Генетические алгоритмы

Задача оптимизации

Задача оптимизации - задача по нахождению экстремумов (минимума/максимума) целевой функции в некотором векторном конечномерном пространстве

Виды решения:

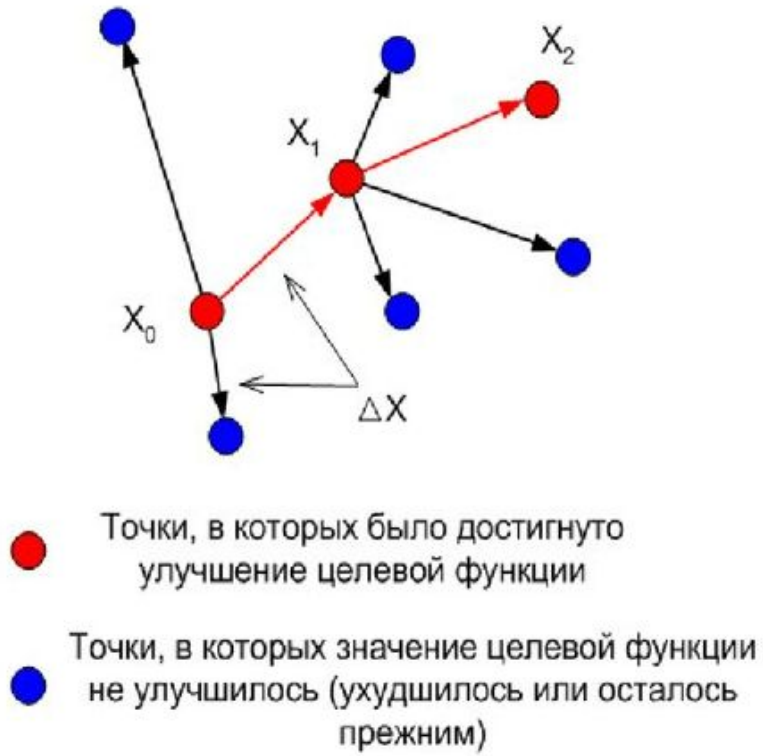
- Аналитические
- Численные
- Графические

Способы оптимизации

- Аналитические способы - точное нахождение решения по известному алгоритму, например, метод наименьших квадратов в линейной регрессии
- Градиентные способы - приближенный поиск решения путем итеративного вычисления градиентов/матрицы Гессе
- Решетчатый поиск - систематическое исследование пространства решений
- Случайный поиск - поиск путем случайного выбора решений

Случайный поиск

- При случайном поиске выбирается набор случайных векторов для смещения параметров, и происходит изменение по направлению наибольшего улучшения целевой функции.
- Одна из модификаций, координатный спуск. Изменения происходит только по одному из параметров.



Генетические алгоритмы

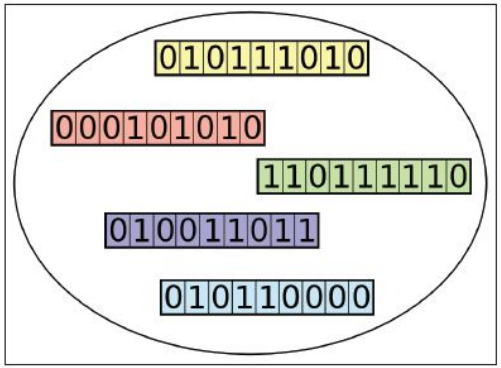
- Генетические алгоритмы (ГА) - семейство поисковых/оптимизационных алгоритмов, которые в основе содержат идею дарвиновской эволюции:
  - Изменчивость - в рамках одной популяции особи могут различаться и иметь уникальные признаки
  - Наследственность - свойства особей устойчиво передаются их потомкам. Поэтому потомки похожи на своих родителей.
  - Естественный отбор - особи с лучшими свойствами успешнее в борьбе, и могут породить новых потомков.
- Важные механизмы эволюции:
  - Скрещивание/рекомбинация - потом перенимает признаки родителей
  - Мутация - случайные вариации признаков

Терминология (1)

- Задача ГА - найти оптимальное решение некоторой задачи, путем развития популяции потенциальных решений называемых индивидуумами. Решения итеративно оцениваются и используются для создания следующего поколения.
- Генотип - описание одного индивидуума, набор генов сгруппированных в хромосому. При скрещивании хромосома содержит гены своих родителей.

Простое двоичное кодирование хромосомы

- Популяция - множество индивидуумов, то есть потенциальных решений, которые хранит генетический алгоритм. Популяция всегда отображает текущее поколение.



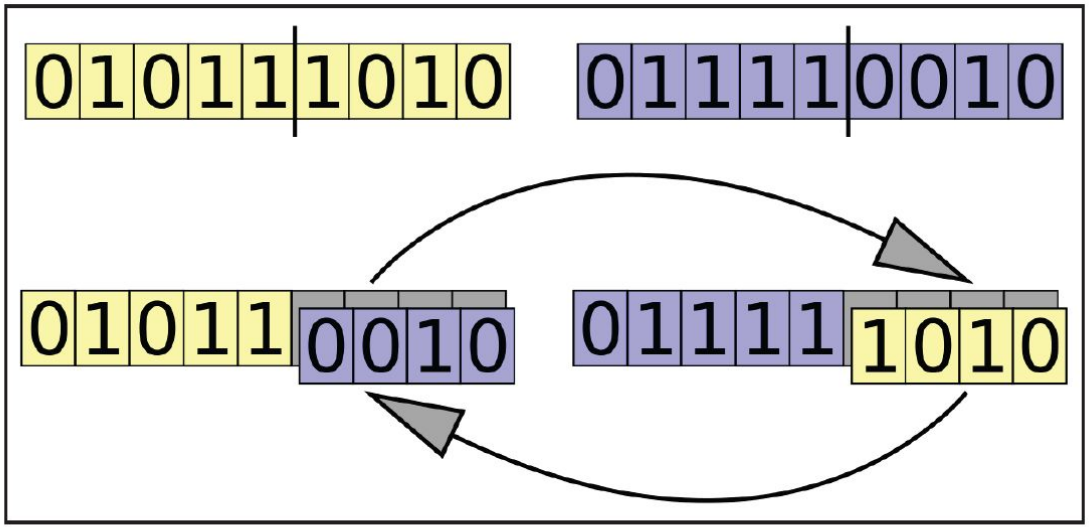
Терминология (2)

- Функция приспособленности/целевая функция - функция, которую необходимо оптимизировать в рамках решаемой задачи. Является функцией от индивидуума, и показывает качество решения представленным хромосомой. На каждой итерации ГА рассчитывает приспособленность индивидуума для формирования нового поколения.

- Отбор - формирование множества индивидуумов, которые будут использоваться для формирования следующего поколения. Отбор основывается на приспособленности индивидуума, и чем он лучше, тем больше вероятность его отобрать. Причем хромосомы дающие низкое значение приспособленности не исключают возможность отбора. Таким образом приспособленность популяции увеличивается.

Терминология - скрещивание

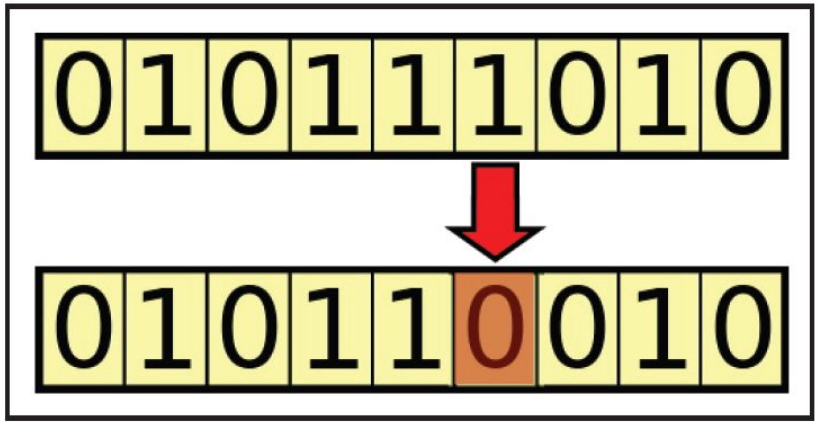
- Скрещивание - процесс создания пары новых индивидуумов (потомков), путем комбинации хромосом двух родителей из текущей популяции



Операция скрещивания двух двоично-кодированных хромосом

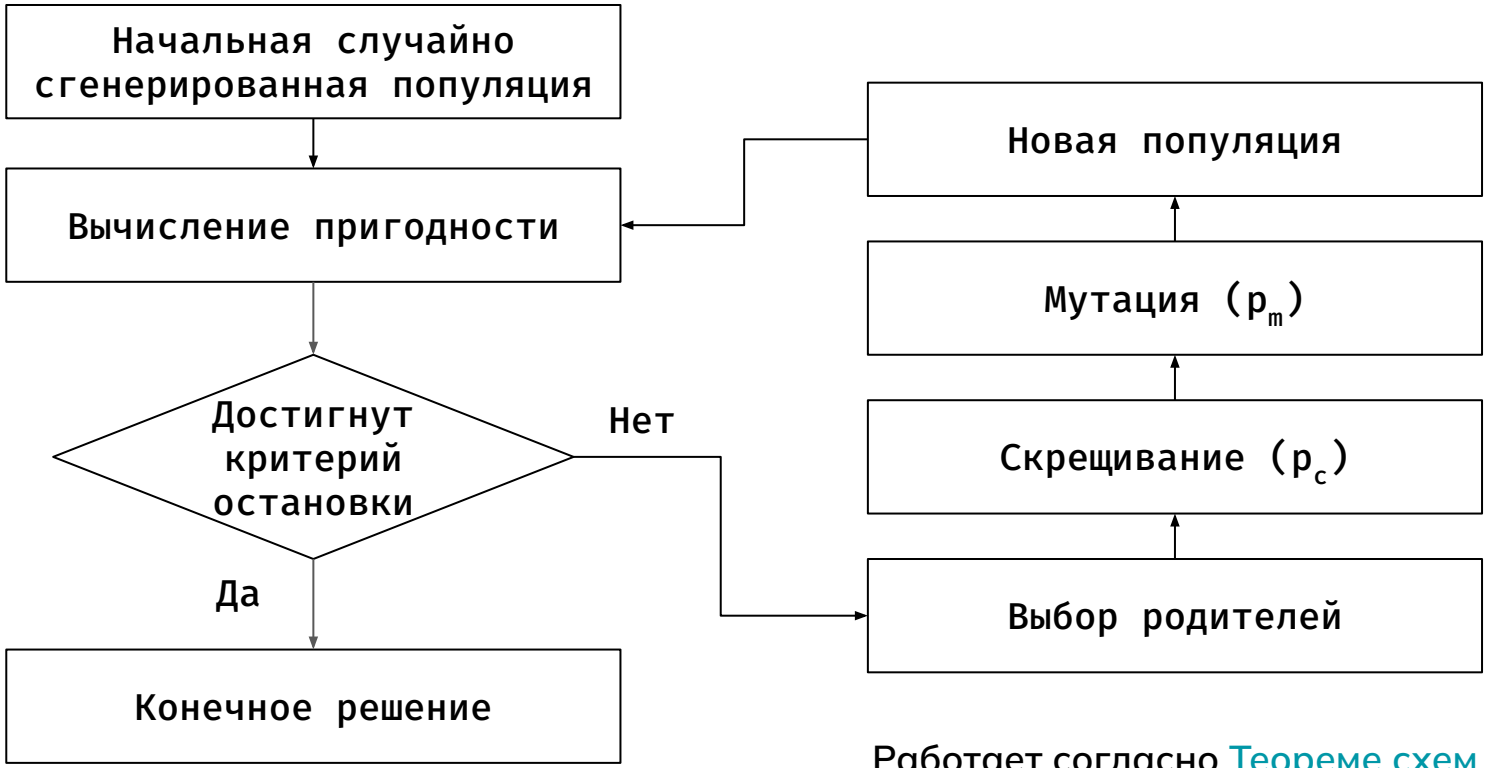
Терминология - мутация

- Мутация- процесс случайного изменения хромосомы, чтобы был поиск еще неисследованных областей, а также поддержание разнообразия популяции



Применение оператора мутации
к двоично-кодированной хромосоме

ГА - алгоритм в общем виде



Работает согласно Теореме схем

Особенность ГА

- В отличии от многих традиционных алгоритмов ГА позволяет:
  - Поддержание популяции решений - параллельная обработка множества возможных решений.
  - Генетическое представление - ГА работает с кодированным представлением решения. ГА не знают про предметную область.
  - Функция приспособленности - ГА работают только со значением приспособленности решения, поэтому не накладывают никаких ограничений ну целевую функцию.
  - Вероятностное поведения - ГА не детерминирован

Преимущества ГА

- Глобальная оптимизация - за счет работы с множеством решений, а также наличию мутаций решения. ГА способны не попадать в локальный экстремум, и находить лучшее решение задачи.
- Применимость к сложным задачам - так как ГА работает только со значением целевой функции, то можно решать задачи в которых функция не дифференцируема или трудно дифференцируема.
- Решение задач без мат. представления - можно решать задачи в которых нет математического представления, а также задачи где нет значения целевой функции, но можно сравнивать решения.
- Устойчивость к шуму
- Распараллеливание

Ограничения ГА

- Специальность определения - решение нужно закодировать, определить целевую функцию, способы скрещивания и мутации
- Настройка гиперпараметров - нет определенных алгоритмов подбора гиперпараметров
- Вычислительная сложность
- Преждевременная сходимость - в популяции может оказаться такое решение, которое более приспособлено чем любое другое из популяции, и в итоге может случится, что последующие популяции будут только из этого решения
- Отсутствие гарантированного решения - ГА не гарантирует, что глобальный оптимум найдется

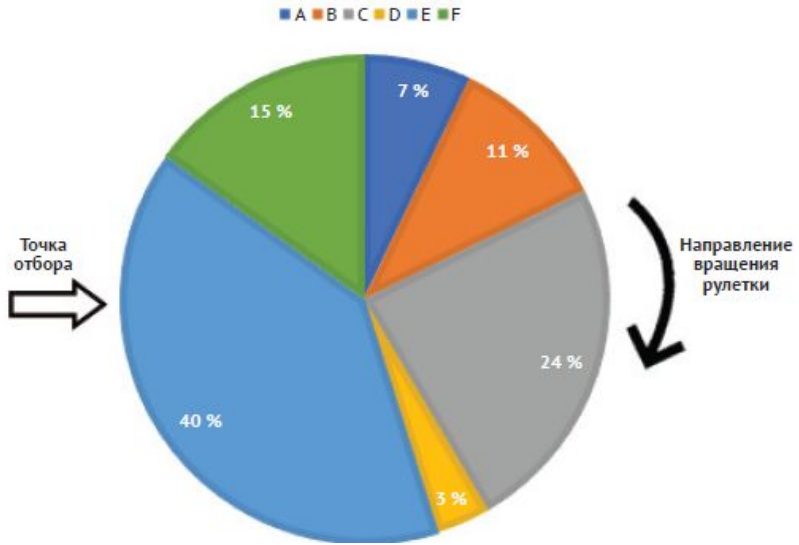
Применение ГА

- Решение задач в которых нет математического решения
- Решение задач в которых сложная целевая функция
- Решение комбинаторных задач
- Нахождение игровых стратегий
- Оптимизация параметров моделей машинного обучения
- Определение признаков для оптимизации модели машинного обучения
- Симуляция поведения

Метод отбора - Правило рулетки

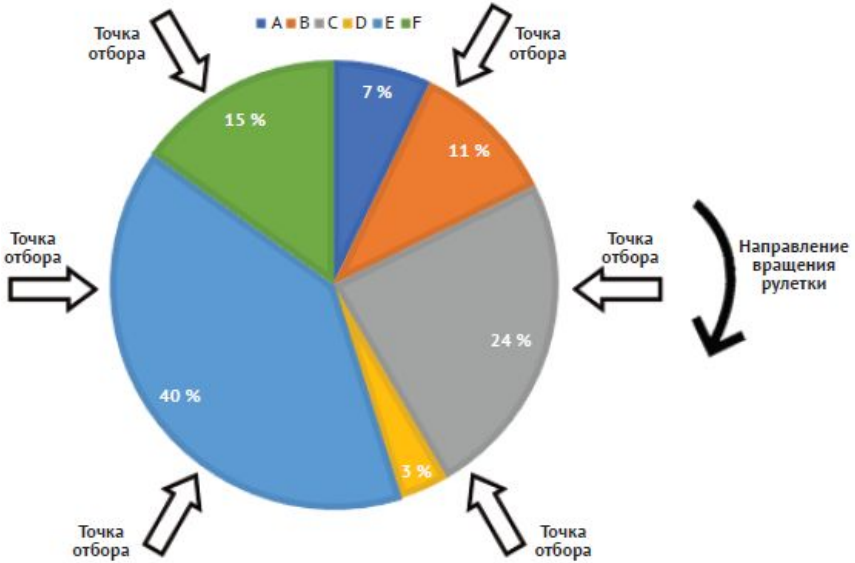
- Отбор пропорционально приспособленности
- Пока не наберется достаточное кол-во родителей, из популяции выбирается случайное решение с вероятностью пропорциональной приспособленности. Одно и то же решение может быть отобрано несколько раз.

Индивидуум	Приспособленность	Относительная доля
A	8	7 %
B	12	11 %
C	27	24 %
D	4	3 %
E	45	40 %
F	17	15 %



Метод отбора - Стохастическая универсальная выборка

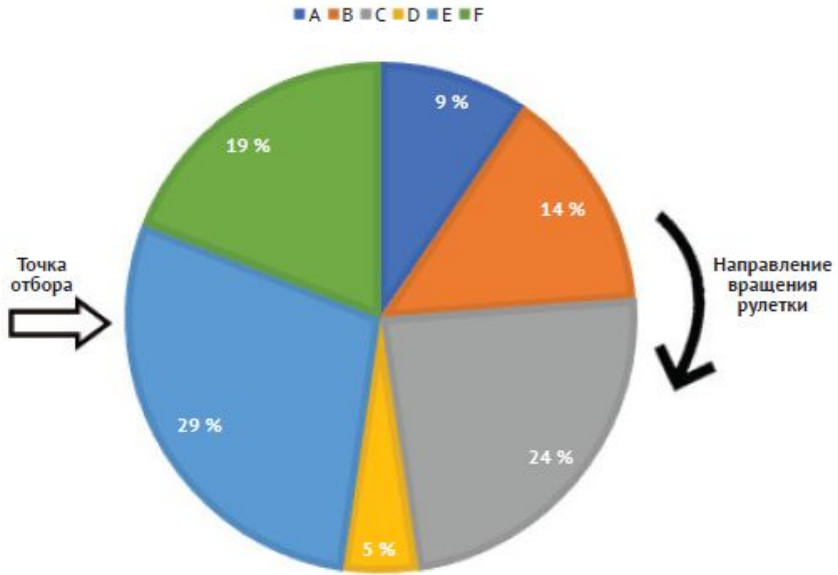
- Стохастическая универсальная выборка – модифицированный вариант правила рулетки.
- “Рулетка” крутится один раз, и за раз выбираются сразу все родители



Метод отбора - Ранжированный отбор

- В методе ранжированного отбора каждому решению сопоставляется ранг (порядковый номер после сортировки по возрастанию), и вероятность определяется на основе ранга

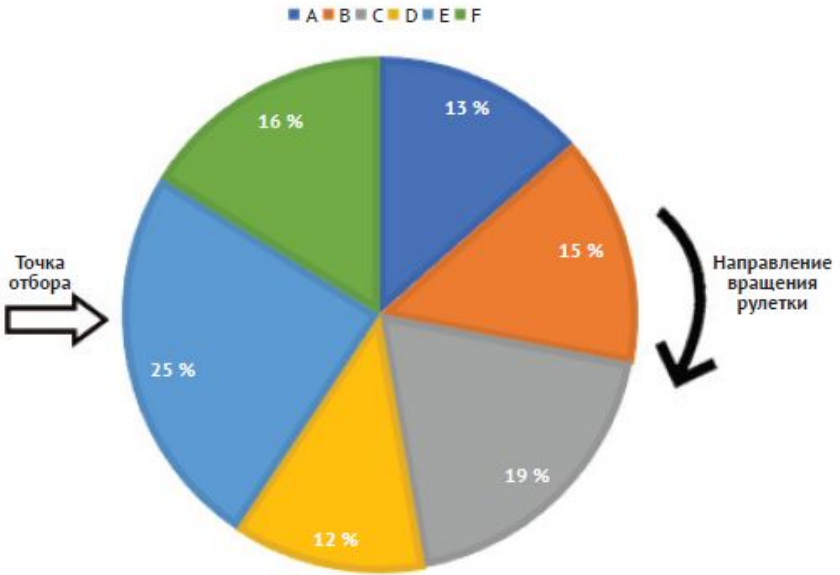
Индивидуум	Приспособленность	Ранг	Относительная доля
A	8	2	9 %
B	12	3	14 %
C	27	5	24 %
D	4	1	5 %
E	45	6	29 %
F	17	4	19 %



Метод отбора - Масштабирование приспособленности

- Значения всех приспособленностей линейно приводится к заданному диапазону. Можно гарантировать, например, что вероятность отбора худшего решения не меньше чем в 2 раза лучшего решения.

Индивидуум	Приспособленность	Масштабированная приспособленность	Относительная доля
A	8	55	13 %
B	12	60	15 %
C	27	78	19 %
D	4	50	12 %
E	45	100	25 %
F	17	66	16 %



Метод отбора - Турнирный отбор

- Случайной выбирается n (размер турнира) кандидатов, из них выбирается наилучшее решение. Алгоритм повторяется до тех пор, пока не будет сформировано следующее поколение.
- Данный метод может работать без самого значения приспособленности, только сравнивая решения попарно.

Индивидуум	Приспособленность
A	8
B	12
C	27
D	4
E	45
F	17

Метод отбора

- Отбор усечением - сначала из популяции убирается доля самых худших решений, а затем недостающие решения формируются из оставшихся лучших решений путем скрещивания и мутации

- Элитарный отбор - сначала в новую популяцию отбираются n лучших решений, а остальные решения создаются другим способом отбора. Причем лучшие отобранные решения также могут участвовать в скрещивании.

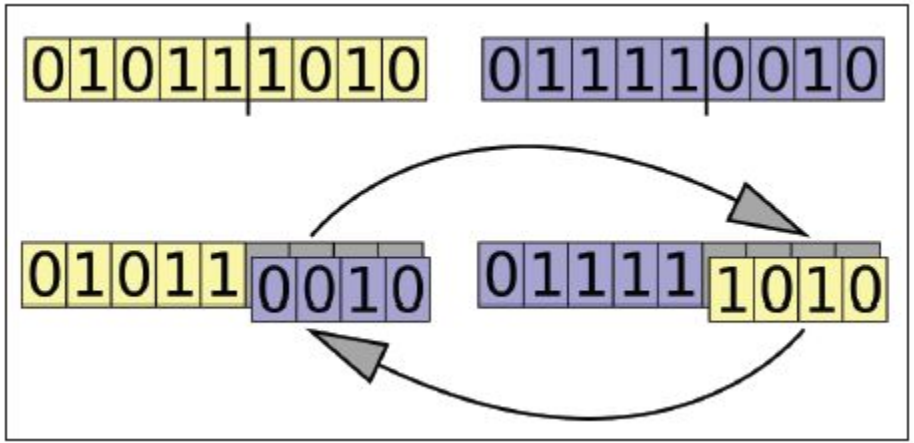
- Отбор вытеснением - сначала формируются потомки стандартным способом отбора, а затем в новую популяцию добавляются наиболее разнообразные решения.

Скрещивание

- Скрещивание для двух родителей происходит с некоторой заданной вероятностью p_c, если скрещивание не происходит, то родители передаются дальше в алгоритм без изменений.
- Выбор скрещивания зависит от тип хромосомы. Основные виды хромосомы:
  - Бинарная кодировка - решение представлено в виде массива 0 и 1
  - Порядковая кодировка - решение представлено в виде массива натуральных чисел, где учитывается их порядок
  - Вещественная кодировка - решение представлено в виде массива вещественных числа. Целые числа частный случай.

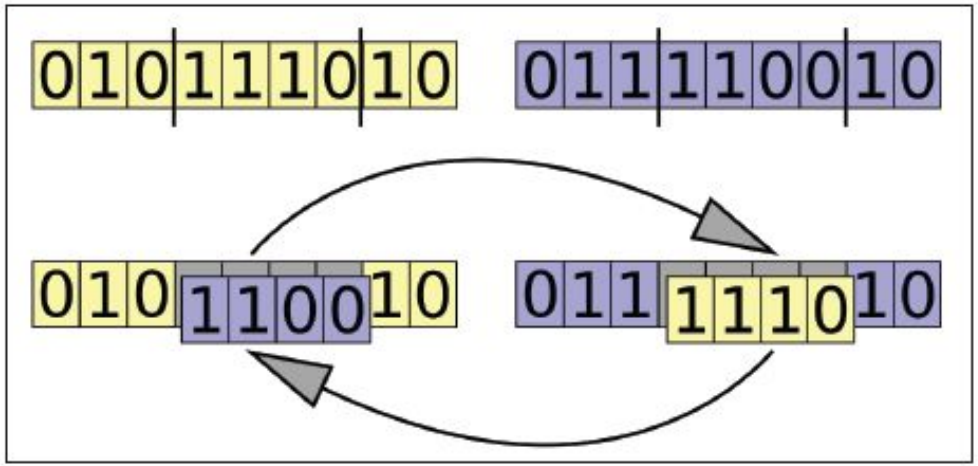
Методы скрещивания - Одноточечное скрещивание

- Подходит в основном для бинарной кодировки.
- Выбирается точка скрещивания хромосомы (индекс элемента), получается две части до и после индекса. Для формирования потомков, родители обмениваются одной из частей.



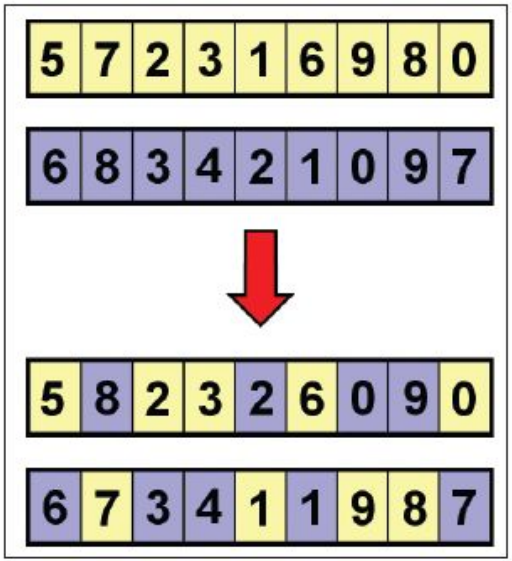
Методы скрещивания - Двухточечное скрещивание

- Принцип такой же как и у одноточечного скрещивания, но выбирается 2 точки скрещивания. Родители обмениваются всеми четными или нечетными частями.
- k-точечное скрещивание когда выбирается k точек



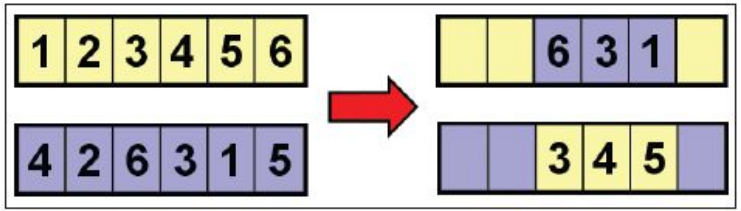
Методы скрещивания - Равномерное скрещивание

- Для каждого гена с равным шансом выбирается значение первого или второго родителя для первого потомка. Для второго потомка выбирается ген того родителя, который не попал в первого потомка.

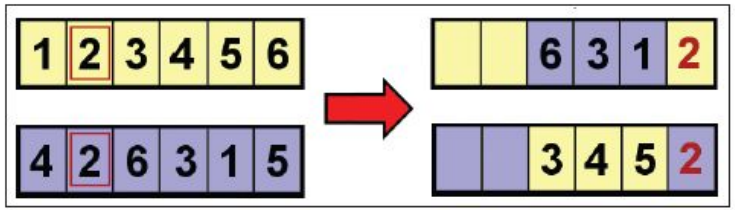


Методы скрещивания - Упорядоченное скрещивание

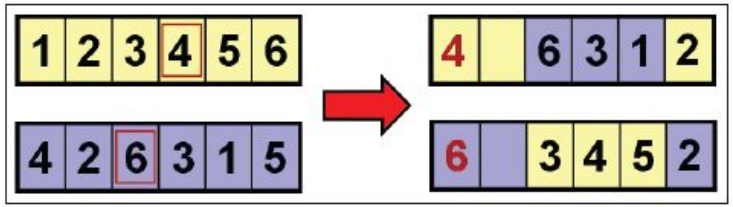
- Упорядоченного скрещивания - пытается сохранять относительно порядок элементов. Сначала происходит двухточечное скрещивание, а затем оставшиеся значения генов записываются на новые места с сохранением порядка как в родительской хромосоме.



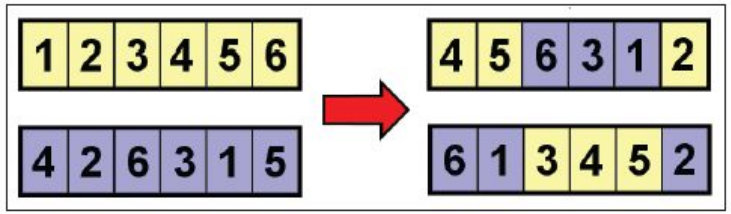
Пример упорядоченного скрещивания – шаг 1



Пример упорядоченного скрещивания – шаг 2



Пример упорядоченного скрещивания – шаг 3

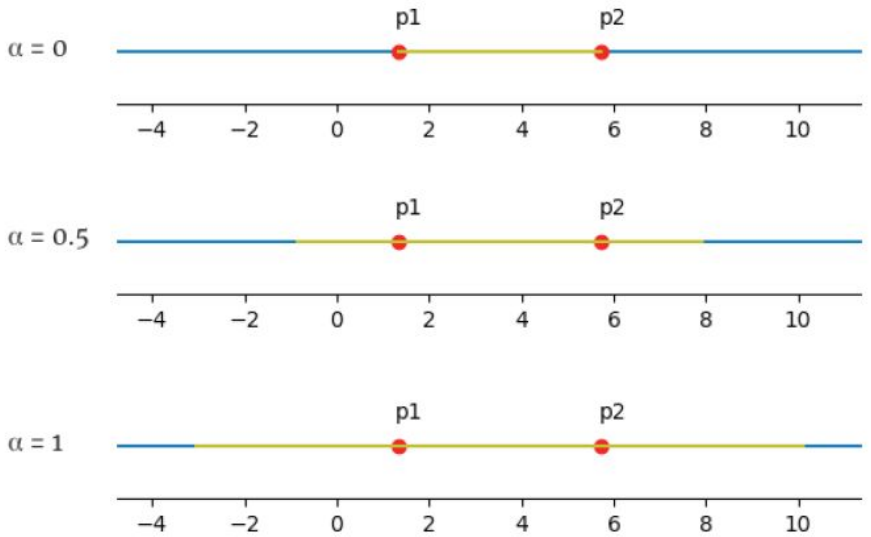


Пример упорядоченного скрещивания – шаг 4

Методы скрещивания - Скрещивание смешением

- Скрещивание смешением - метод для вещественных чисел, где ген каждого потомка генерируется из диапазона задаваемого родителем

$$[parent_1 - \alpha(parent_2 - parent_1), parent_2 + \alpha(parent_2 - parent_1)]$$



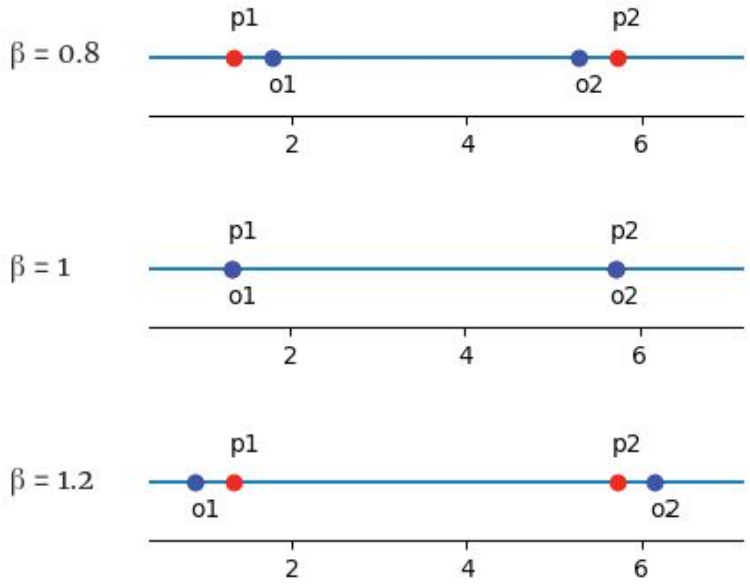
Методы скрещивания - Имитация двоичного скрещивания

- Имитация двоичного скрещивания. Симуляция одноточечного скрещивания, где средние значения генов потомков такие же как и у родителей.

$$offsping_1 = ½\ [(1 + \beta]parent_1 + (1 - \beta)parent_2]);$$
$$offsping_2 = ½\ [(1 - \beta]parent_1 + (1 + \beta)parent_2]),$$

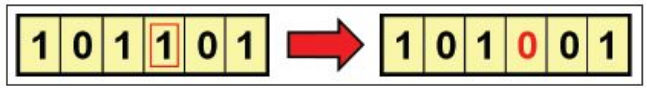
- Для выбора β генерируется случайное число u из [0, 1]

- если $u \leqslant 0.5$: $\beta = (2u)^{\frac{1}{\eta+1}}$,
- иначе: $\beta = \left[\frac{1}{2}(1-u)\right]^{\frac{1}{\eta+1}}$

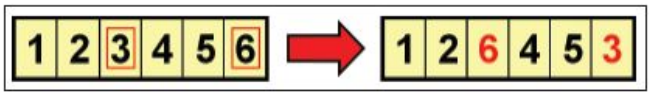


Мутация

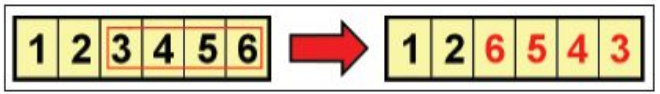
- Применяется к потомку после скрещивания с вероятностью p_m.
- Инвертирование бита - для бин. кодировки. каждый ген изменяется с вер. p_{inv}
- Мутация обменом - случайным образом меняются значения 2-х генов
- Мутация обращением - инвертируется порядок генов случайно выбранного участка
- Мутация перетасовкой - гены в случайно выбранном участке меняются случайным образом между собой
- Вещественная мутация - для вещ. чисел изменение на случайную величину.



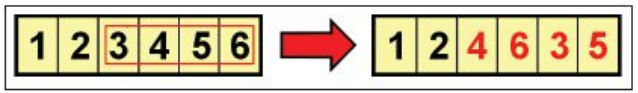
Пример мутации инвертированием бита



Пример мутации обменом



Пример мутации обращением



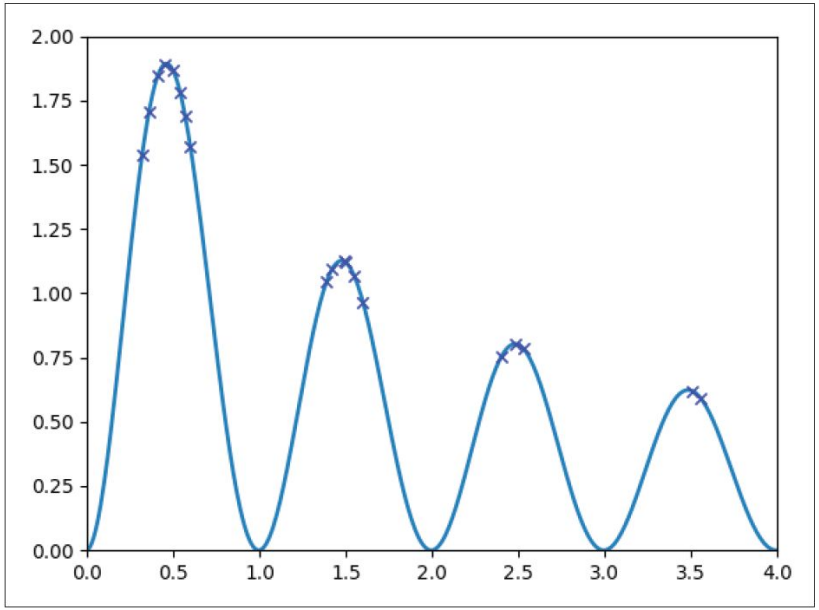
Пример мутации перетасовкой

Жесткие и мягкие ограничения

- На решение могут быть поставлены мягкие и жесткие ограничения
- Мягкие ограничения могут нарушаться, но нужно минимизировать их кол-во. Для этого в целевую функцию добавляется штраф
- Жесткие ограничения не могут нарушаться. Чтобы их контролировать:
  - Использовать кодировку, не допускающую нарушений. *Не всегда можно.*
  - Отбрасывать нарушающие решения. *Потеря информации.*
  - Исправление нарушений решений. *Не всегда можно*
  - Накладывать очень большой штраф в целевой функции.

Образование ниш

- Генетический алгоритм всегда пытается найти один глобальный максимум, но бывают задачи, когда нужно найти все глобальные максимум или все локальные максимумы тоже.
- Последовательное образование ниш - Ищем 1 максимум, далее накладываем штраф на найденный максимум и повторяем ГА
- Параллельное образование ниш- В ходе алгоритма накладывать штраф на решения лежащие кучно. Таким образом, индивидуумы будут искать “незанятые” максимумы



Результаты идеального генетического алгоритма с образованием ниш