Отчет

1. Описание проблемы

Необходимо решить задачу вместимости 0/1-рюкзака. Изначально дана совокупность предметов. Каждый предмет описан весом и ценностью . Имеется рюкзак с максимальной грузоподъемностью W. Нужно выбрать подмножество предметов так, чтобы суммарная ценность взятых предметов был максимально возможной и суммарный вес не превышал W.

Реализовать решение задачи при помощи генетического алгоритма.

1. Ввод

Генетический алгоритм приближенно решает задачу оптимизации посредством эволюционных операций над популяцией бинарных хромосом, где каждая хромосома соответствует выбору предметов.

Основные шаги выполнения генетического алгоритма:

1. Инициализация, то есть создание случайной популяции битовых векторов
2. Оценка приспособленности, то есть для каждой хромосомы вычисляется фитнес – суммарная ценность при укладке в рюкзак или 0, если вес превышается
3. Селекция, турнирно выбираются родительские хромосомы для получения потомков
4. Мутация, то есть вносятся небольшие изменения (флип битов) для сохранения разнообразия
5. Замена, потомки формируют новую популяцию, что повторяется до сходимости.
6. Имплементация
   1. Входные данные

Дан входной файл data\_knapsack01.txt. В первой строке указан максимальный вес рюкзака, начиная со второй строки и далее в каждой строке приведен один предмет, описанный весом и ценностью.

* 1. Структура программы
     1. Функция load\_data

Функция load\_data парсит этот файл data\_knapsack01.txt, извлекая W (максимальный вес вместимости) и массивы весов и ценностей.

* + 1. Функция fitness

Фитнесс-функция fitness вычисляет ценность, если вес больше W или равен ему, иначе 0

* + 1. Функция tournament\_selection

Турнирный отбор родителей, в котором выбираем k случайных индексов и возвращаем лучшего, на каждом месте выбирается наибольшая и идет дальше,  
в результате получаем набор родителей того же размера, что и исходная популяция.

3.3.3. Функции one\_point\_crossover, two\_point\_crossover

Два оператора скрещивания, первый – одноточечное скрещивание, вторая функция – двуточечное скрещивание

* + 1. Функция mutate

Оператор мутации (битовый флип).

* + 1. Функция run\_ga

Функция основного цикла генетического алгоритма, в котором присходят следующие шаги: инициализация популяции, за тем оценка приспособлености всех особей, селекция родителей, скрещивание и мутация. Функция возвращает лучшую хромосому (best\_chrom), ее фитнес (best\_fit) и историю сходимости (history).

* + 1. Главная функция main

Функция загружает данные из файла (максимальную вместимость, весы и ценности предметов), запускает функцию генетического алгоритма, подсчитывает итоговый вес и выводит фитнес, весы и хромосомы, так же декодирует и выводит список выбранных предметов для понимания какие предметы вошли в рюкзак.

* 1. Критерии оптимальности

Критерием оптимальности является максимизация суммарной ценности выбранных предметов при строгом соблюдении ограничения по весу.

Используется фитнес функция, которая вычисляет суммарный вес решения и если total\_w <= w\_maх, тогда при превышении функция вернет 0.

Таким образом, допустимые решения получают положительную оценку по своей ценности, а недопустимые автоматически отбрасываются.

* 1. Способ реализации операторов мутации и скрещивания

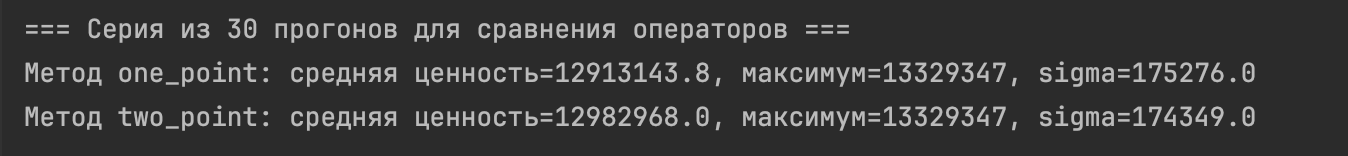
В реализации операторов скрещивания используем два оператора для сравнения и дальнейшего выбора наилучшего.

При одноточечном скрещивании выбирается случайная точка разрыва, и потомки формируются как сочетание левой части одного родителя и правой части другого. Этот метод сохраняет непрерывные блоки бит (состав предметов), что часто полезно для задачи рюкзака, где однородные группы предметов могут давать высокую совокупную ценность.

При двуточечном скрещивании выбираются две точки разрыва, и средний сегмент родителей обменивается. Двуточечный кроссовер более агрессивно перемешивает гены, что может способствовать лучшему исследованию пространства решений, но одновременно разрушает крупные структуры (блоки предметов), которые уже хорошо работают вместе.

Была проведена экспериментальная оценка:

При прочих равных параметрах (размер популяции 100, 200 поколений, вероятность скрещивания 0.8 и мутации 0.02) мы получили:



По итогам эксперимента:

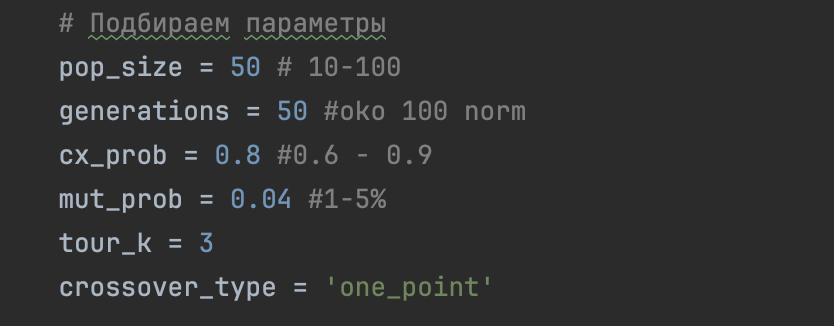
Двуточечный кроссовер даёт более высокую среднюю ценность (≈ 12 982 968) по сравнению с одноточечным (≈ 12 913 143), при этом оба метода достигают одинакового максимума. Одноточечный кроссовер сходится быстрее и демонстрирует большую стабильность в ранних поколениях.

**Выбранный метод в данном проекте:**  
В данной реализации окончательно **выбрано одноточечное скрещивание**, так как оно обеспечивает более быструю сходимость, стабильность при сохранении удачных комбинаций и меньшую деструкцию эффективных блоков предметов, что критично для последовательного приближения оптимального решения.

* 1. Стратегия отбора особей для скрещивания

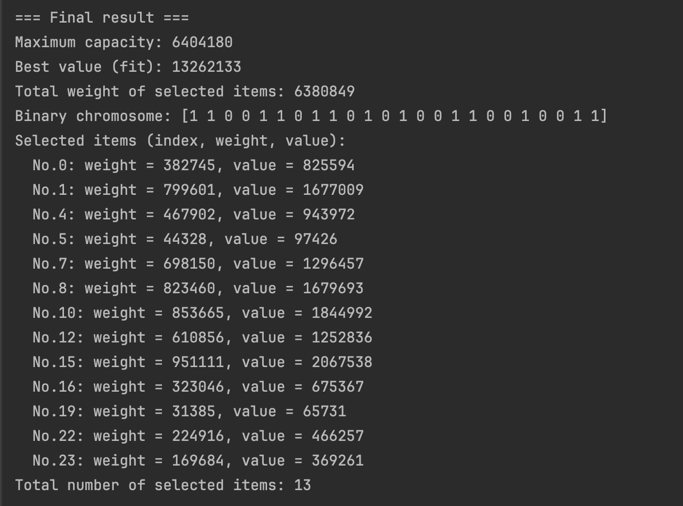
Для отбора родителей используется турнирная селекция (k-турнир) с параметром k=3, то есть случайно выбираются k особей из текущей популяции, среди них выбирается единственный победитель — особь с наибольшим значением фитнес-функции. Процесс повторяется до формирования нового пула родителей того же размера, что и исходная популяция.

* 1. Выбор параметров алгоритма



* 1. Результаты работы алгоритма

Пример вывода результатов:



Алгоритм работает четко, общий вес выбранных предметов не превышает вместимость рюкзака (6 404 180), а суммарная ценность будет близка к теоретическому максимуму. Алгоритм продемонстрировал устойчивую сходимость и высокое качество решения.

1. Заключение

Представленный генетический алгоритм решает задачу рюкзака-0/1. Одноточечный кроссовер выбран за счет быстрой сходимости и сохранения эффективных блоков. Турнирный обор (k=3) балансирует жестокий отбор и разнообразие. Размер популяции 50 и число поколений до 200 — сбалансированные значения, дающие приемлемую скорость работы и высокое качество результата.