

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Дударевич А. А.

Научный руководитель – Стрелюхин А. В.
Белорусский национальный технический университет
г. Минск, Беларусь

Аннотация. Генетический алгоритм является самым известным на данный момент представителем эволюционных алгоритмов и по своей сути является алгоритмом для нахождения глобального экстремума многоэкстремальной функции. Он заключается в параллельной обработке множества альтернативных решений. При этом поиск концентрируется на наиболее перспективных из них. Это говорит о возможности использования генетических алгоритмов при решении любых задач искусственного интеллекта, оптимизации, принятия решений.

Введение

Генетический алгоритм – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе [1]. Он является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и кроссинговер [2].

Из этого определения вытекает основная идея алгоритма: поиск оптимального, но не лучшего, решения путём естественного отбора.

Основными сферами использования генетического алгоритма являются:

- задачи на графы;
- задачи компоновки;
- создание «Искусственного интеллекта».

Общий принцип действия алгоритма

Задача формализуется таким образом, чтобы её решение могло

быть закодировано в виде вектора генов, где каждый ген может быть битом, числом или неким другим объектом. Некоторым, обычно случайным, образом создаётся множество генотипов начальной популяции [3]. Они оцениваются с использованием «функции приспособленности», в результате чего с каждым генотипом ассоциируется определённое значение («приспособленность»), которое определяет насколько хорошо фенотип, им описываемый, решает поставленную задачу.

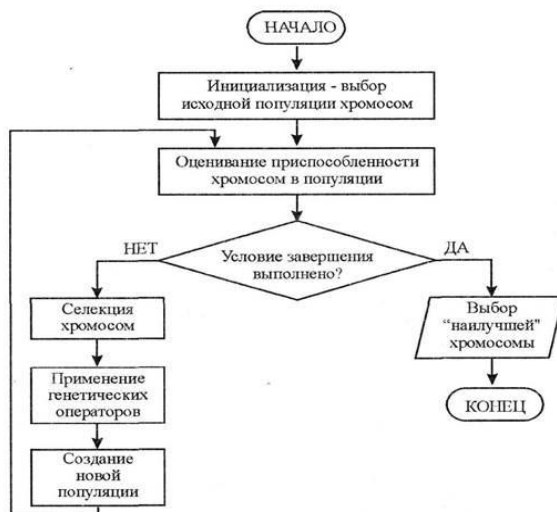
Алгоритм делится на 3 этапа:

- формирование нового поколения;
- отбор;
- скрещивание.

Алгоритм работает до тех пор, пока не выполнятся одно из условий:

- количество итераций достигнет выбранного максимума;
- достижение результата;
- исчерпание времени на мутацию.

Ниже приведена блок-схема алгоритма (рисунок 1).



**Рисунок 1. – Блок-схема генетического алгоритма
Стадии выполнения алгоритма**

Создание новой популяции. На этом шаге создается начальная популяция, которая будет абсолютно случайной, однако велика вероятность, что алгоритм эту проблему исправит. Главное, чтобы они соответствовали «формату» и были «приспособлены к размножению». Чем больше популяция, тем скорее алгоритм придёт к решению поставленной задачи.

Отбор. На этом этапе начинается процесс эволюции, в котором создается выборка из наиболее приспособленных особей из популяции. Выборка происходит при помощи алгоритма, который определяет успешность особи относительно средней успешности популяции.

Размножение. Размножение в генетических алгоритмах требует для производства потомка нескольких родителей, обычно двух. Можно выделить несколько операторов выбора родителей:

Панмиксия — оба родителя выбираются случайно, каждая особь популяции имеет равные шансы быть выбранной.

Инбридинг — первый родитель выбирается случайно, а вторым выбирается такой, который наиболее похож на первого родителя.

Аутбридинг — первый родитель выбирается случайно, а вторым выбирается такой, который наименее похож на первого родителя.

Мутации. Чтобы полученные в результате отбора особи не выродились и не скатывались в одно определённое решение, существуют мутации. Мутации, в зависимости от их параметра, изменяют вектора генов особей из доминирующей выборки, что в свою очередь помогает популяции не вырождаться и не следовать по пути одного решения.

Выполняя эту последовательность действий, генетический алгоритм приходит к оптимальному решению, так же как и эволюция привела человека к верховенству над другими видами.

Пример реализации алгоритма

Покажем наш алгоритм в упрощённом виде без скрещивания на практике. Для этого на рисунке 2 будет приведён простейший код на языке программирования C++ с комментариями, поясняющими происходящее [1].

```
#include<cstdlib>
```

```

#include<ctime>
#include<algorithm>
#include<iostream>
#include<numeric>

intmain()
{
    srand((unsignedint)time(NULL));
    constsize_t N =1000;
    int a[N] = { 0 };
    for ( ; ; )
        {
            //мутация в случайную сторону каждого элемента:
            for (size_t i=0; i< N; ++i)
                a[i] += ((rand() %2==1) ?1:-1);
            //теперь выбираем лучших, отсортировав по возрастанию
            std::sort(a, a + N);
            //и тогда лучшие окажутся во второй половине массива.
            //скопируем лучших в первую половину, куда они оставили потом-
            ство, а первые умерли:
            std::copy(a + N /2, a + N, a);
            //теперь посмотрим на среднее состояние популяции. Как видим,
            оно всё лучше и лучше.
            std::cout<<std::accumulate(a, a + N, 0) / N <<std::endl;
        }
}

```

Рисунок 2. – Простейший пример реализации генетического алгоритма на языке программирования C++

Заключение

Несмотря на то, что модель эволюционного развития, применяемая в генетических алгоритмах, сильно упрощена по сравнению со своим природным аналогом, тем не менее генетический алгоритм является достаточно мощным средством и может с успехом применяться для широкого класса прикладных задач, включая те, которые

трудно, а иногда и вовсе невозможно, решить другими методами. Однако, генетический алгоритм, как и другие методы эволюционных вычислений, не гарантирует обнаружения глобального решения за лучшее 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 методами, почти всегда такие методы будут эффективнее чем генетический алгоритм и в быстрой реакции, и в точность найденных решений. Главным же преимуществом генетических алгоритмов является то, что они могут применяться даже на сложных задачах, для которых не существует никаких специальных методов. В задачах, где хорошо работают существующие методики, результат можно улучшить сочетанием их с генетическим алгоритмом.

Литература

1. Википедия. Свободная энциклопедия. Генетический Алгоритм [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC. – Дата доступа: 06.04.2021.

2. Применение генетических алгоритмов для обучения нейронных сетей [Электронный ресурс]. – Режим доступа: http://masters.donntu.org/2008/kita/dzhura/library/ref_2.htm. – Дата доступа: 06.04.2021.

3. Популярно о генетических алгоритмах [Электронный ресурс]. – Режим доступа: <http://algolist.manual.ru/ai/ga/ga1.php#ga>. – Дата доступа: 13.04.2021.