

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ЭТАПА МУТАЦИИ

Носкова Е. Е., Дружинская Е. В.

Уфимский государственный нефтяной технический университет
Уфа, Россия

E-mail: noskova1610@gmail.com, alena1806@mail.ru

В статье представлен краткий обзор генетического алгоритма, и программная реализация его этапа мутации. Статья посвящена комплексному исследованию этапа мутации генетических алгоритмов. Произведен сбор и анализ научной литературы, проведены систематизация и структурирование сведений о генетическом алгоритме, особое внимание было уделено этапу мутации: выделены и описаны основные алгоритмы изменения последовательности на данном этапе и критерии окончания цикла, представлен словесно-описательный алгоритм и результаты его программной реализации. Кратко даны определения понятий, используемых в генетических алгоритмах, рассмотрены операторы работы с данными алгоритмами, построена обобщённая модель генетического алгоритма в виде блок-схемы. Реализация позволяет продемонстрировать использование этапа мутации в генетическом алгоритме.

ВВЕДЕНИЕ

Естественный отбор является основным фактором эволюции. Суть естественного отбора заключается в том, что в результате действий в популяции увеличивается число наиболее приспособленных особей, так как они имеют больше шансов на выживание и размножение. Потомки получают лучшие качества от родителей вследствие генетического наследования. Но в природе не всегда побеждает сильнейший из-за нерегулируемых случайностей. В отличие от природных явлений, программно реализованные генетические алгоритмы предоставляют возможность принудительного выбора и сохранения лучших экземпляров набора. Основная идея генетических алгоритмов заключается в том, что с использованием принципов «естественного отбора» среди пробных решений выбирается наиболее качественное. Генетические алгоритмы дают возможность оптимизировать функции дискретных переменных, непрерывные функции, компьютерные программы, а также, принимать решение о стратегии выполнения активности для достижения поставленной цели и сокращать перебор в сложных задачах. Алгоритм позволяет сократить число переборов при нахождении значений функции, описывающей сложные задачи. А также, позволяет в условиях неопределённости выполнять подбор наиболее правильного решения с максимальным соблюдением всех ограничений, накладываемых на результат.

Генетические алгоритмы применяются для настройки и обучения искусственной нейронной сети, задач компоновки, составления расписаний и игровых стратегиях, а также, робототехнике и создание моделей искусственной жизни.

I. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Алгоритм работает на сгенерированной начальной популяции – некотором массиве случайных объектов, возможно, но не обязательно при-

ближенных к решению. Выборка производится из представленного набора данных, и все операции применяются к выборке до достижения поставленных ограничений [1].

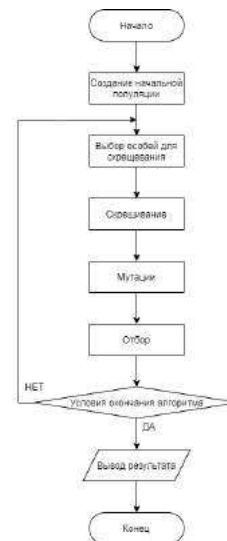


Рис. 1 – Графическая модель алгоритма

II. ОПЕРАТОРЫ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Реализация и эффективность генетических алгоритмов зависит от выбора операторов. Основными операторами генетического алгоритма являются мутация, отбор (селекция), скрещивание и выбор родителей [4].

Рассмотрены наиболее распространённые операторы выбора пары, которые станут родителями:

Панимиксия реализует случайное бинарное отношение. При этом некоторые элементы могут образовать несколько пар, а какие-то не участвовать в данном отношении. Поскольку с каждым применением оператора популяция увеличивается, то эффективность алгоритма падает за счет зашумленности данных.

Инбридинг – метод, при котором оба родителя являются членами одной популяции, и первый родитель выбирается случайным образом, а второй родитель является ближайшим «родственником» к первому [3].

При аутбридинге используется понятие схожести особей, в котором брачные пары формируются из максимально далеких по родству особей [1].

Оператором скрещивания выступает рекомбинация, которая применяется сразу же после отбора предков для получения нового поколения. Смысл воспроизведения заключается в том, что потомки наследуют генную информацию от обоих родителей.

III. МУТАЦИИ

Нахождение решений внутри замкнутой системы приводит к ее энтропии.

Этап мутации необходим для вывода популяции из локального экстремума. Изменение случайно выбранного элемента в объекте препятствует преждевременной сходимости популяции к неправильному решению. Следующие операторы мутации изменяют состав объекта:

1. присоединение случайного к концу последовательности;
2. вставка случайного гена в случайно выбранную позицию в последовательности;
3. удаление случайно выбранного гена из последовательности;
4. обмен местами в последовательности двух соседей одного случайно выбранного гена.

В программной реализации случайным образом генерируется последовательность, после чего, происходит изменение последовательности в соответствии с выбранным оператором мутации. Реализованы и запущены все 4 оператора. На вход поступает последовательность генов, после чего происходит изменение в составе объекта, которые и обозначает мутацию. На выходе имеем последовательность с тем же количеством генов, измененную по некоторому правилу, изменяющим популяцию.

Последовательность задается массивом чисел, который генерируется случайным образом используя функцию `rand()` и аргумент `Time (NULL)`, позволяющий генерировать неповторяющуюся последовательность генов, используя текущее время. После этого данные массива отправляем в функцию, в которой используя случайные числа генерируется элементы для изменения структуры объекта.

```

0 1 1 1 1 1 0
Присоединение к концу последовательности
1 1 1 1 1 1 0
Вставка случайного гена из совокупности всевозможных значений генов в случайно выбранную позицию
0
0 1 1 1 1 1 0
Удаление случайно выбранного гена из последовательности
2
0 1 1 1 1 1 0
Обмен местами в последовательности двух соседей одного случайно выбранного гена
2
0 1 1 1 1 1 0

```

Рис. 2 – Программная реализация этапа мутации после первой итерации

```

1 0 1 1 0 1 1
Присоединение к концу последовательности
0 1 1 0 1 1 1
Вставка случайного гена из совокупности всевозможных значений генов в случайно выбранную позицию
5
0 0 1 1 1 1 1
Удаление случайно выбранного гена из последовательности
2
1 1 1 0 1 1 1
Обмен местами в последовательности двух соседей одного случайно выбранного гена
2
0 1 1 1 0 1 1

```

Рис. 3 – Программная реализация этапа мутации после второй итерации

Для создания маркированного списка нужно использовать команду `\List`, в которой каждый пункт списка обозначается тегом `\item`.

IV. ВЫБОРКА НОВОГО ПОКОЛЕНИЯ

На этапе построения выборки из нового поколения применимы несколько типов операторов. Отбор усечением использует целевую функцию, проверяющую каждую особь на пригодность к решению. Ранжирование популяции производится по убыванию значения целевой функции. Этот метод не имеет аналогов в естественной эволюции и обычно используется для больших популяций [2]. В элитарном отборе участвуют как родители, так и потомки, которые создают промежуточную популяцию. А затем, члены этой популяции оцениваются и объекты с высшей оценкой образуют следующее поколение. В отборе вытеснением выбор особи в новое поколение зависит от величины пригодности и хромосомного набора. Из всех особей с одинаковой пригодностью и приспособленностью выбирают особи с разными генотипами, таким образом поддерживается структурное разнообразие объектов.

Вновь построенная выборка проверяется на соответствие ограничивающих условий для решения. При соответствии алгоритм возвращает результат – полученную выборку. Иначе выборка рассматривается как начальная, и алгоритм запускает новую итерацию [1].

Применение генетического алгоритма позволяет в условиях неопределённости выполнять подбор наиболее правильного решения с максимальным соблюдением всех ограничений, накладываемых на результат.

V. СПИСОК ЛИТЕРАТУРЫ

1. Панченко, Т. В. Генетические алгоритмы / Ю. Ю. Тарасевич // Астрахань: Издательский дом «Астраханский университет» – 2007. – С. 57-65.
2. Модификации генетических алгоритмов [Электронный ресурс] / . – Режим доступа: <https://intuit.ru/studies/courses/14227/1284/lecture/24172?page=2> – Дата доступа: 25.09.2021.
3. Гладков, Л. А., Курейчик, В. В., Курейчик В. М. Генетические алгоритмы / В. М. Курейчик // М.: ФИЗМАТЛИТ. – 2010. – С. 124–137.
4. Генетический алгоритм. Просто о сложном [Электронный ресурс] / . – Режим доступа: <https://habr.com/ru/post/128704/> – Дата доступа: 10.10.2021.