Лабораторная работа 3.

Использование генетических алгоритмов для решения задачи дискретной оптимизации.

1. **Теоретическое введение.**

Генетические алгоритмы

Естественный отбор в природе

Согласно эволюционной теории каждый биологический вид целенаправленно развивается и изменяется для того, чтобы наилучшим образом приспособиться к окружающей среде. Эволюция в этом смысле представляет процесс оптимизации всех живых организмов. Природа решает эту задачу оптимизации путем естественного отбора. 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 особей 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.

Дадим краткую справку о том, как устроены механизмы генетического наследования. В каждой клетке любого животного содержится вся генетическая информация данной особи. Эта информация записана в виде набора молекул ДНК, каждая из которых представляет собой цепочку, состоящую из молекул нуклеотидов четырех типов, обозначаемых А, T, C и G. Собственно информацию несет порядок следования нуклеотидов в ДНК. Таким образом, генетический код особи – это длинная строка, где используются всего 4 символа. 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 особей 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, 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.

Что такое генетический алгоритм

Пусть дана некоторая сложная целеваяфункция, зависящая от нескольких переменных, и требуется решить задачу оптимизации, т. е.найти такие значения переменных, при которых значение функции максимально или минимально.

Эту задачу можно решить, применяя известные биологические эволюционные подходы к оптимизации. Будем рассматривать каждый вариант (набор значений переменных) как особь, а значение целевой функции для этого варианта – как приспособленность данной особи. Тогда в процессе эволюции приспособленность особей будет возрастать, а значит, будут появляться все более и более оптимальные варианты. Остановив эволюцию в некоторый момент и выбрав лучший вариант, можно получить достаточно хорошее решение задачи.

*Генетический алгоритм (ГА)* – это последовательность управляющих действий и операций, моделирующая эволюционные процессы на основе аналогов механизмов генетического наследования и естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде.

*Хромосома* – вектор (последовательность) из нулей и единиц, каждая позиция (бит) которого называется *геном*.

*Особь (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.

Генетические алгоритмы представляют собой скорее подход, чем единые алгоритмы. Они требуют содержательного наполнения для решения каждой конкретной задачи.

На рис. 3.20 показан один из вариантов структуры генетического алгоритма. Вначале генерируется случайная популяция – несколько особей со случайным набором хромосом (числовых векторов). Генетический алгоритм имитирует эволюцию этой популяции как циклический процесс скрещивания особей, мутации и смены поколений (отбора).

В течение жизненного цикла популяции, т. е. в результате нескольких случайных скрещиваний (посредством кроссовера) и мутаций, к ней добавляется какое-то количество новых вариантов. Далее происходит отбор, в результате которого из старой популяции формируется новая, после чего старая популяция погибает. После отбора к новой популяции опять применяются операции кроссовера и мутации, затем опять происходит отбор, и так далее.



Рис. Вариант структуры генетического алгоритма

Отбор в генетическом алгоритме тесно связан с принципами естественного отбора следующим образом:

* приспособленность особи соответствует значению целевой функции на заданном варианте;
* выживание наиболее приспособленных особей соответствует тому, что популяция следующего поколения вариантов формируется с учетом целевой функции. Чем приспособленнее особь, тем больше вероятность ее участия в кроссовере, т. е. в размножении.

Таким образом, модель отбора определяет, как следует строить популяцию следующего поколения. Как правило, вероятность участия особи в скрещивании берется пропорциональной ее приспособленности. Часто используется так называемая стратегия элитизма, при которой несколько лучших особей 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 особей перестает заметно увеличиваться, процесс останавливают и в качестве решения задачи оптимизации берут наилучший из найденных вариантов.

Математически, изложенное можно представить следующим образом. Пусть имеется некоторая целевая функция от многих переменных, у которой необходимо найти глобальный максимум или минимум:

*f*(*x*1, *x*2, *x*3, …, *xn*).

Представим независимые переменные в виде хромосом. Для этого выполним кодирование независимых переменных либо в двоичном формате, либо в формате с плавающей запятой.

В случае двоичного кодирования используется *n* бит для каждого параметра, причем, *n* может быть различным. Если параметр может изменяться между минимальным (min) и максимальным (max) значениями, можно использовать следующие формулы для преобразования:

, ,

где *g* – значение параметра в двоичном формате, *r* – значение параметра в формате с плавающей запятой.

Хромосомы в формате с плавающей запятой задаются путем последовательного размещения закодированных параметров друг за другом.

Наиболее хорошие результаты дает вариант представления хромосом в двоичном формате (особенно, при использовании кодов Грея). Однако в этом случае необходимо постоянно осуществлять кодирование/декодирование параметров (генов).

Рассмотрим работу генетического алгоритма более подробно. Заранее подбирается некоторое представление для рассматриваемого решения, размер и структура популяции. В первом поколении случайным образом генерируется популяция хромосом. Обычно, размер популяции постоянен. Определяется «полезность» хромосом. После чего генетический алгоритм может начинать генерировать новую популяцию.

Далее осуществляется репродуцирование, состоящее из:

* селекции;
* и трех генетических операторов: кроссовера, мутации и инверсии, порядок применения которых не важен.

Из трех генетических операторов кроссовер является наиболее важным. Он генерирует новую хромосому потомка, объединяя генетический материал двух родителей. Существует несколько вариантов кроссовера. Наиболее простым является одноточечный, в котором берутся две хромосомы и «перерезаются» в случайно выбранной точке. Хромосома потомка получается из начала одной и конца другой родительских хромосом:

001100101110010|11000 → 001100101110010***11100***

***110101101101000|11100***

Мутация представляет собой случайное изменение хромосомы (обычно простым изменением состояния одного из битов на противоположное). Данный оператор позволяет, во-первых, более быстро находить локальные экстремумы и, во-вторых, перескочить в другой локальный экстремум:

00110010111001***0***11000 → 00110010111001***1***11000

Инверсия изменяет порядок бит в хромосоме путем циклической перестановки (случайное количество раз). Многие модификации ГА обходятся без данного генетического оператора:

001100101110010***11000*** → ***11000***001100101110010

По скорости определения оптимума целевой функции генетические алгоритмы на несколько порядков превосходят случайный поиск. Причина этому заключается в том, что большинство систем имеют довольно независимые подсистемы. Вследствие чего при обмене генетическим материалом от каждого из родителей берутся гены, соответствующие наиболее удачному варианту определенной подсистемы (неудачные варианты постепенно погибают). Генетический алгоритм позволяет накапливать удачные решения для таких систем в целом.

Генетические алгоритмы менее применимы для систем, которые сложно разбить на подсистемы. Кроме того, они могут давать сбои из-за неудачного порядка расположения генов (например, если рядом расположены параметры, относящиеся к различным подсистемам), при котором преимущества обмена генетическим материалом сводятся к нулю. Это замечание несколько сглаживается в системах с диплоидным (двойным) генетическим набором.

Данные, которые закодированы в генотипе, могут представлять собой команды какой-либо виртуальной машины. В таком случае можно говорить об эволюционном или генетическом программировании. В простейшем случае, можно ничего не менять в генетическом алгоритме. Однако в таком случае, длина получаемой последовательности действий (программы) получается не отличающейся от той (или тех), которая является «затравкой» на этапе инициализации. Современные алгоритмы генетического программирования функционируют в системах с переменной длиной генотипа.

**2. Варианты заданий.**

С помощью генетического алгоритма решить задачу дискретной оптимизации.

Перечень заданий:

1. Задача об упаковке в контейнеры.
2. Нахождение максимальной клики в графе.
3. Нахождение максимального внутренне устойчивого множества в графе.
4. Решить задачу коммивояжера в полном взвешенном графе.
5. Нахождение минимального внешне устойчивого множества в графе.
6. Нахождение максимального паросочетания в двудольном графе.
7. Решение задачи о выполнимости КНФ.
8. Решение задачи о рюкзаке.
9. Решение задачи о назначениях.
10. Нахождение гамильтонова обхода в неориентированном графе.
11. Нахождение гамильтонова обхода в ориентированном графе.
12. Решение задачи о минимальном покрытие множествами [1].
13. Решение задачи о минимальном множестве ребер, разрезающих циклы [1].
14. Решение задачи о ранце [1].
15. Решение задачи о нахождении правильной раскраски наименьшем числом красок [3].
16. Решение задачи о нахождении наибольшей клики в графе [2].

**3. Требования к оформлению отчета.**

Отчет по лабораторной работе должен содержать:

* постановку задачи,
* подробное описание генетического алгоритма решения задачи,
* описание (блок-схему) алгоритма,
* текст программы,
* распечатки результатов выполнения программы для разных начальных данных.

**4. Контрольные вопросы.**

* 1. Приведите приближенный алгоритм решения поставленной задачи.
  2. Объясните, почему для решения поставленной задачи используют приближенные алгоритмы.
  3. Перечислите основные этапы реализации генетических алгоритмов.
  4. Когда генетический алгоритм прекращает порождение новых популяций.
  5. Приведите примеры возможной реализации процедуры естественного отбора в генетических алгоритмах.

**Литература**

* + 1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979.
    2. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982.
    3. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Телеком, 2001.
    4. Фролов А.Б., Андреев А.В., Болотов А.А., Коляда К.В. Прикладные задачи дискретной математики и сложность алгоритмов. – М.: Из-во МЭИ, 1997.