

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Влияние выбора генов на результат работы

генетических алгоритмов в «задачи о рюкзаке»

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студентки 2 курса 247 группы

направление 09.04.03 — Прикладная информатика

механико-математического факультета

Беляев Владимир Евгеньевич

Научный руководитель

зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Зав. кафедрой

зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2025

Введение. «Задача о рюкзаке» — это классическая задача комбинаторной оптимизации, которая заключается в нахождении оптимального способа упаковки предметов в рюкзак. Эта задача имеет множество применений в различных областях, таких как логистика, планирование производства и другие.

Генетические алгоритмы — это алгоритмы, основанные на принципах естественного отбора и эволюции. Они используются для решения задач оптимизации, включая задачу о рюкзаке.

Сравнение решений задачи о рюкзаке генетическими алгоритмами с разными наборами генов позволяет оценить эффективность и точность алгоритма. В зависимости от выбранного набора генов, алгоритм может давать разные результаты. Например, если набор генов содержит только веса предметов, то алгоритм будет искать оптимальное распределение весов в рюкзаке. Если же набор генов содержит также размеры предметов, то алгоритм будет учитывать и размеры при поиске оптимального решения.

Сравнение решений задачи о рюкзаке генетическими алгоритмами с разными наборами генов позволяет выбрать наиболее эффективный и точный алгоритм для конкретной задачи. Это может быть полезно при разработке программного обеспечения для оптимизации процессов в различных областях.

Цель магистерской работы состоит в исследовании эффективности различных методов кодирования решений в генетических алгоритмах для задачи о рюкзаке (Knapsack Problem) и определение оптимальных подходов в зависимости от параметров задачи.

Задача о рюкзаке — одна из фундаментальных проблем комбинаторной оптимизации, имеющая богатую историю, уходящую корнями в XIX век. Её изучение связано с развитием математики, экономики и криптографии.

В 1897 первая формальная постановка (Тобиас Данциг). Хотя подобные задачи рассматривались и ранее, первую чёткую математическую формулировку дал Тобиас Данциг (отец Джорджа Данцига, создателя симплекс-метода). В работе «Задача о наилучшем выборе» он описал задачу упаковки предметов в ограниченный контейнер.

В 1898 работа Джорджа Мэтьюса. Британский математик Джордж Мэтьюс исследовал задачу упаковки в контексте оптимизации грузов в логистике. Он предложил первые эвристические методы решения.

1930 год связь с экономикой и теорией полезности. Задача стала применяться в экономике для моделирования потребительского выбора: «Как выбрать набор товаров с максимальной полезностью при ограниченном бюджете?» Экономисты Вильфредо Парето и Джон Хикс использовали аналогичные модели.

В 1940–1950 военные приложения и динамическое программирование. Во время Второй мировой войны задача использовалась для: оптимизации загрузки транспортных самолётов, распределения ресурсов в условиях дефицита. В 1957 году Ричард Беллман (основатель динамического программирования) применил свой метод к задаче о рюкзаке.

Псевдокод генетического алгоритма.

1. Инициализация: создать начальную популяцию P из N случайных особей.
2. Оценка: вычислить *fitness* для каждой особи.
3. Пока не выполнено условие остановки
(макс. поколений / достаточный *fitness*):
 - 3.1. Селекция: выбрать родителей из P .
 - 3.2. Скрещивание: создать потомков.
 - 3.3. Мутация: применить мутацию к потомкам.
 - 3.4. Оценка потомков.
 - 3.5. Замена: сформировать новую популяцию P' .
 - 3.6. $P = P'$.
4. Вернуть лучшую особь.

Представление: Гены представляют не отдельные предметы, а группы или паттерны.

Пример. Если предметы можно объединить в категории (еда, инструменты), то ген $[1, 0]$ означает взятие всех предметов из первой категории и отказ от второй.

Плюсы:

- Уменьшает размерность задачи.
- Учитывает зависимости между предметами.

Минусы:

- Теряет гибкость в выборе отдельных предметов.

Таблица 1 — Сравнительная таблица представления генов в генетическом алгоритме

Тип кодирования	Допустимость решений	Гибкость	Сложность реализации	Эффективность
Бинарное	Может нарушать ограничения	Высокая	Низкая	Средняя
Порядковое	Всегда допустимо	Низкая	Средняя	Высокая (для жадных стратегий)
Вещественное	Требует преобразования	Средняя	Высокая	Зависит от задачи
Групповое	Зависит от структуры	Низкая	Средняя	Высокая (для специфичных случаев)

Согласно таблицы 1 можно сделать следующие выводы: Выбор представления генов в генетическом алгоритме для задачи о рюкзаке зависит от:

- Размера задачи — бинарное кодирование хуже масштабируется.
- Ограничений — если важно избегать недопустимых решений, подходит порядковый метод.
- Структуры данных — групповое кодирование эффективно при наличии категорий.

На практике часто комбинируют подходы (например, бинарное кодирование + штрафы за перевес) или используют гибридные алгоритмы.

Программа использует генетический алгоритм для поиска оптимального набора предметов, максимизирующего ценность без превышения максимального веса. Алгоритм включает селекцию, скрещивание и мутацию, имитируя естественный отбор.

Различают следующие виды кроссоверов для задачи о рюкзаке:

1. Одноточечный и двухточечный - простые, но могут создавать недопустимые решения
2. Равномерный - дает больше разнообразия, но также может создавать недопустимые решения
3. Weight-Aware - учитывает ограничение по весу при создании потомка
4. Repairing - сначала создает потомка, затем исправляет его до допустимого решения
5. Mask-Based - использует битовую маску для выбора генов

Для работы жадных алгоритмов применяется бинарное кодирование для генов. Хотя для работы с жадными алгоритмами более удобно порядковое кодирование. Это сделано специально для большой совместимости реализации ГА и жадных алгоритмов. Также реализация операций кроссовера фактически соответствуют такой, как работают жадные алгоритмы.

Используя бинарное представление родителей организуем их скрещивание за счет операции «|» побитового «или». В результате все предметы используемые родителями попадут в предметы потомка. Затем используя полученные гены заполняем рюкзак. Не используемые гены отбраковываются, т.е. становятся равными нулю.

Жадный алгоритм по максимальной ценности. Этот метод реализует жадный кроссовер на основе максимальной ценности для генетического алгоритма, решающего задачу о рюкзаке. Он комбинирует гены родителей, выбирая предметы с наибольшей ценностью, и возвращает допустимое решение (не превышающее максимальный вес рюкзака).

Логика работы:

1. Формирование кандидатов:
 - Создается список `child`, включающий индексы предметов, которые есть хотя бы у одного родителя (`parent1[i] | parent2[i]`).
 - Для каждого предмета сохраняется его индекс и отрицательная ценность (для сортировки по убыванию).
2. Сортировка по ценности:
 - Предметы сортируются по убыванию ценности (используется отрицательное значение для удобства).

Особенности:

- Жадная стратегия: Всегда выбирает локально оптимальный вариант (максимальная ценность), что может не гарантировать глобальный оптимум, но работает быстро.
- Допустимость решения: Вес рюкзака никогда не превысит `max_weight`.
- Роль в ГА: Такой кроссовер ускоряет сходимость алгоритма, так как потомки наследуют «лучшие» гены родителей.

Ограничения:

- Не всегда оптимально: Жадный выбор может пропустить комбинацию с меньшей ценностью, но большей суммарной полезностью.
- Зависит от данных: Эффективность снижается, если предметы с высокой ценностью имеют большой вес.

Этот метод особенно полезен в гибридных ГА для быстрого получения качественных решений.

Жадный алгоритм по минимальному весу. Этот метод реализует жадный кроссовер на основе минимального веса для генетического алгоритма, решающего задачу о рюкзаке. Он комбинирует гены родителей, выбирая предметы с наименьшим весом, чтобы максимизировать количество предметов в рюкзаке, не превышая ограничение по весу.

Логика работы:

1. Формирование кандидатов:

- Создается список `child`, включающий индексы предметов, которые есть хотя бы у одного родителя (`parent1[i] | parent2[i]`).
- Для каждого предмета сохраняется его индекс и вес (для сортировки по возрастанию).

2. Сортировка по весу:

- Предметы сортируются по возрастанию веса (чтобы сначала выбирать самые легкие).

Особенности:

- Жадная стратегия: Выбирает предметы с наименьшим весом, чтобы уместить как можно больше предметов в рюкзак. Это полезно, когда важно количество предметов, а не их суммарная ценность.
- Допустимость решения: Вес рюкзака никогда не превысит `max_weight`.

- Роль в ГА: Такой кроссовер помогает исследовать решения с большим количеством предметов, что может быть полезно для задач, где важна «комплектность».

Ограничения:

- Не максимизирует ценность: Алгоритм может выбрать много легких, но малоценных предметов, пропуская более ценные, но тяжелые варианты.
- Зависит от данных: Эффективность снижается, если легкие предметы имеют низкую ценность.

Этот метод полезен в гибридных ГА для задач, где важно количество предметов (например, упаковка мелких товаров).

Этот метод реализует жадный кроссовер на основе удельной ценности (ценность/вес) для генетического алгоритма, решающего задачу о рюкзаке. Он комбинирует гены родителей, выбирая предметы с наибольшей удельной ценностью, чтобы максимизировать суммарную ценность при ограниченном весе.

Логика работы:

1. Формирование кандидатов:

- Создается список `child`, включающий индексы предметов, которые есть хотя бы у одного родителя (`parent1[i] | parent2[i]`).
- Для каждого предмета сохраняется его индекс и отрицательная удельная ценность (для сортировки по убыванию).

2. Сортировка по удельной ценности:

- Предметы сортируются по убыванию удельной ценности (т.е. сначала самые «выгодные» — с высоким отношением ценность/вес).

Особенности:

- Оптимальность для дробного рюкзака: Для «дробной» задачи о рюкзаке (где предметы можно делить) этот алгоритм гарантирует точное решение. Для «0-1» задачи (предметы нельзя делить) он дает приближенное, но часто близкое к оптимальному.
- Баланс ценности и веса: Учитывает как ценность, так и вес, в отличие от `_greedy_by_value` (только ценность) или `_greedy_by_weight` (только вес).

- Эффективность в ГА: Позволяет быстро находить высококачественные решения, что ускоряет сходимость генетического алгоритма.

Ограничения:

- Не всегда глобальный оптимум: Для «0-1» задачи жадный алгоритм может пропустить оптимальную комбинацию (например, если два менее выгодных предмета вместе дают большую ценность, чем один выгодный).
- Зависит от данных: Эффективность снижается, если предметы с высокой удельной ценностью имеют большой вес (например, ценность=100, вес=99).

Этот метод лучший выбор для большинства задач, где важно и количество, и ценность.

Для сравнения результатов работы ГА с разными видами кроссовера был выбраны современные результаты полученные применением алгоритм летучих мышей.

На основе алгоритмов разработанных в данной работе были произведены вычисления используя разные виды кроссоверов:

- GAs – одноточечный;
- GAt – двухточечный;
- GAu – равномерный;
- GAw – учитывает ограничение по весу при создании потомка;
- GAr – сначала создает потомка, затем исправляет его до допустимого решения;
- GAm – используется битовая маска, которая определяет; от какого родителя брать каждый ген;
- GAv – скрещивание жадным алгоритмом по максимальной ценности
- GAw – скрещивание жадным алгоритмом по возрастанию веса;
- GAd – скрещивание жадным алгоритмом по убыванию удельной ценности.

Всего было произведено по каждому, вышеперечисленному алгоритму по 200 вычислений.

Вторая колонка «мак.» содержит максимальное значение полученное для данного алгоритма для всех 200 вычислений.

Третья колонка «мин.» содержит минимальное значение полученное для данного алгоритма для всех 200 вычислений.

Четвертая колонка «сред.» содержит среднее значение полученное для данного алгоритма для всех 200 вычислений.

Последняя колонка «реш.» содержит самое оптимальное значение для рассматриваемого рюкзака.

Таблица 3.6, пример k5 (100 предметов, $\text{max_weight} = 3818$, оптимальное решение = 15170):

- Р-АМВА достиг максимума 14904, что близко к оптимуму.
- ГА:
 - GAgd (жадный по удельной ценности) снова оказался лучшим, достигнув оптимума (15170) и имея высокое среднее (15018.9).
 - GAgv (жадный по ценности) также показал отличные результаты (максимум 14937, среднее 14764.9).
 - Классические ГА (GAs, GAt, GAu) уступили жадным методам (максимум около 14200–14500, среднее 13000–14000).
 - GAw остался неэффективным (среднее 8799.3).

Выводы по результатам вычислений:

1. Жадные алгоритмы в кроссовере:

- GAgd (жадный по удельной ценности) показал наилучшие результаты для задач средней и большой размерности (k3–k5), часто достигая оптимума или близких к нему значений.
- GAgv (жадный по ценности) также был эффективен, особенно для задач с высокой ценностью отдельных предметов.
- GAgw (жадный по весу) оказался слабым, так как его стратегия не учитывает ценность предметов.

2. Классические ГА:

- Одноточечный (GAs), двухточечный (GAt) и равномерный (GAu) кроссоверы показали схожую эффективность, но уступили жадным методам для больших задач.
- Маскированный кроссовер (GAm) и ремонтирующий (GAr) дали хорошие результаты, но не превзошли жадные алгоритмы.

3. Алгоритмы летучих мышей:

- Р-АМВА оказался самым эффективным среди ВВА и его модификаций, но для больших задач (k_4 – k_5) уступил жадным ГА.

4. Неэффективные методы:

- GAw (учет веса) показал низкую эффективность во всех тестах, так как не учитывал ценность предметов.

Таким образом, гибридные генетические алгоритмы с жадными кроссоверами (особенно по удельной ценности) демонстрируют высокую эффективность для решения задачи о рюкзаке, особенно для больших и сложных примеров. Они сочетают преимущества генетического поиска (разнообразие решений) и жадных стратегий (быстрое нахождение качественных решений).

Операторы кроссовера играют ключевую роль в генетических алгоритмах, определяя, как комбинируются генетические материалы родителей для создания потомков. В исследовании сравнивались различные типы кроссоверов, включая классические (одноточечный, двухточечный, равномерный) и гибридные (жадные алгоритмы). Результаты показали, что:

- Классические кроссоверы (GAs, GAt, GAu) обеспечивают разнообразие решений, но могут генерировать недопустимые варианты (превышение веса рюкзака), что требует дополнительной коррекции.
- Гибридные кроссоверы (GAgv, GAgw, GAgd) демонстрируют более высокую эффективность, особенно GAgd (жадный по удельной ценности), который часто достигает оптимальных решений. Эти операторы учитывают ограничения задачи, минимизируя генерацию недопустимых решений.
- Специализированные кроссоверы (GAw, GAr, GAM) показали смешанные результаты. Например, GAw (учет веса) оказался неэффективным, в то время как GAr (ремонтирующий) и GAM (маскированный) показали хорошие результаты, но уступили жадным методам.

Заключение. В данной магистерской работе проведено исследование эффективности генетических алгоритмов (ГА) для решения задачи о рюкзаке (0/1 Knapsack Problem) с различными методами кодирования решений и операторами. Основное внимание уделено сравнению классических подходов с гибридными методами, включающими жадные алгоритмы в операторах кроссовера. Результаты работы демонстрируют значительное влияние выбо-

ра стратегии кодирования и операторов на точность и скорость сходимости алгоритмов.

Проведенное исследование подтвердило, что генетические алгоритмы, дополненные жадными стратегиями, являются мощным инструментом для решения задачи о рюкзаке. Оптимальный выбор методов кодирования и операторов позволяет достигать высокой точности и устойчивости даже для NP-трудных задач. Результаты работы открывают перспективы для дальнейшего совершенствования алгоритмов и их применения в реальных сценариях оптимизации.