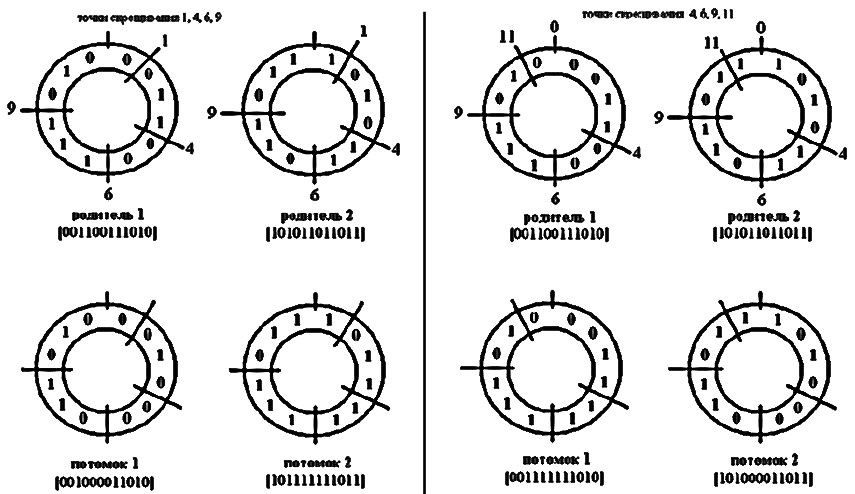


Рис. 6 четырехточечный кроссинговер

Кроссинговер с нечетным количеством точек скрещивания можно представить таким же способом, если добавить дополнительную точку скрещивания в нулевой позиции.

Все типы кроссинговера обладают общим свойством: они контролируют баланс между дальнейшим использованием уже найденных хороших подобластей пространства и исследованием новых подобластей. Достигается это за счет не разрушения общих блоков внутри хромосом-родителей, сохраняющем "хорошие" паттерны, и одновременном исследовании новых областей в результате обмена частями строк (хромосом). Совместное использование отбора и кроссинговера приводит к тому, что области пространства, обладающие лучшей средней оптимальностью, содержат больше элементов популяции, чем другие. Таким образом, эволюция популяции направляется к областям, содержащим оптимум с большей вероятностью, чем другие.

Оператор мутации, то есть случайное изменение части хромосомы. К каждому биту хромосомы с маленькой вероятностью применяется мутация – то есть бит (аллель) изменяет свое значение (рис. 6). Мутация нужна для расширения пространства исследования и предотвращения невосстановимой потери бит в аллелях. Мутации вносят новизну и предотвращают невосстановимую потерю аллелей в определенных позициях, которые не могут быть восстановлены кроссинговером, тем самым ограничивая преждевременное сжатие пространства поиска. Циклическое применение последовательности отбор-мутация направляет эволюцию элементов популяции к наиболее хорошим точкам пространства поиска.

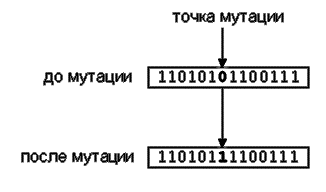


рис.6 мутация

Существует также оператор воспроизведения, называемый инверсией, который заключается в реверсировании бит между двумя случайными позициями, однако для большинства задач он не имеет практического смысла и поэтому неэффективен.

**2.4 Достоинства и недостатки стандартных и генетических методов**

Генетический алгоритм – один из новых способов решения оптимизационной задачи, но не единственно. Издавна известны два основных пути решения таких задач - переборный и локально-градиентный. У данных 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).



рис.7

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 1030 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.

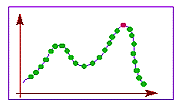


рис.8

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.

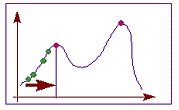


рис.9

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



рис.10

Однако, при комбинировании переборного и градиентного метода, можно надеяться, что получится 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.

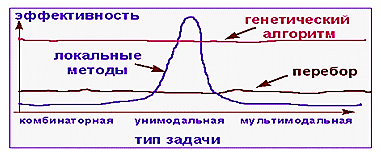


рис.11

На рисунке 11 показано, что такая комбинация позволяет обеспечить устойчиво хорошую эффективность генетического поиска для любых типов задач.

Стоит заметить, генетические алгоритмы обладают рядом преимуществ, например, такими как:

* генетические алгоритмы не требуют никакой информации о поведении функции (например, дифференцируемости и непрерывности);
* разрывы, существующие на поверхности ответа, имеют незначительный эффект на полную эффективность оптимизации;
* генетические алгоритмы относительно стойки к попаданию в локальные оптимумы;
* генетические алгоритмы пригодны для решения крупномасштабных проблем оптимизации;
* генетические алгоритмы могут быть использованы для широкого класса задач;
* генетические алгоритмы просты в реализации;
* генетические алгоритмы могут быть использованы в задачах с изменяющейся средой.

В то же время существует ряд трудностей в практическом использовании генетического алгоритма, а именно:

* с помощью генетического алгоритма проблематично найти точный глобальный оптимум;
* генетический алгоритм неэффективно применять в случае оптимизации функции, требующей большого времени на вычисление;
* генетический алгоритм непросто смоделировать для нахождения всех решений задачи;
* не для всех задач удается найти оптимально кодирование параметров;
* в многоэкстремальных задачах генетический алгоритм сталкивается с множеством аттракторов: на графике функции Растригина от одной переменной (рис. 12)

видно, что истинный минимум достигается при x = 0;

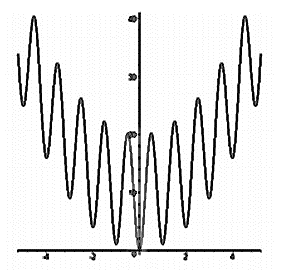


Рис. 12. Функция Растригина

* генетический алгоритм трудно применить для изолированных функций. Изолированность («поиск иголки в стоге сена») — проблема для любого метода оптимизации, поскольку функция не предоставляет никакой информации, подсказывающей, в какой области искать максимум (рис. 13). Лишь случайное попадание особи в глобальный экстремум может решить задачу (рис. 13);

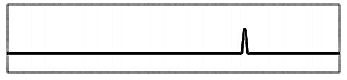


рис. 13. Изолированная функция

* дополнительный шум (nоise) разбрасывает значения приспособленности шим, поэтому часто даже хорошие шимы малого порядка не проходят отбор, что замедляет поиск решения ГА (рис. 14).

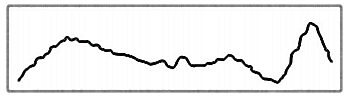


Рис. 14. Зашумленная функция

* для некоторых функций шимы малого порядка уводят популяцию к локальному оптимуму. Такую характеристику функции называют обманчивостью (deceptiоn). Например, пусть строка состоит из 10 четырехбитных подстрок. Пусть равно количеству единиц в -ой подстроке. Зададим функцию следующей таблицей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
|  | 3 | 2 | 1 | 0 | 4 |

и пусть функция приспособленности равна сумме по всем

Локальный максимум достигается при всех битах, равных 0, глобальный — при всех 1. В большинстве случаев при добавлении единицы в подстроку приспособленность особи будет падать (за исключением случая, когда все остальные биты подстроки уже равны 1). При замене 1 на 0 она будет расти. Поэтому с большой вероятностью популяция сойдется к решению, при котором большинство подстрок будут состоять из всех нулей, и лишь некоторые из всех единиц. Однако это не будет глобальным максимумом. Из этого решения попасть в глобальный максимум, то есть заменить все нули единицами для генетического алгоритма будет сложно. Рассмотренный пример схождения к ложному оптимуму демонстрирует явление эпистаза.

Итак, если на некотором множестве задана сложная функция от нескольких переменных, то генетический алгоритм - это алгоритм (или программа), который 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.

Однако, генетический алгоритм, как и другие методы эволюционных вычислений, не гарантирует обнаружения глобального решения за полиномиальное время. Генетические алгоритмы не гарантируют и того, что глобальное решение будет найдено, но они хороши для поиска "достаточно хорошего" решения задачи "достаточно быстро". Там, где задача может быть решена специальными методам, почти всегда такие методы будут эффективнее генетического алгоритма и в быстродействии, и в точность найденных решений. Главным же преимуществом генетических алгоритмов является то, что они могут применяться даже на сложных задачах, там, где не существует никаких специальных методов. Даже там, где хорошо работаю существующие методики, можно достигнуть улучшения сочетанием их с ГА.

1. **Влияние параметров генетического алгоритма на эффективность поиска**

**3.1 Операторы кроссовера и мутации**

Наиболее традиционным подходом является не традиционная схема "размножения", используемая в большинстве реализованных генетических алгоритмах 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, применяя 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 двух точечного 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.

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.

**3.2 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 ставиться 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, 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.

**3.3 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 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.

1. **Наиболее актуальные проблемы генетического алгоритма и интересные модификации**

Неудачный выбор упорядочивания и кодирования битов в хромосоме может вызвать преждевременное схождение к локальному оптимуму, ведя алгоритм по ложному пути. Для преодоления этого недостатка можно выбирать способ кодирования основываясь на дополнительной информации о задаче. Существуют так же мобильные ГА, с переменной длиной хромосом. Для некоторых задач такое представление естественно, для других надо вводить специальные правила чтения хромосом. Вместо оператора кроссинговера используются операторы разрезания строк, вероятность которого повышается с увеличением длины строки, и оператор сцепления строк, который выбирает из популяции два куска и сцепляет их в один кусок. Популяция представляет собой совокупность пере- и не доопределенных строк. Правила чтения таких строк могут быть, например, такими: в переопределённой строке при чтении слева направо не учитываются уже использованные гены, а в не доопределенных строках отсутствующие места заимствуются у лучшего элемента.

Проблема баланса исследования новых областей и приспособления к найденным. Это необходимо для предотвращения преждевременной сходимости к локальному оптимуму. Используются модифицированные способы отбора хромосом в промежуточную популяцию (отказ от вероятностных принципов в пользу элитного и отбора с вытеснением) и выбора родительской пары (случайный, селективный, на основе близкого или дальнего родства).

Интересные результаты может дать сочетание генетического алгоритма с другими методами.

И в заключение остановимся на том, когда же следует использовать генетические алгоритмы. В одних обстоятельствах они хороши, в других - не очень. В общем случае генетические алгоритмы не находят оптимального решения очень трудных задач. Если оптимальное решение задачи (например, задачи коммивояжера с очень большим числом городов) не может быть найдено традиционными способами - например, методом сцепления ветвей, - то и генетический алгоритм вряд ли найдет оптимум. С другой стороны, вполне возможно, что генетический алгоритм найдет достаточно хорошее решение. В конце концов, коммивояжеру в любом случае надо ехать продавать свои товары, даже если мы не уверены в абсолютной оптимальности маршрута! Но есть примеры, когда в очень трудных задачах, в том числе и в задачи коммивояжера, с помощью генетических алгоритмов были получены очень хорошие решения.

В двух случаях генетические алгоритмы очень хороши.

* Первый случай: когда не известен способ точного решения задачи. Если мы знаем, как оценить приспособленность хромосом, то всегда можем заставить генетический алгоритм решать эту задачу.
* Второй случай: когда способ для точного решения существует, но он очень сложен в реализации, требует больших затрат времени и денег, то есть, попросту говоря, дело того не стоит. Пример - создание программы для составления персонального расписания на основе техники покрытия множеств с использованием линейного программирования.

Несмотря на то, что конструирование хромосом и фитнес-функций может потребовать значительных усилий, генетические алгоритмы легко реализуются даже с нуля и способны решать широкий круг задач. Используя аналогию с развитием живых организмов от простых форм к более сложным, генетические алгоритмы приблизились к тому, чтобы стать общим методом решения задач. Другие "природоподобные" парадигмы, такие как моделирование отжига (simulated annealing) и табу-поиск (tabоо search), тоже позволяют решать аналогичные задачи.

1. **Пример.**

**5.1 Решение Диофантова уравнения**

Рассмотрим диофантово (только целые решения) уравнение: , где a, b, c и d - некоторые положительные целые. Применение ГА за очень короткое время находит искомое решение (a, b, c, d).

Конечно, Вы можете спросить: почему бы не использовать метод грубой силы: просто не подставить все возможные значения a, b, c, d (очевидно, ) ?

Архитектура ГА-систем позволяет найти решение быстрее за счет более 'осмысленного' перебора. Мы не перебираем все подряд, но приближаемся от случайно выбранных решений к лучшим.

Для начала выберем 5 случайных решений: . Вообще говоря, мы можем использовать меньшее ограничение для b,c,d, но для упрощения пусть будет 30.

|  |  |
| --- | --- |
| Хромосома | (a,b,c,d) |
| 1 | (1,28,15,3) |
| 2 | (14,9,2,4) |
| 3 | (13,5,7,3) |
| 4 | (23,8,16,19) |
| 5 | (9,13,5,2) |

Таблица 3: 1-е поколение хромосом и их содержимое

Чтобы вычислить коэффициенты выживаемости (fitness), подставим каждое решение в выражение a+2b+3c+4d. Расстояние от полученного значения до 30 и будет нужным значением.

Коэффициенты выживаемости первого поколения хромосом:

Так как меньшие значения ближе к 30, то они более желательны. В нашем случае большие численные значения коэффициентов выживаемости подходят, увы, меньше. Чтобы создать систему, где хромосомы с более подходящими значениями имеют большие шансы оказаться родителями, мы должны вычислить, с какой вероятностью (в %) может быть выбрана каждая. Одно решение заключается в том, чтобы взять сумму обратных значений коэффициентов, и исходя из этого вычислять проценты.

Вычислим вероятность оказаться родителем.

где 0.135266 — сумма обратных коэффициентов.

Для выбора 5-и пар родителей (каждая из которых будет иметь 1 потомка, всего - 5 новых решений), представим, что у нас есть 10000-стонняя игральная кость, на 880 сторонах отмечена хромосома 1, на 3080 - хромосома 2, на 2640 сторонах - хромосома 3, на 556 - хромосома 4 и на 2640 сторонах отмечена хромосома 5. Чтобы выбрать первую пару кидаем кость два раза и выбираем выпавшие хромосомы. Таким же образом выбирая остальных, получаем:

|  |  |
| --- | --- |
| Хромосома отца | Хромосома матери |
| 3 | 1 |
| 5 | 2 |
| 3 | 5 |
| 2 | 5 |
| 5 | 3 |

Таблица 4: Симуляция выбора родителей

Каждый потомок содержит информацию о генах и отца и от матери. Вообще говоря, это можно обеспечить различными способами, однако в нашем случае можно использовать т.н. "кроссовер" (crоss-оver). Пусть мать содержит следующий набор решений: a1,b1,c1,d1, а отец - a2,b2,c2,d2, тогда возможно 6 различных кроссоверов (| = разделительная линия):

|  |  |  |
| --- | --- | --- |
| Хромосома-отец | Хромосома-мать | Хромосома-потомок |
| a1 | b1,c1,d1 | a2 | b2,c2,d2 | a1,b2,c2,d2 оr a2,b1,c1,d1 |
| a1,b1 | c1,d1 | a2,b2 | c2,d2 | a1,b1,c2,d2 оr a2,b2,c1,d1 |
| a1,b1,c1 | d1 | a2,b2,c2 | d2 | a1,b1,c1,d2 оr a2,b2,c2,d1 |

Таблица 5: Кросс-оверы между родителями

Есть достаточно много путей передачи информации потомку, и кроссовер - только один из них. Расположение разделителя может быть абсолютно произвольным, как и то, отец или мать будут слева от черты.

А теперь попробуем проделать это с нашими потомками

|  |  |  |
| --- | --- | --- |
| Хромосома-отец | Хромосома-мать | Хромосома-потомок |
| (13 | 5,7,3) | (1 | 28,15,3) | (13,28,15,3) |
| (9,13 | 5,2) | (14,9 | 2,4) | (9,13,2,4) |
| (13,5,7 | 3) | (9,13,5 | 2) | (13,5,7,2) |
| (14 | 9,2,4) | (9 | 13,5,2) | (14,13,5,2) |
| (13,5 | 7, 3) | (9,13 | 5, 2) | (13,5,5,2) |

Таблица 6: Симуляция кроссоверов хромосом родителей

Теперь мы можем вычислить коэффициенты выживаемости потомков.

|  |  |
| --- | --- |
| Хромосома-потомок | Коэффициент выживаемости |
| (13,28,15,3) | |126-30|=96 |
| (9,13,2,4) | |57-30|=27 |
| (13,5,7,2) | |57-30|=22 |
| (14,13,5,2) | |63-30|=33 |
| (13,5,5,2) | |46-30|=16 |

Таблица 7: Коэффициенты выживаемости потомков

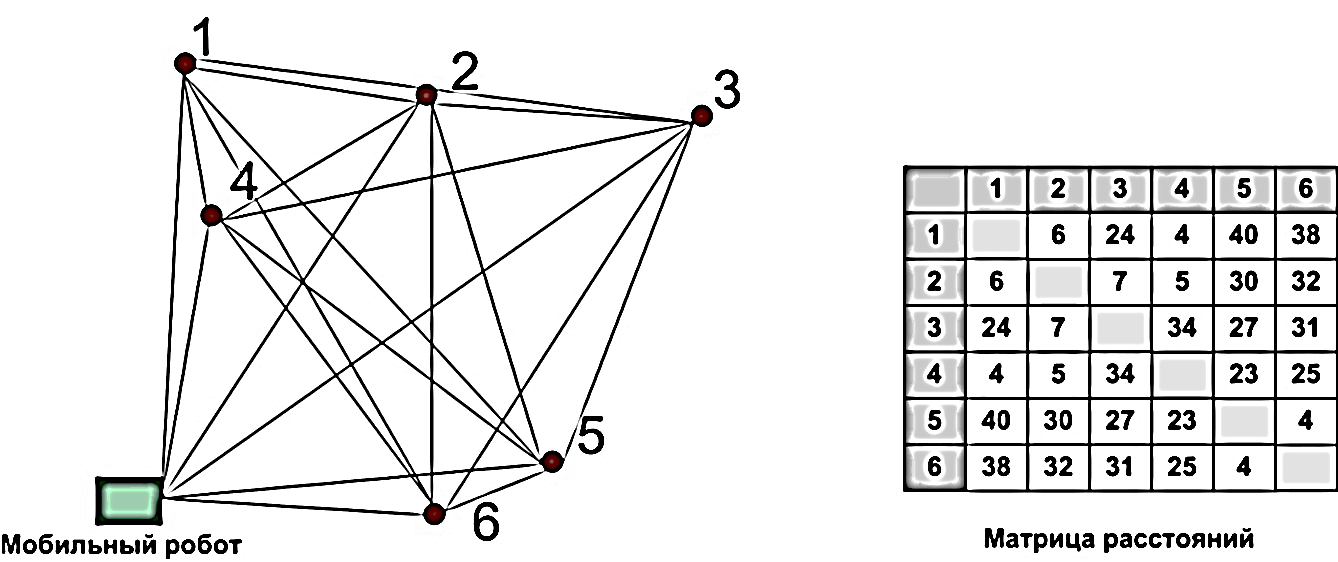
Средняя приспособленность потомков оказалась 38.8, в то время как у родителей этот коэффициент равнялся 59.4. Следующее поколение может мутировать. Например, мы можем заменить одно из значений какой-нибудь хромосомы на случайное целое от 1 до 30.

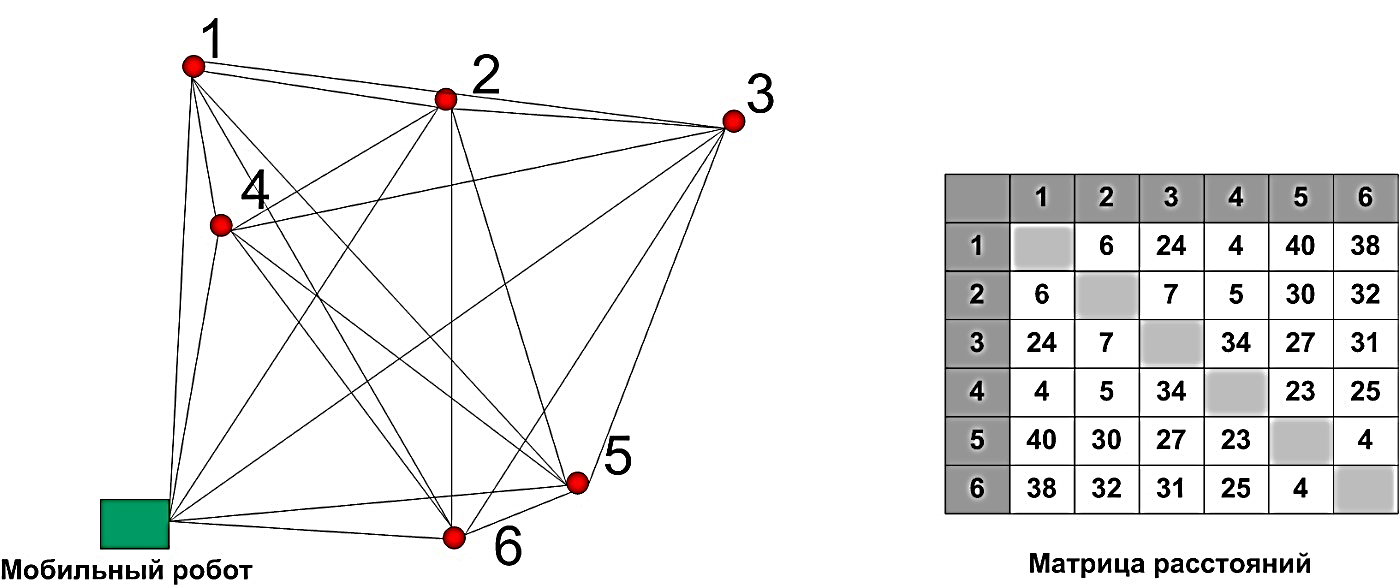
Продолжая таким образом, одна хромосома в конце концов достигнет коэффициента выживаемости 0, то есть станет решением.

Системы с большей популяцией (например, 50 вместо 5-и сходятся к желаемому уровню (0) более быстро и стабильно.

**5.2 Вариация коммивояжёра**

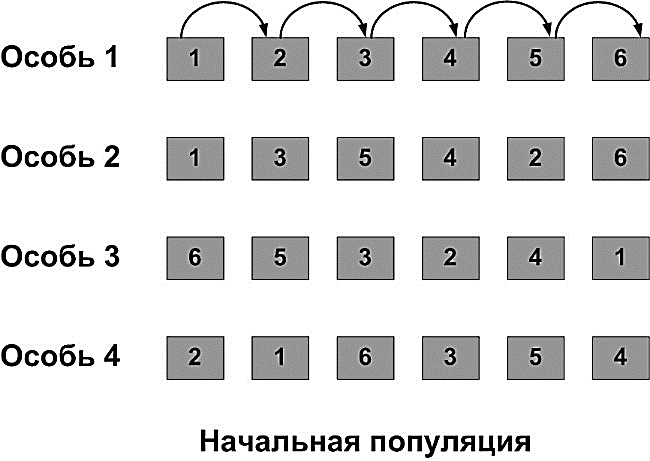
Допустим, роботу необходимо объехать шесть контрольных точек за наименьшее время. Расстояние от каждой точки до каждой задано в виде матрицы расстояний.

[](http://lazysmart.ru/wp-content/uploads/2016/01/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%801.jpg)

[](http://lazysmart.ru/wp-content/uploads/2016/01/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%801.jpg)

Это вариация задачи о коммивояжёре (путешественнике) – относится к классу NP-полных, проще говоря, не может быть решена с помощью математических формул.

Решение задачи – это последовательность прохождения контрольных точек. Возьмём несколько возможных решений (особей)– это и есть начальная популяция.

[](http://lazysmart.ru/wp-content/uploads/2016/01/%D0%9D%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D0%BF%D0%BE%D0%BF%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F.jpg)

Определения качества решений

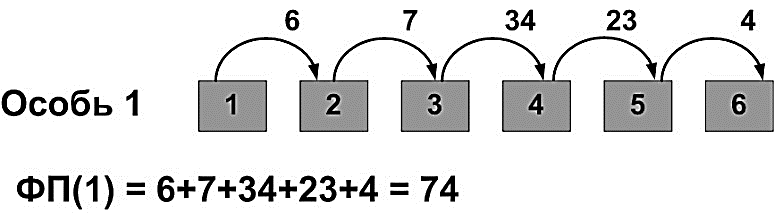
Функция пригодности – функция определяющая качество особей популяции. В нашем примере это будет сумма расстояний от точки до точки в выбранном маршруте.

ФП = Р(1)+Р(2)+Р(3)+Р(4)+Р(5)+Р(6),

где Р(1) … Р(6) – расстояние между точками в соответствующем переходе из матрицы расстояний.

Нам необходимо найти минимальное расстояние, поэтому, чем меньше значение ФП для особи, тем лучше.

Давайте посчитаем функции пригодности. Для первой особи:

[](http://lazysmart.ru/wp-content/uploads/2016/01/%D0%A4%D0%9F1.jpg)

Для остальных особей таким же образом получаем:

ФП(2) = 111

ФП(3) = 47

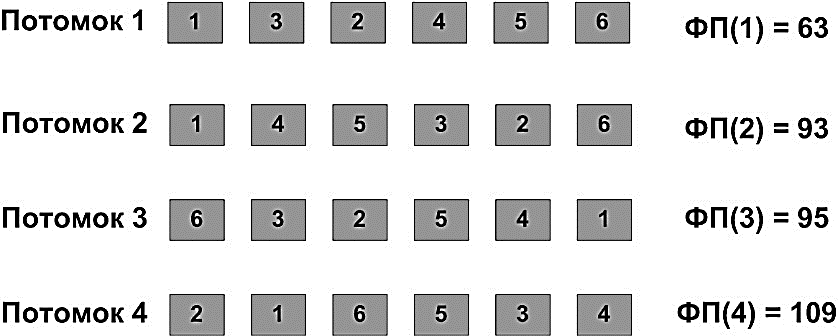
ФП(4) = 125

Тут всё очевидно: особь №3 – лучшая, а №4 – самая плохая.

Генетические операторы

Дальше согласно алгоритму, необходимо слегка изменить исходных особей, так чтобы они были похожи на своих родителей, но немного отличались. Так реализуется биологическое понятие «изменчивость».

После применения генетических операторов (селекции, мутации) мы получим еще несколько особей – потомков. Допустим таких:

[](http://lazysmart.ru/wp-content/uploads/2016/01/%D0%BF%D0%BE%D1%82%D0%BE%D0%BC%D0%BA%D0%B8.jpg)

Для потомков тоже посчитаны функции пригодности.

Оператор селекции

Настало время искусственного отбора. На этом шаге алгоритм выберет лучших особей и отбросит наименее приспособленных, подобно тому, как делает селекционер, создавая новый вид растений.

Алгоритмы селекции тоже могут быть различны, не будем пока заострять на этом внимание. Просто возьмем и отбросим из первой популяции (родители + потомки) четыре худших особи. Останутся родитель 1 и 3, и потомки 1 и 2. Эти особи сформируют новую популяцию. Далее алгоритм будет повторяться.

Критерий останова генетического алгоритма

Вспомним, что мы искали кратчайший путь прохождения робота через все контрольные точки. Абсолютно правильный ответ будет получен, только если перебрать все варианты, а их очень много даже для шести точек (а если точек будет больше?). Поэтому генетический алгоритм ищет не правильное решение, а оптимальное, исходя из условий, которые задаёт пользователь.

Критерий останова – условие, по которому генетический алгоритм останавливает свою работу.

С генетическим алгоритмом мы ищем не самое лучшее решение, а то решение, которое нас устроит.

В нашем случае мы можем указать, например, один из следующих критериев останова:

* Суммарный путь меньше 50
* Время работы алгоритма 1 час
* Число циклов алгоритма 10
* В течение 3 поколений не появляются особи лучше тех, которые были
* и т.д.
  1. **Простой пример генетического алгоритма**

Рассмотрим принципы работы генетических алгоритмов на максимально простом примере.

Пусть требуется найти глобальный минимум функции

на отрезке [0; 7] (рис. 1). На этом отрезке функция принимает минимальное значение в точке . Очевидно, что в точке функция попадает в локальный минимум. Если для нахождения глобального минимума использовать градиентные методы, то в зависимости от начального приближения можно попасть в данный локальный минимум.

Рассмотрим на примере данной задачи принцип работы генетических алгоритмов. Для простоты положим, что x принимает лишь целые значения, т.е. . Это предположение существенно упростит изложение, сохранив все основные особенности работы генетического алгоритма.

Выберем случайным образом несколько чисел на отрезке . Будем рассматривать эти числа в качестве пробных решений нашей задачи.

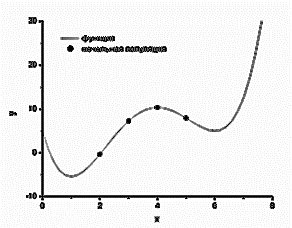


График целевой функции с выбранными значениями пробных решений

Основной идеей генетических алгоритмов является организация «борьбы за существование» и «естественного отбора» среди этих пробных решений. Запишем пробные решения в двоичной форме: . Как известно, принцип естественного отбора заключается в том, что в конкурентной борьбе выживает наиболее приспособленный. В нашем случае приспособленность особи определяется целевой функцией: чем меньше значение целевой функции, тем более приспособленной является особь, т.е. пробное решение, использовавшееся в качестве аргумента целевой функции (см. табл. 8).

Таблица 8

Исходная популяция

|  |  |  |  |
| --- | --- | --- | --- |
| № | Особь | | |
| Целое число | Двоичное число | Приспособленность |
| 1 | 2 | 010 | -0,33 |
| 2 | 3 | 011 | 7,25 |
| 3 | 5 | 101 | 7,92 |
| 4 | 4 | 100 | 10,33 |

Теперь приступим к процессу размножения: попробуем на основе исходной популяции создать новую, так чтобы пробные решения в новой популяции были бы ближе к искомому глобальному минимуму целевой функции. Для этого сформируем из исходной популяции брачные пары для скрещивания. Поставим в соответствие каждой особи исходной популяции случайное целое число из диапазона от 1 до 4. Будем рассматривать эти числа как номера членов популяции. При таком выборе какие-то из членов популяции не будут участвовать в процессе размножения, так как образуют пару сами с собой. Какие-то члены популяции примут участие в процессе размножения неоднократно с различными особями популяции. Процесс размножения (рекомбинация) заключается в обмене участками хромосом между родителями. Например, пусть скрещиваются две хромосомы и . Определяем случайным образом точку разрыва хромосомы, пусть, это будет 3: . Теперь хромосомы обмениваются частями,

Для нашей популяции процесс создания первого поколения потомков показан в таблице 9.

Таблица 9

Одноточечный кроссинговер

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Особь популяции | Выбранный номер | Вторая  особь-родитель | Точка кроссинговера | Особи-потомки |
| 1 | 010 | 1 | 010 | 1 | 000 |
| 2 | 011 | 4 | 100 | 1 | 110 |
| 3 | 101 | 3 | 101 | 2 | 100 |
| 4 | 100 | 1 | 010 | 2 | 011 |

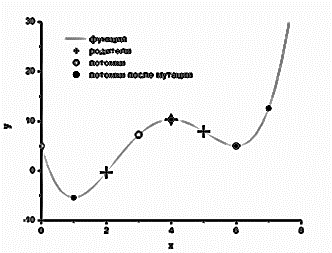
Следующим шагом в работе генетического алгоритма являются мутации, т.е. случайные изменения полученных в результате скрещивания хромосом. Пусть вероятность мутации равна 0,3. Для каждого потомка возьмем случайное число на отрезке , и если это число меньше 0,3, то инвертируем случайно выбранный ген (заменим 0 на 1 или наоборот) (см. табл. 10).

Таблица 10

Мутация потомков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  № | Особи-потомки | Случайное число | Выбранный ген  для мутации | Потомок после  мутации | Приспособленность  потомка до  мутации | Приспособленность  потомка после  мутации |
| 1  1 | 000  000 | 0,1  0,1 | 3  3 | 001  001 | 5  5 | -5,42 |
| 2 | 110 | 0,6 | - | 110 | 5 | 5 |
| 3 | 100 | 0,5 | - | 100 | 10,33 | 10,33 |
| 4 | 011 | 0,2 | 1 | 111 | 7,25 | 12,58 |

Как видно на примере, мутации способны улучшить (первый потомок) или ухудшить (четвертый потомок) приспособленность особи-потомка. В результате скрещивания хромосомы обмениваются «хвостами», т.е. младшими разрядами в двоичном представлении числа. В результате мутаций изменению может подвергнуться любой разряд, в том числе, старший. Таким образом, если скрещивание приводит к относительно небольшим изменениям пробных решений, то мутации могут привести к существенным изменениям значений пробных решений (см. рис. Изменение популяции в процессе естественного отбора).



Изменение популяции в процессе естественного отбора

Теперь из четырех особей-родителей и четырех полученных особей потомков необходимо сформировать новую популяцию. В новую популяцию отберем четыре наиболее приспособленных особей из числа «старых» особей и особей-потомков (см. табл. 11).

В результате получим новое поколение, которое представлено на рис. Минимизируемая функция и особи новой популяции.

Получившуюся популяцию можно будет вновь подвергнуть кроссинговеру, мутации и отбору особей в новое поколение. Таким образом, через несколько поколений мы получим популяцию из похожих и наиболее приспособленных особей. Значение приспособленности наиболее «хорошей» особи (или средняя приспособленность по популяции) и будет являться решением нашей задачи. Следуя этому, в данном случае, взяв наиболее приспособленную особь 001 во втором поколении, можно сказать, что минимумом целевой функции является значение , соответствующее аргументу . Тем самым попадания в локальный минимум удалось избежать! На данном примере разобран вариант простого генетического алгоритма. При дальнейшем использования ГА к разным задачам возможно моделирование основных операторов алгоритма.

Таблица 11

Формирование новой популяции

из особей-родителей и особей-потомков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Особь | Приспособленность | Новая популяция | Приспособленность особей в новой популяции |
| 1 | 010 | -0,33 | 001 | -5,42 |
| 2 | 011 | 7,25 | 010 | -0,33 |
| 3 | 101 | 7,92 | 110 | 5 |
| 4 | 100 | 11,33 | 011 | 7,25 |
| 5 | 001 | -5,42 |  |  |
| 6 | 110 | 5 |
| 7 | 100 | 10,33 |
| 8 | 111 | 12,58 |

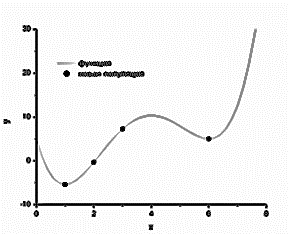


Рис. 3. Минимизируемая функция и особи новой популяции

**Заключение**

Изложенный подход является эвристическим, т. е. показывает хорошие результаты на практике, но плохо поддается теоретическому исследованию и обоснованию. Естественно задать вопрос — следует ли пользоваться такими алгоритмами, не имеющими строгого математического обоснования? Как и в вопросе о нейронных сетях, здесь нельзя ответить однозначно. С одной стороны, в математике существует достаточно большой класс абсолютно надежных (в смысле гарантии получения точного решения) методов решения различных задач. С другой стороны, речь идет о действительно сложных практических задачах, в которых эти надежные методы часто неприменимы. Нередко эти задачи выглядят настолько необозримыми, что не предпринимается даже попыток их осмысленного решения. Например, фирма, занимающаяся транспортными перевозками, в современных условиях российского бизнеса скорее предпочтет нанять лишних водителей и повысить цены на свои услуги, чем оптимизировать маршруты и расписания поездок. На западном рынке, где уже давно действуют законы более или менее честной конкуренции, оптимальность деятельности компании значительно влияет на ее доходы и даже может стать решающим фактором для ее выживания. Поэтому любые идеи, позволяющие компании стать «умнее» своих конкурентов, находят там широкое применение. Генетические алгоритмы — реализация одной из наиболее популярных идей такого рода. Эта популярность вызвана, по-видимому, исключительной красотой подхода и его близостью к природному механизму. Подобным образом популярность нейросетевой технологии подогревается во многом ее сходством с работой мозга. По-настоящему активное развитие эвристических подходов, как мы видим, непосредственно связано с развитием свободного рынка и экономики в целом.

**Приложение 1**

**Теория Дарвина**

Теория Дарвина биологической эволюции была изложена в 1859 г. в работе «Происхождение видов». Она рассматривает изменение популяции в определенной статичной или динамичной внешней среде. Под популяцией понимается большое количество особей одного вида, способных давать плодовитое потомство. Все члены популяции характеризуются индивидуальными внешними параметрами (фенотипом). Некоторые из параметров оказываются полезными для выживания и размножения, другие скорее вредят. Все внешние данные особи кодируются ее цепью ДНК (генотипом). Отдельные участки этой цепи (гены) определяют различные параметры особи.

В основе модели эволюции Дарвина лежат случайные изменения отдельных материальных элементов живого организма при переходе от поколения к поколению. Целесообразные изменения, которые облегчают выживание и производство потомков в данной конкретной внешней среде, сохраняются и передаются потомству (т.е. наследуются). Особи, не имеющие соответствующих характеристик, погибают, не оставив потомства или оставив его меньше, чем приспособленные (считается, что количество потомства пропорционально степени приспособленности).

Естественный отбор происходит в условиях конкуренции особей популяции, а иногда и различных видов, друг с другом за различные ресурсы, такие, например, как пища или вода. Кроме того, члены популяции одного вида часто конкурируют в привлечении брачного партнера. Те особи, которые наиболее приспособлены к окружающим условиям, проживут дольше и создадут более многочисленное потомство, чем их собратья. Это означает, что гены от высоко адаптированных или приспособленных особей будут распространяться в увеличивающемся количестве потомков на каждом последующем поколении. Некоторые потомки совместят в себе части цепи ДНК, отвечающие за наиболее удачные качества родителей, и, таким образом, окажутся еще более приспособленными. В итоге генотип слабо приспособленных особей с большой вероятностью исчезнет из генофонда популяции.

Изредка происходит мутация: некоторый случайный нуклеотид цепи ДНК особи может измениться на другой. Если полученная цепь будет использоваться для создания потомства, то возможно появление у потомков совершенно новых качеств. Поэтому в результате естественного отбора возникает популяция из наиболее приспособленных особей, которая может стать основой нового вида.

Естественный отбор, скрещивание (кроссинговер) и мутация обеспечивают развитие популяции. Каждое новое поколение в среднем более приспособлено, чем предыдущее, т.е. оно лучше удовлетворяет требованиям внешней среды. Такой процесс называется эволюцией.

Генетические алгоритмы моделируют процесс эволюции в природе. Тем самым для решения задачи оптимизации многопараметрической функции и применяется теория Дарвина.

**Приложение 2**

**Некоторые понятия из теории оптимизации**

Теория оптимизация занимается исследованием методов решения оптимизационных задач.

Оптимизационные задачи — это задачи, в которых требуется найти

наилучшее решение; при этом, как правило, существуют различные ограничения на область изменения управляющих переменных, что не позволяет воспользоваться методами классического математического анализа, а требует применения разнообразных вычислительных методов.

Методы оптимизации — поиск экстремума функций (в практических задачах — критерий оптимальности) при наличии ограничений или без ограничений. Это прежде всего оптимальное проектирование (выбор наилучших номинальных технологических режимов, элементов конструкций, структуры технологических цепочек, условий экономической деятельности, повышение доходности и т.д.), оптимальное управление, построение нелинейных математических моделей объектов управления и т.д.

При рассмотрении любой оптимизационной задачи определяют:

1. целевую (объектную) функцию;
2. управляющие параметры.

Дадим определения этим понятиям.

Целевой функцией называется функция вида при наложенных условиях — ограничений.

Управляющие параметры — параметры, изменение которых влечет изменение значения целевой функции.

Если целевая функция зависит только от одного управляющего пара-

метра, то такая оптимизационная задача называется одномерной, иначе — многомерной.

Ограничения в оптимизационной задаче могут быть:

* общими, если они накладываются на множество допустимых значений управляющих параметров, например,
* специальными, если они накладываются на значения какой-либо функции, зависящей от значений управляющих параметров, например,
* типа неравенств (собственно ограничения), например, fi(x) > 0,  
   где и — целевая функция;
* типа равенств (связи), например, ,   
  где и f — целевая функция;
* активными, например, xi min 6 xi 6 xi max. Активными называются такие ограничения, на границе которых находится решение (например, ограничения типа равенств);
* автономными.

Функции, задающие ограничения, могут формировать допустимую область (допустимое множество) с различными свойствами: монотонными, колебательными, с большей и малой кривизной и т.д.

Решение допустимой оптимизационной задачи состоит в выборе оптимального решения из допустимого множества.

Точка , удовлетворяющая ограничениям, называется допустимым решением оптимизационной задачи (в дальнейшем ОЗ).

Множество всех допустимых решений ОЗ называется допустимым множеством этой задачи.

Если ОЗ имеет хотя бы одно допустимое решение (т.е. ее допустимое множество не пусто), она называется допустимой, если ОЗ не имеет ни одного допустимого решения, то она называется недопустимой.

Точка называется оптимальным решением ОЗ, если:

* она является допустимым решением этой ОЗ;
* на этой точке целевая функция достигает глобального максимума (или минимума) среди всех точек, удовлетворяющих ограничениям.

При моделировании оптимизационной задачи можно пользоваться следующими условиями окончания поиска оптимального решения:

* малость изменения критерия оптимальности за 1 цикл;
* невозможность улучшения критерия оптимальности ни по одной переменной;

Для одномерной оптимизации: величина отрезка, содержащего оптимум, меньше заданной погрешности.