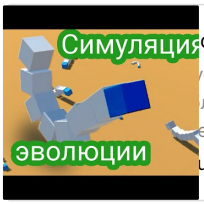
Генетический алгоритм

Сделал генетический алгоритм | симуляция ЭВОЛЮЦИИ
В этом видео я использую генетический алгоритм и нейросеть, чтобы существа самостоятельно эволюционировали и обучались выполнять задачу.Изначально они не даж...
https://www.youtube.com/watch?v=JaPwn-pvHTs



ственного отбора [Озвучка Primer]
outu.be/0ZGbIKd0XrMКак будут вести
ладающие тремя различающимися
е из них будет наиболее важным в
utube.com/watch?v=PyR0EQInCSY

Здесь нет теории про написание кода, но в целом интересно

Genetic Algorithms - Introduction
Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and
https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm

https://en.wikipedia.org/wiki/Genetic_algorithm

Medium
https://towardsdatascience.com/genetic-algorithm-implementation-in-python-5ab67bb124a6?gi=4ed7f090739c

Задание

Введение

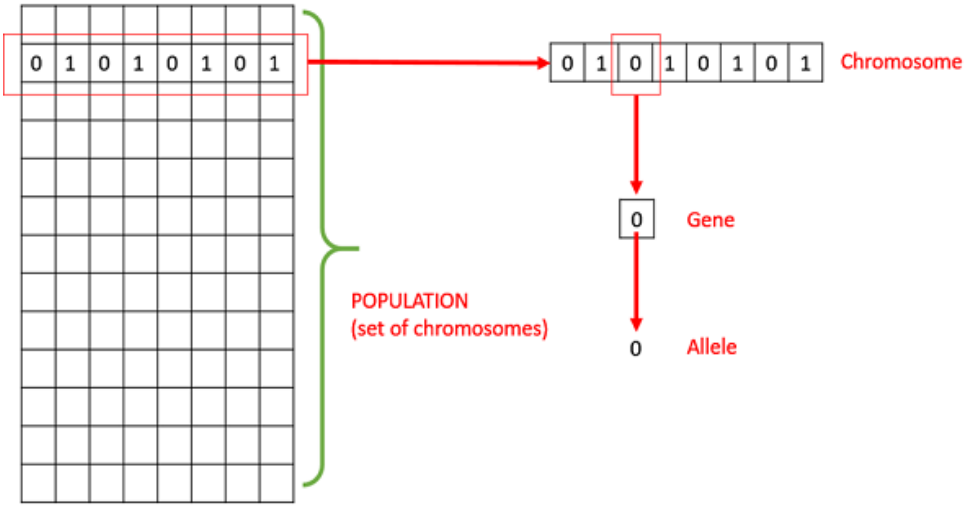
Генетический алгоритм является техникой оптимизации, позволяющей улучшить показатели работы решения за счёт применения механизма естественного отбора. Оптимизируется при этом некоторая **целевая функция**.

Сам принцип основан на **эволюционном вычислении**. Изначально мы обладаем набором решений описанной задачи. Эти решения скрещиваются между собой, с ними происходят мутации и этим они производят новое поколение - новый **набор решений**.
В каждом поколении решениям присваиваются значения - веса, определяющие их **ценность**. В зависимости от этого значения конкретное решение будет иметь больший или меньший шанс произвести потомство - новые решения.
Это повторяется до тех пор, пока не будет выполнено **условие завершения**.

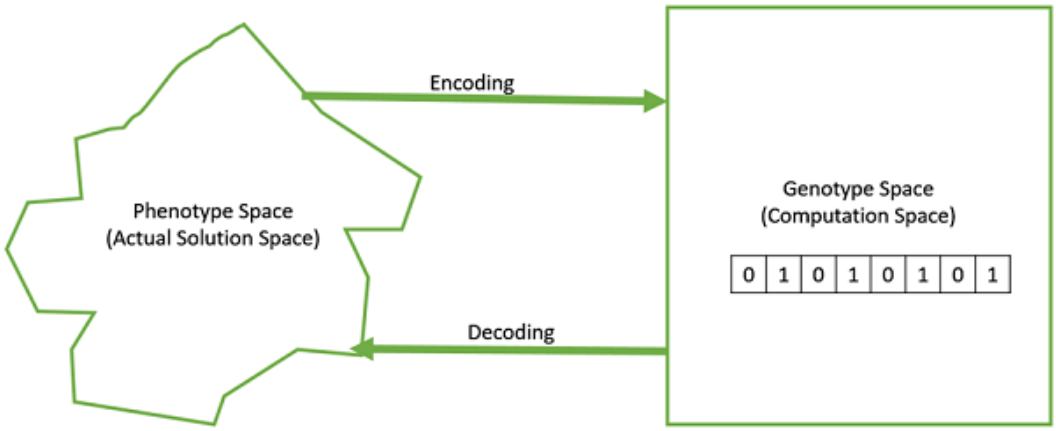
Базовые понятия

- **Популяция** - подмножество всех возможных решений описанной задачи

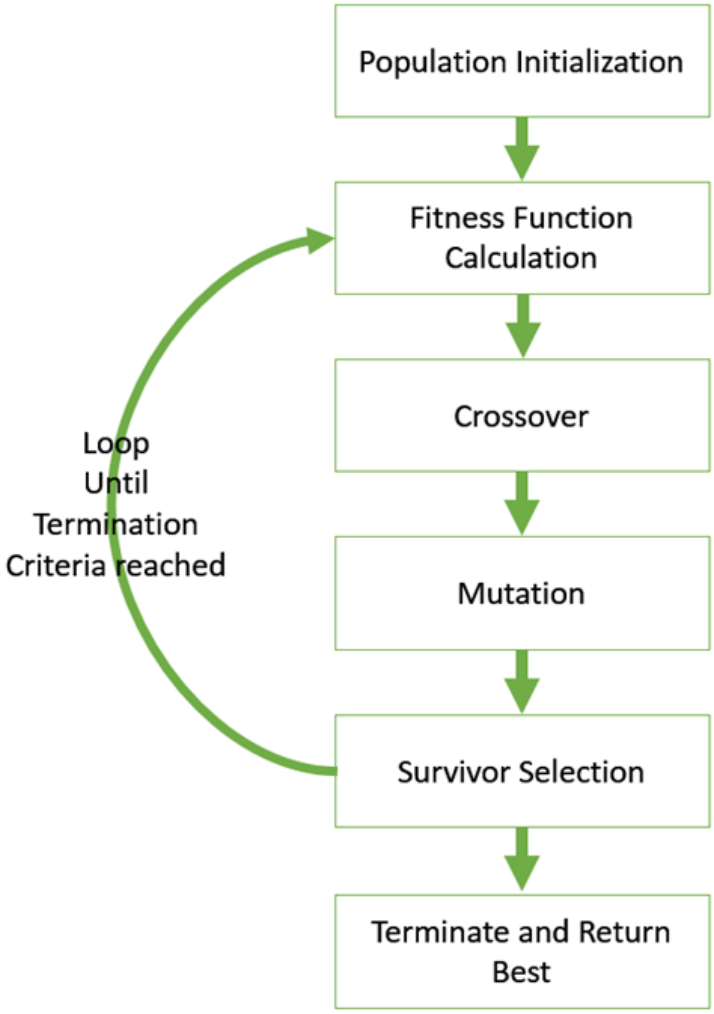
- **Хромосома** (геном) - одно из решений задачи
- **Ген** - составная часть хромосомы (генома)
- **Адель** - богиня, **аллель** - значение гена в рамках хромосомы



- **Генотип** - популяция в вычислительном представлении
- **Фенотип** - популяция в естественном представлении
- **Декодирование** - трансформации решения из генотипа в фенотип
- **Кодирование** - трансформация решения из фенотипа в генотип



- **Функция приспособления** - принимает хромосому и возвращает значение, отражающее степень её **ценность**; значение показывает, "насколько хорошо" или "насколько плохо" данное решение удовлетворяет условиям задачи
- **Генетические операторы** - собственно генетическая композиция - мутации, скрещивание, выбор



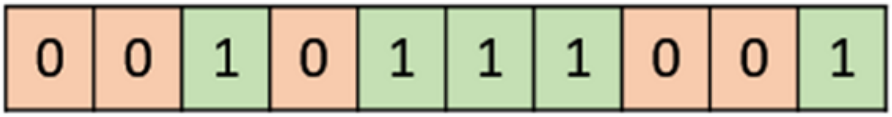

```
GA()
   initialize population
   find fitness of population

   while (termination criteria is reached) do
      parent selection
      crossover with probability pc
      mutation with probability pm
      decode and fitness calculation
      survivor selection
      find best
   return best
```

Представление алгоритма

Бинарное представление

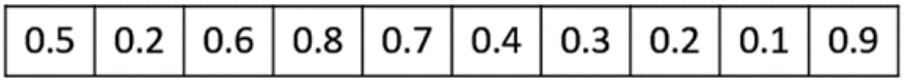
Является наиболее простым и широко распространённым способом представления генетического алгоритма. Генотип в этом представлении состоит из набора строк, в которых записаны биты.

Представление удобно, когда, например, фенотип представлен значениями "да" и "нет" (1 и 0 соответственно)



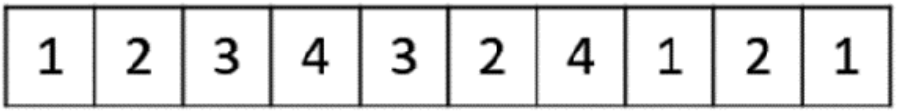
Вещественное представление

Если необходимо интерполирование. В целом точность зависит от вычислительных способностей системы.



Целочисленное представление

Расширение бинарного представления, когда есть значения, отличные от "да" и "нет".



Популяция

Это набор возможных решений (хромосом). При работе с популяцией важно держать во внимании следующие факторы:

- Отличия между индивидами в популяции должны поддерживаться и контролироваться, иначе их отсутствие приведёт к преждевременному схождению к одному образу
- Размер популяции не должен быть слишком большим, чтобы не замедлять работу алгоритма, но и не слишком маленьким, чтобы можно было найти пару для скрещивания. Подходящий размер определяется методом проб и ошибок

Популяция обычно определяется с помощью матрицы $m \cdot n$, где m - размер популяции, а n - количество генов в хромосоме.

Инициализация популяции

- **Случайная инициализация** - подразумевает наполнение начальной популяции полностью случайными комбинациями хромосом

- **Эвристическая инициализация** - начальная популяция наполняется уже известной эвристикой - заданными условиями - для описанной задачи

В ходе наблюдений было установлено, что **эвристическую инициализацию** следует проводить с некоторым набором случайных значений, так как это повышает степень разрозненности решений. И в общем случае случайная инициализация выигрывает, так как позволяет найти больше новых \ подходящих решений из-за повышенной разрозненности.

Популяционные модели

- **Модель устойчивого состояния** - в каждом поколении появляются несколько (1-2) потомков, которые **заменяют** такое же количество уже существующих решений
- **Поколенческая модель** - генерируется n потомков, которые полностью заменяют прошлое поколение; n- размер популяции

Функция оценки

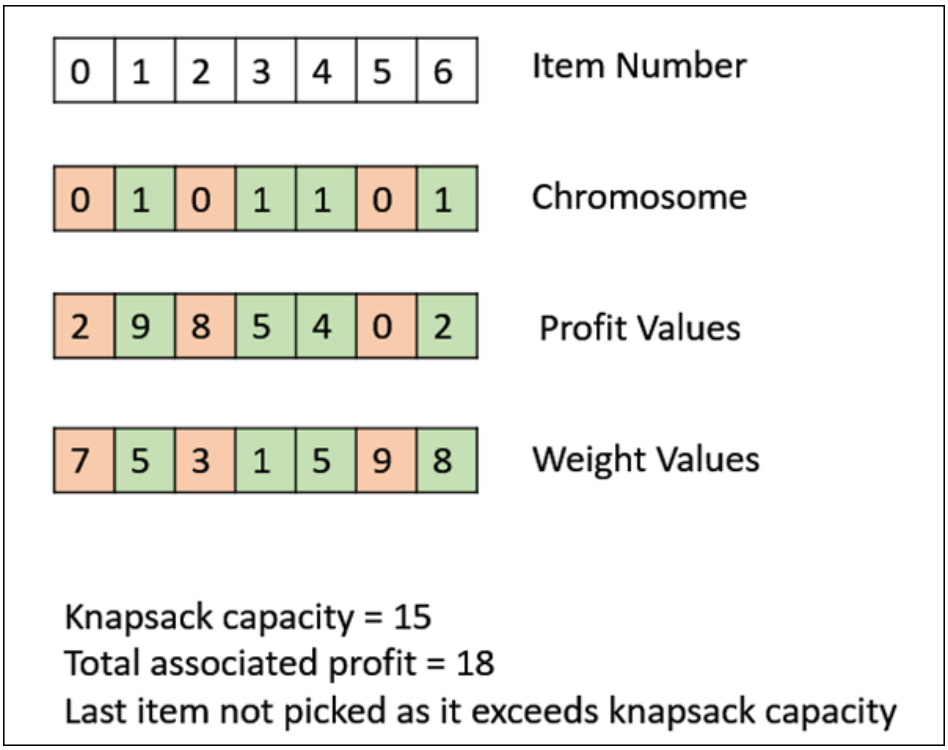
https://towardsdatascience.com/how-to-define-a-fitness-function-in-a-genetic-algorithm-be572b9ea3b4

Функция должна быть достаточно быстрой, чтобы не замедлять работу всего алгоритма.

Часто функция оценки и целевая функция - одни и те же. Но это может быть не так, когда у алгоритма несколько целей оптимизации.

Функция оценки должна удовлетворять следующим характеристикам:

- Быть достаточно быстрой для вычислений
- Полно измерять ценность решения



Селекция родителей

Selection (genetic algorithm) - Wikipedia

Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (using the crossover operator). A generic selection procedure may be implemented as follows: The fitness function is evaluated for each individual, providing fitness values, which are then normalized.

𝕎 https://en.wikipedia.org/wiki/Selection_(genetic_algorithm)

Это процесс выбора индивидов из текущего поколения для их рекомбинации и создания потомства для следующего поколения.

Необходимо внимательно подходить к скрещиванию, так как между поколениями может сохраняться решение, которое отлично подходит для решения задачи, но из-за этого остальные решения становятся похожими на него, что в конечном счёте приводит к тому, что алгоритм не оправдывает себя; этот случай называется **преждевременной конвергенцией**.
Поддержание достаточно высокой степени разности в решениях важно для успешности алгоритма.

Пропорциональная селекция

Один из наиболее популярных подходов к селекции. В нём каждое решение может стать родителем с вероятностью, пропорциональной ценности этого решения. Таким образом, более подходящие решения имеют более высокий шанс дать потомство. Тем самым, со временем остаются только те решения, которые больше других удовлетворяют условиям.

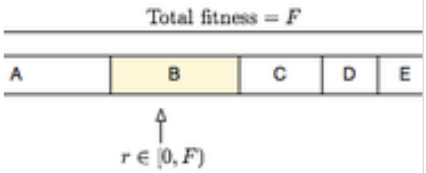
У этой селекции есть несколько реализаций.

Колесо фортуны

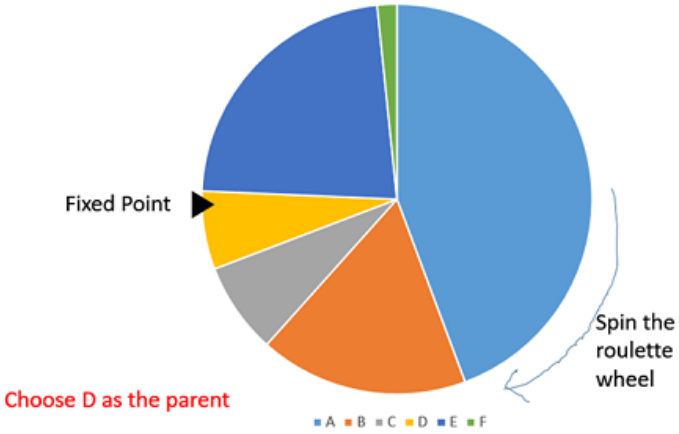
Fitness proportionate selection - Wikipedia

Fitness proportionate selection, also known as roulette wheel selection, is a genetic operator used in genetic algorithms for selecting potentially useful solutions for recombination. In fitness proportionate selection, as in all selection methods, the fitness

𝕎 https://en.wikipedia.org/wiki/Fitness_proportionate_selection



На колесе каждое из решений занимает область, пропорциональную его ценности.



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Где-то на колесе выбирается фиксированная точка. Колесо крутится и после его остановки в родители попадает то решение, на область которого указывает фиксированная точка.

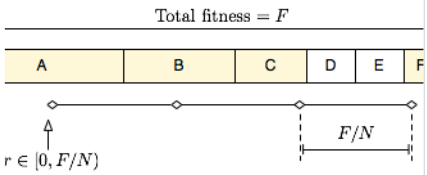
```
for all members of population
    sum += fitness of this individual
end for

for all members of population
    probability = sum of probabilities + (fitness / sum)
    sum of probabilities += probability
end for

loop until new population is full
    do this twice
        number = Random between 0 and 1
        for all members of population
            if number > probability but less than next probability
                then you have been selected
        end for
    end
    create offspring
end loop
```

Стохастическая выборка

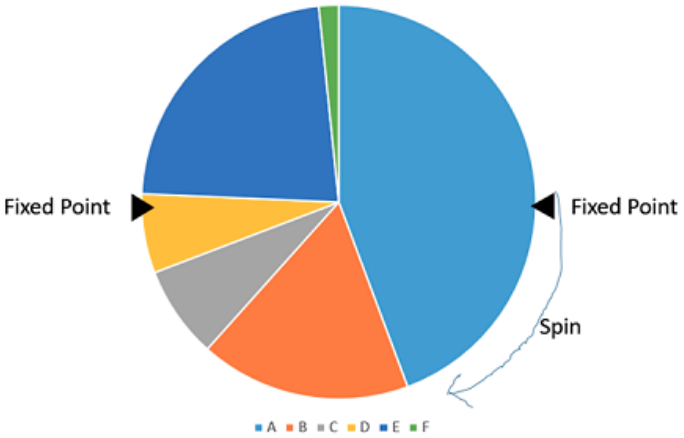
Stochastic universal sampling - Wikipedia

Stochastic universal sampling (SUS) is a technique used in genetic algorithms for selecting potentially useful solutions for recombination. It was introduced by James Baker. SUS is a development of fitness proportionate selection (FPS) which exhibits no

𝕎 https://en.wikipedia.org/wiki/Stochastic_universal_sampling



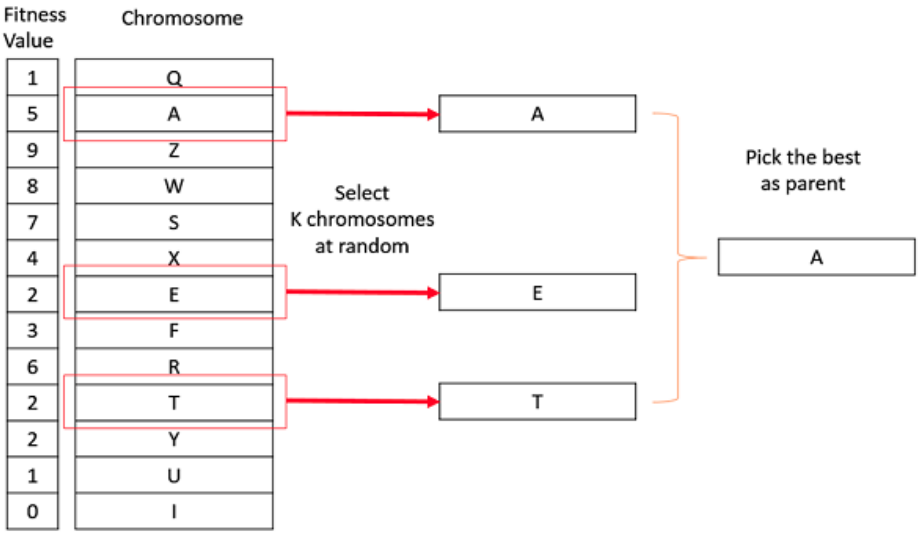
Почти как колесо фортуны, только выбираются две фиксированные точки. Колесо крутится и выбираются хромосомы, которые попали в точки.



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Турнирная селекция

Выбираются K хромосом и из их числа выбирается та, у которой наибольшая ценность.



Ранговая селекция

Selection (genetic algorithm) - Wikipedia

Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (using the crossover operator). A generic selection procedure may be implemented as follows: The fitness function is evaluated for each individual, providing fitness values, which are then normalized.

𝕎 https://en.wikipedia.org/wiki/Selection_(genetic_algorithm)#Rank_Selection

Теряюсь в догадках. Суть в том, что ценность обязательно на n-й итерации поколений у решений станет одинаковой и они будут занимать одинаковую площадь на колесе. И чтобы этого избежать, вместо ценностей используется ранг. По сути - число, которое присваивается решению случайным (?) образом.

Случайная селекция

Здесь нет никаких правил, каждый родитель выбирается **абсолютно** случайно.

Скрещивание

Crossover (genetic algorithm) - Wikipedia

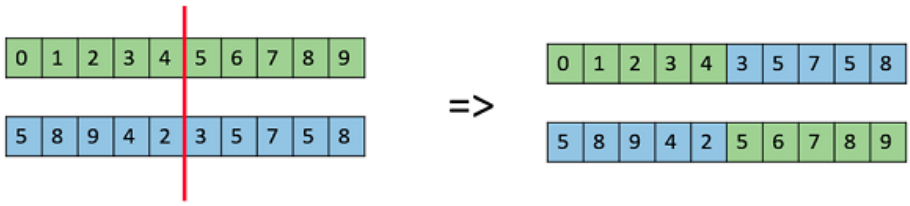
In genetic algorithms and evolutionary computation, crossover, also called recombination, is a genetic operator used to combine the genetic information of two parents to generate new offspring. It is one way to stochastically generate new solutions from an existing population, and is analogous to the crossover that happens during sexual reproduction in biology.

𝕎 https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)

Функция скрещивания обладает такими же свойствами, что и биологический аналог. Потомки создаются из генетического материала, которым обладают родители. И есть несколько видов скрещивания.

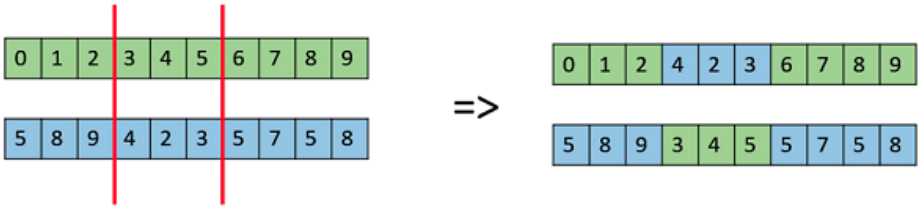
Одноточечное скрещивание

При этом скрещивании в хромосоме выбирается (случайно или предопределённо) точка, которая делит хромосому на голову и хвост. Две хромосомы обмениваются хмостами и получаются потомки.



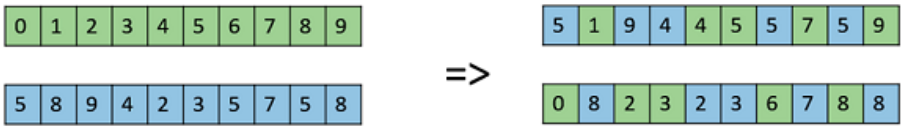
Многоточечное скрещивание

Выбираются несколько точек для разбиения и соответственно несколько получившихся частей меняются между хромосомами.



Однородное скрещивание

Хромосома не делится на сегменты, вместо этого каждый ген рассматривается отдельно и случайным образом меняется с соответствующим геном в соседней хромосоме.



Пропорциональное скрещивание

Беруются хромосомы родителей и считается их взвешенная сумма

Weighted sum model - Wikipedia

In decision theory, the weighted sum model (WSM), also called weighted linear combination (WLC) or simple additive weighting (SAW), is the best known and simplest multi-

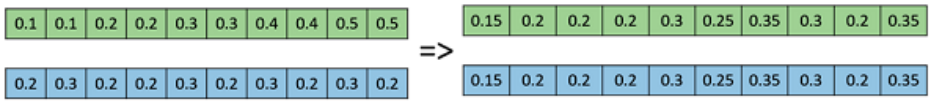
𝕎 https://en.wikipedia.org/wiki/Weighted_sum_model

Весовая функция - Википедия

Весовая функция - математическая конструкция, используемая при проведении суммирования, интегрирования или усреднения с целью придания

𝕎 https://ru.wikipedia.org/wiki/Весовая_функция

$$Child1 = \alpha \cdot x + (1 - \alpha) \cdot y$$
$$Child2 = \alpha \cdot y + (1 - \alpha) \cdot x$$

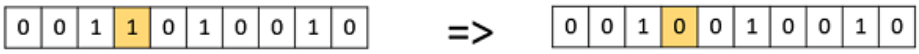


Мутация

Мутация - это небольшое изменение генов в хромосоме с целью получения нового решения. Используется для того, чтобы внести разнообразие в популяцию.

Мутация смены бита

Значение бита меняется на противоположение, применимо при бинарном кодировании генотипа.

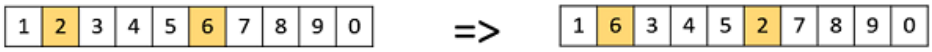


Случайная перестановка

Расширение бинарной мутации на целочисленную кодировку, при которой отдельному гену присваивается новое значение из списка допустимых аллелей.

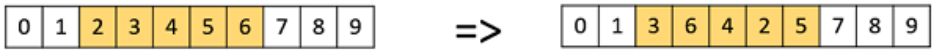
Мутация обмена

Выбираются несколько генов, значения которых меняются местами.



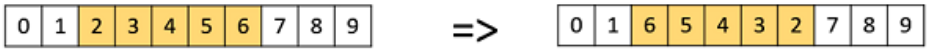
Скрэмбл

Выбирается подмножество генов, значения которых тасуются в случайном порядке.



Инверсия

Выбирается подмножество генов и они переставляются в обратном порядке ю



Замещение

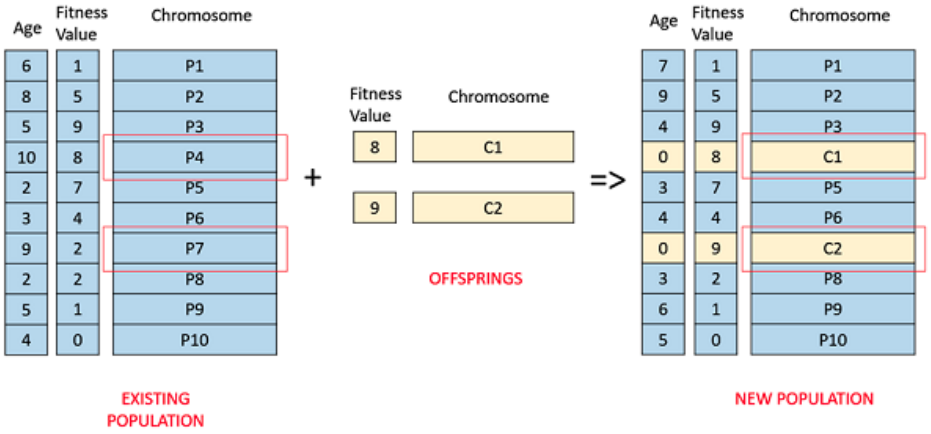
Это процесс определения того, какие решения будут оставлены в следующем поколении, а какие будут отброшены. Это необходимо, чтобы поддерживать более подходящие решения, но при этом важно сохранять разнообразие остающихся решений.

Иногда применяется принцип **элитарности**. Он подразумевает, что решение, обладающее самой высокой ценностью, **обязательно** попадёт в следующее поколение.

Но в целом наиболее простым подходом является отбрасывание случайных решений, но он часто приводит к проблемам конвергенции.

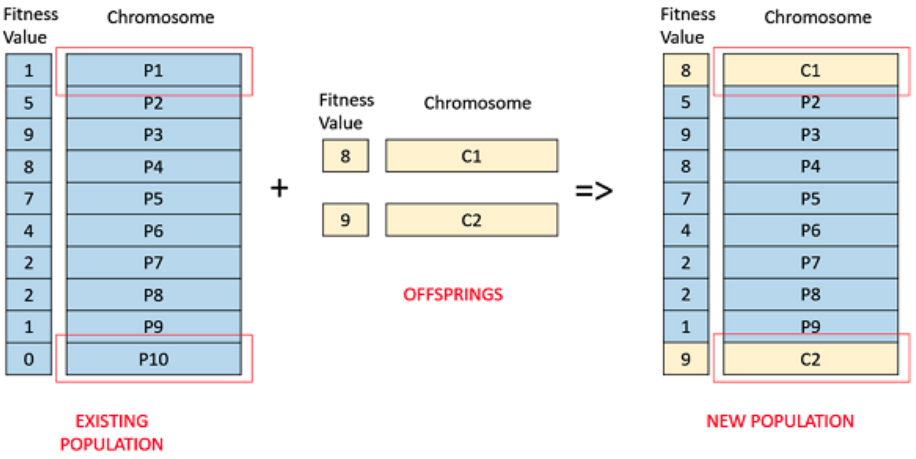
Замещение по возрасту

Здесь не используется понятие ценности, а вводится ограничение по возрасту. Это ограничение определяет, какое максимальное количество поколений может прожить каждое решение. При достижении своего максимума решение отбрасывается вне зависимости от того, насколько оно было ценным. На их место могут вставать их потомки либо их места будут оставаться пустыми.



Замещение по ценности

В этом типе наименее ценные предки заменяются наиболее ценными потомками.



Условие завершения

Это условие необходимо, чтобы знать, когда алгоритму следует завершить свою работу (иначе он может работать бесконечно).

Из наблюдений установлено, что на начальных этапах алгоритм в большом количестве производит новые решения, отличные от прошлых. И с ростом числа итераций разница между решениями уменьшается, что в какой-то момент делает дальнейшую работу алгоритма неэффективной.

Обычно рассматриваются следующие условия завершения:

- Когда не появляется улучшений между поколениями на X итерации
- Когда достигается заданное заранее число итераций
- Когда целевая функция достигла заданного заранее значения