

Генетический алгоритм – эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искоемых параметров с использованием механизмов, аналогичных естественному отбору в природе. Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и кроссинговер. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

Генетические алгоритмы применяются для решения следующих задач:

1. Оптимизация функций
2. Оптимизация запросов в базах данных
3. Разнообразные задачи на графах (задача коммивояжера, раскраска, нахождение паросочетаний)
4. Настройка и обучение искусственной нейронной сети
5. Задачи компоновки
6. Составление расписаний
7. Игровые стратегии
8. Теория приближений
9. Искусственная жизнь
10. Биоинформатика (фолдинг белков)
11. Синтез конечных автоматов

Вот основные компоненты и шаги генетического алгоритма:

1. Представление решений: Решения представляются в виде генетических строк или хромосом, которые состоят из генов. Гены могут быть битами, числами или любыми другими формами представления данных.
2. Определение функции приспособленности: Функция приспособленности (fitness function) оценивает качество каждого решения в

популяции. Она определяет, насколько хорошо решение соответствует требованиям задачи оптимизации.

3. Инициализация популяции: Начальная популяция решений создается случайным образом. Чем более разнообразны решения в популяции, тем лучше.

4. Операторы селекции: Операторы селекции выбирают родительские решения для размножения на основе их приспособленности. Часто используется метод рулеточного колеса, где вероятность выбора решения пропорциональна его приспособленности.

5. Операторы скрещивания: Операторы скрещивания (кроссовер) комбинируют гены родительских решений, чтобы создать потомство. Различные методы скрещивания определяют, как гены родителей будут комбинироваться.

6. Операторы мутации: Операторы мутации случайным образом изменяют гены потомства, чтобы внести некоторую степень разнообразия и избежать преждевременной сходимости. Мутации помогают исследовать новые регионы пространства решений.

7. Формирование новой популяции: С использованием операторов селекции, скрещивания и мутации создается новая популяция решений, заменяя предыдущую.

8. Повторение: Шаги 4-7 повторяются до достижения критерия останова, который может быть достижение определенного количества поколений, достижение определенного уровня приспособленности или достижение заданного времени выполнения.

9. Оценка и выбор лучшего решения: После завершения итераций генетического алгоритма, оценивается приспособленность каждого решения в конечной популяции. Затем выбирается лучшее решение, которое является оптимальным или приближенно оптимальным решением задачи оптимизации.

10. Возвращение оптимального решения: Оптимальное решение возвращается в качестве результата генетического алгоритма, которое может быть использовано для принятия решений или решения задачи оптимизации.

Алгоритм делится на три этапа:

1. Скрещивание
2. Селекция (отбор)
3. Формирования нового поколения

Если результат нас не устраивает, эти шаги повторяются до тех пор, пока результат нас не начнет удовлетворять или произойдет одно из ниже перечисленных условий:

- Количество поколений (циклов) достигнет заранее выбранного максимума.
- Исчерпано время на мутацию.