

## **ИЗУЧЕНИЕ РАЗЛИЧНЫХ ПОСТАНОВОК ЗАДАЧИ О РЮКЗАКЕ И МЕТОДОВ ИХ РЕШЕНИЯ**

**Додонова М.М.,**

**научный руководитель: канд. физ.-мат. наук, доцент Баранова И.В.**

*Сибирский Федеральный Университет,*

*Институт математики и фундаментальной информатики*

### **Введение**

Практика порождает все новые и новые задачи оптимизации, причем их сложность растет. Реальные прикладные задачи дискретной оптимизации очень сложны. Современные методы оптимизации далеко не всегда справляются с решением реальных задач без помощи человека. Требуется новые математические модели и методы, которые позволяют учитывать наличие многих критериев и проводят глобальный поиск оптимума. Следует отдавать предпочтение таким методам, которыми проще управлять в процессе решения задачи.

Целью данной работы является изучение различных вариантов постановки задачи о рюкзаке, реализация основных методов решения данных задач, классификация и сравнение этих методов.

**Задача о ранце (рюкзаке)** — одна из NP-полных задач комбинаторной оптимизации. Своё название задача получила от оптимизационной задачи укладки как можно большего числа ценных вещей в рюкзак при условии, что общий объём (или вес) всех предметов, способных поместиться в рюкзак, ограничен. Задача о рюкзаке является актуальной и достаточно востребованной с точки зрения ее приложения в реальной жизни. Задача о загрузке (о рюкзаке) и её модификации часто возникают в экономике, прикладной математике, криптографии, генетике и логистике для нахождения оптимальной загрузки транспорта (самолёта, поезда, трюма корабля) или склада.

В работе рассматриваются четыре наиболее популярных алгоритма решения данной задачи: полный перебор, жадный и генетический алгоритмы, а также метод ветвей и границ (как сокращение метода полного перебора). Все методы разделены на две группы. Первую группу образуют точные методы, к которым относятся полный перебор и метод ветвей и границ. Вторая группа – приближенные методы, она представлена жадным и генетическим алгоритмом.

### **Классическая постановка задачи о рюкзаке**

Пусть имеется набор предметов, каждый из которых имеет два параметра – вес и ценность. Имеется рюкзак с некоторым заданным значением вместимости. Задача заключается в том, чтобы собрать рюкзак с максимальной ценностью предметов внутри, соблюдая при этом весовое ограничение рюкзака.

Математически постановка задачи формулируется следующим образом: пусть имеется  $n$  предметов. Для каждого  $i$ -го предмета задан его вес  $p_i > 0$  и стоимость (ценность)  $c_i > 0$ ,  $i = 1, 2, \dots, n$ . Задано ограничение на максимальный вес рюкзака –  $P$ . Каждый  $x_i$  может принимать только одно из двух значений:  $x_i = 1$ , если  $i$ -й предмет упаковывают в рюкзак, или  $x_i = 0$ , в противном случае.

Требуется выбрать из заданного множества предметов набор с максимальной суммарной стоимостью  $\sum_{i=1}^n c_i x_i$  при одновременном соблюдении ограничения на суммарный вес найденного набора  $\sum_{i=1}^n p_i x_i \leq P$ .

### Классификация и сравнение методов решения задачи о рюкзаке

Существует несколько разновидностей задачи о ранце, отличия между которыми заключаются в условиях, наложенных на рюкзак, предметы или их выбор:

1. *Рюкзак 0-1* или *Классическая задача* – каждый предмет можно брать только один раз;
2. *Ограниченный рюкзак* – каждый предмет можно брать определенное количество раз;
3. *Неограниченный рюкзак* – каждый предмет можно брать сколько угодно раз;
4. *Непрерывный рюкзак* – возможно брать любую дробную часть от предмета;
5. *Рюкзак с мультивыбором* – есть несколько классов предметов, из которых можно брать одного представителя, причём некоторые вещи имеют больший приоритет, чем другие;
6. *Мультипликативный рюкзак* – есть несколько рюкзаков с различной вместимостью;
7. *Многомерный рюкзак* – есть более одного ограничения на рюкзак.

Существует ряд методов, позволяющих находить решение задачи о ранце. В данной работе рассматривались следующие методы (для всех семи приведенных выше постановок задачи):

1. Полный перебор,
2. Метод ветвей и границ,
3. Жадный алгоритм,
4. Генетический алгоритм.

Задача о рюкзаке является NP-полной, т.е. для нее не существует полиномиального алгоритма, решающего её за разумное время. Если перебирать всевозможные подмножества данного набора из  $n$  предметов, то получится решение сложности не менее чем  $O(2^n)$ . В настоящее время неизвестен (и, скорее всего, вообще не существует) алгоритм решения этой задачи, сложность которого является многочленом от  $n$ .

Таблица 1

Предметы для практической задачи рюкзака

	предмет	вес	стоимость		предмет	Вес	стоимость
1	рыба	6	5	11	колбаса	5	9
2	молоко	4	3	12	огурцы	6	9
3	сок	3	1	13	клей	2	5
4	хлеб	2	3	14	вата	4	5
5	телефон	5	6	15	тетрадь	5	10
6	мясо	7	9	16	кефир	7	8
7	вода	4	8	17	кофе	4	7
8	бумага	2	6	18	курица	9	10
9	яблоки	9	10	19	масло	5	7
10	сыр	3	7	20	шоколад	3	6

В работе рассматривался практический пример задачи о рюкзаке с максимальной вместимостью рюкзака  $P = 30$  и количеством предметов  $n = 20$ . С помощью перечисленных методов были найдены решения всех семи вариантов задачи о рюкзаке, а также выполнено сравнение методов по скорости и точности нахождения решения. Проведенное сравнение показано в таблице 1.

Таблица 2

Сравнительный анализ методов

Метод	Тип алгоритма	Сложность	Плюсы	Минусы
Полный перебор	точный	$O(N!)$	простота реализации; точное решение	входные данные не велики; временная сложность
Метод ветвей и границ	точный		возможно значительное сокращение времени; простота реализации	работает как полный перебор
Жадный алгоритм	приближенный	$O(N \cdot \log(N))$	высокая скорость; может работать с большими значениями $N$ ; простота реализации	решение неточное
Генетический алгоритм	приближенный		высокая скорость; может работать с большими значениями $N$ ; независимость от вида исходных данных	не гарантирует нахождение оптимального решения
Метод динамического программирования	точный	$O(W \cdot N)$	независимость от вида исходных данных; точное решение	большой объем вычислительной работы

При выборе алгоритма решения приходится выбирать между точными алгоритмами, которые не применимы для рюкзаков большой размерности, и приближенными, которые работают быстро, но не обеспечивают оптимального решения задачи. Выбор использования того или иного метода является спорным вопросом. Всё зависит от постановки задачи, а также от того, какие цели поставлены. Если требуется найти точное решение, то необходимо использовать точные методы. В

ситуации небольшого набора входных данных (до 10-20 предметов), лучше применять методы перебора или ветвей и границ, в силу простоты реализации. Если же точность решения не играет большой роли, или входные данные таковы, что ни один из точных методов не работоспособен, то для решения задачи можно использовать лишь приближенные алгоритмы.

## **Выводы**

Поскольку метод полного перебора очень трудоемкий, то размер рюкзака и количество предметов при его использовании должны быть ограничены. Следовательно, этот метод неприменим для неограниченного и непрерывного рюкзаков. При решении перебором задач мультипликативного и ограниченного рюкзаков необходимо уменьшить количество предметов до 10, чтобы решение было найдено.

Очевидным недостатком алгоритма метода ветвей и границ при решении задач большой размерности является необходимость перебрать слишком большое количество вариантов перед тем, как будет найден оптимальный. Несомненно, метод ветвей повторяет перебор. Однако, благодаря отсечению "бесполезных" решений, может самостоятельно ограничить бесконечное пространство решений. Поэтому его можно применить для неограниченного рюкзака, но нельзя применить для непрерывного рюкзака.

Жадный алгоритм является однокритериальным алгоритмом, поэтому он неприменим для многомерного рюкзака. Однако следует отметить, что для задачи непрерывного рюкзака именно этот алгоритм находит наиболее оптимальное решение.

Генетический алгоритм является одним из самых быстрых алгоритмов. Но ограничением этого алгоритма является то, что его хромосомы кодируются дискретно. Соответственно, его нельзя использовать для решения непрерывных и бесконечных задач. Недостатком генетического алгоритма также является то, что он не гарантирует нахождение оптимального решения в некоторых ситуациях (находит локальный экстремум вместо глобального). Происходит это вследствие того, что алгоритм может заканчиваться не только при достижении оптимального решения, но и следующих условиях:

- пройдено максимальное заданное число итераций;
- прошло максимальное время, заданное для выполнения алгоритма;
- при переходе к новому поколению не происходит существенных изменений.