Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Фундаментальные науки
КАФЕДРА Прикладная математика

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Студент Сербин Георгий Эдуардович

фамилия, имя, отчество

Группа ФН2-41Б

Тип практики: Ознакомительная практика

Название предприятия: НУК ФН МГТУ им. Н.Э. Баумана

Студент _____ Сербин Г.Э.
подпись, дата *фамилия и.о.*

Руководитель практики _____ Савельева И.Ю.
подпись, дата *фамилия и.о.*

Оценка _____

2020 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Кафедра «Прикладная математика»

З А Д А Н И Е
на прохождение производственной практики

на предприятии НУК ФН МГТУ им. Н.Э. Баумана

Студент Сербин Георгий Эдуардович группа ФН2-41Б
(фамилия, имя, отчество; инициалы; индекс группы)

Во время прохождения производственной практики студент должен:

1. Изучить на практике основные возможности языка программирования C++ и систем компьютерной алгебры, закрепить знания и умения, полученные в курсах «Введение в информационные технологии», «Информационные технологии профессиональной деятельности».
2. Изучить способы реализации методов решения задачи о ранце.
3. Реализовать метод ветвей и границ и приближенную схему полностью полиноминального времени.

Дата выдачи задания «1» марта 2020 г.

Руководитель практики _____ Савельева И.Ю.
подпись, дата *фамилия и.о.*

Студент _____ Сербин Г.Э.
подпись, дата *фамилия и.о.*

Оглавление

1. Введение

Основной целью ознакомительной практики 4-го семестра, входящей в учебный план подготовки бакалавров по направлению 01.03.04 – Прикладная математика, является продолжение знакомства с особенностями осуществления деятельности в рамках выбранного направления подготовки и получение навыков применения теоретических знаний в практической деятельности. В первом модуле пройденного курса «информационные технологии профессиональной деятельности» был получен опыт оформления работ в системе компьютерной вёрстки TEX и его наборе расширений LATEX, что позволило закрепить и развить навыки, полученные ранее в курсе «введение в специальность». Этот опыт оказался особенно полезным при написании отчета по ознакомительной практике. Во втором модуле курса «информационные технологии профессиональной деятельности» были получены практические знания о работе в системе компьютерной алгебры MATLAB, что позволило нам использовать этот математический пакет прикладных программ для выполнения второй части ознакомительной практики 4-го семестра.

2. Формулировка задания

Задача о ранце. Имеется набор предметов, каждый из которых характеризуется двумя положительными параметрами — весом и ценностью. Требуется собрать в ранец такую совокупность предметов, чтобы их суммарная ценность была максимальной, при этом вместимость ранца ограничена. Каждый предмет разрешается брать не более одного раза.

Входные данные: количество предметов, вес и ценность каждого предмета, ограничение суммарного веса ранца.

Выходные данные: номера использованных предметов.

3. История рассматриваемой задачи

Задача о ранце (о рюкзаке) — NP[1]-полная задача комбинаторной оптимизации. Комбинаторная оптимизация — область теории оптимизации в прикладной математике, связаная с исследованием операций, теорией алгоритмов и теорией вычислительной сложности. Принадлежность классу NP означает, что для данной задачи не найден алгоритм, решающий её за полиномиальное время. К NP-полным относят

[1] NP-задачи — *от англ.* non-deterministic polynomial.

наиболее трудные из этого класа задачи, к которым можно свести любую другую задачу из класса NP за полиномиальное время. В 70—90-е годы XX века началось интенсивное изучение задачи о ранце как теоретиками, так и практиками. Связано это было с достаточно простой формулировкой и большим количеством свойств данной задачи. В 1972 году данная задача вошла в список NP-полных задач [1].

Существуют различные постановки задачи о ранце. Среди наиболее популярных условий выделяют:

1) берется не более одного экземпляра каждого предмета,
2) берется не более заданного числа экземпляров каждого предмета,
3) может быть взято любое число экземпляров каждого предмета.

В нашем случае каждый предмет берется только один раз, но при необходимости во входных данных можно указать в любом количестве предметы с одинаковой стоимостью и весом. При постановке задачи могут быть также заданы дополнительные условия: вес предметов может быть выражен целым или вещественным числом. В нашей работе предполагается, что вес предметов — вещественное число.

С различными вариациями задачи о рюкзаке можно столкнуться в экономике, прикладной математике, криптографии и логистике и во многих других областях.

4. Обзор методов решения

Существует большое множество алгоритмов, решающих поставленную задачу. Как было сказано выше, задача о ранце относится к классу задач, не имеющих полиномиального алгоритма, поэтому при решении задачи о ранце необходимо выбирать между точными алгоритмами, которые выдают оптимальное решение, но требуют зачастую при большом количестве входных данных неразумно много времени, и приближенными, которые могуть работать значительно быстрее, но не гарантируют оптимального решения задачи. Выделим основные алгоритмы, которые нам в той или иной степени пришлось изучить, при выполнении задания по практике.

К точным алгоритмам относят:

- **Метод перебора.** Наиболее простой и понятный с точки зрения реализации метод. Осуществляется прямой перебор всех возможных вариантов укладки ранца. Применяется для решения задач с малым количеством предметов.
- **Метод ветвей и границ.** Дополненный метод полного перебора, осуществляющий перебор возможных решений с отсечением некоторых бесперспективных вариантов. С помощью этого метода нами была решена задача о ранце. О нем пойдет речь в следующем разделе.

К приближенным алгоритмам относят:

- **Жадный алгоритм.** Один из наиболее быстрых и просто реализуемых методов, однако может получить решение сколь угодно далёкое от оптимального [2]. Жадный алгоритм использовался нами для оценки решений сверху или снизу в обоих реализуемых методах.
- **Приближенная схема полностью полиномиального времени.** Позволяет за полиномиальное время найти некоторое решение, отличающееся от оптимального на заданную точность ε. Это второй метод, которым нами была решена задача о ранце.

5. Метод ветвей и границ

Основой алгоритма, построенного по стратегии ветвей и границ является древовидный полный перебор (см. рис. 1) [3]. В каждом узле правая подветвь соответствует варианту «берем следующий предмет» (при условии, что он помещается в ранец), левая подветвь соответствует варианту «не берем следующий предмет». В этом заключен смысл «ветвей» в данном алгоритме. Каждая листовая вершина нашего дерева перебора будет соответствовать какому-то варианту наполнения рюкзака, и все такие варианты будут перебраны данной стратегией. Перебрав и сравнив все возможные варианты, найдем точное решение нашей задачи.

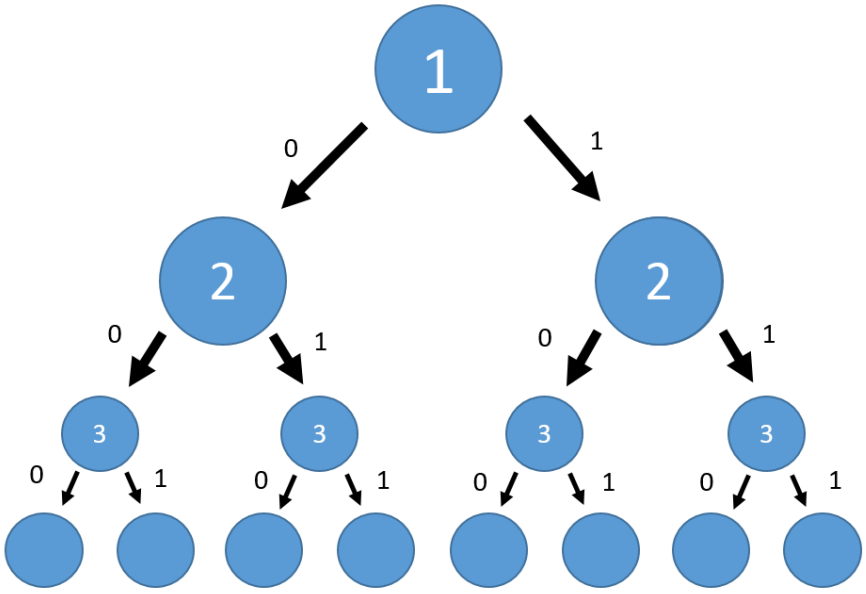

Рис. 1. Дерево полного перебора для 3-х предметов

Идея «границ» в стратегии ветвей и границ сводится к тому, что мы должны уметь относительно быстро заранее оценить наиболее оптимистичный вариант решения для подветви, не проводя полного анализа этой подветви. Если этот наиболее оптимистичный вариант окажется хуже уже имеющихся у нас на данный момент вариантов решения, то не имеет смысла просматривать эту подветвь. В нашем случае требуется получить оценку сверху для наилучшего потенциального решения, которое можно извлечь из подветви.

Отметим, что метод может использовать также оценку снизу. Оценка снизу используется для начального задания возможного оптимального решения. Еще до построения первых решений оценка снизу может помочь на первом этапе убрать бесперспективные ветви. Также имеет смысл отсекать из рассмотрения все подветви, для которых общий вес взятых вещей больше вместимости ранца.

В качестве вычисления верхней оценки стоимости решений в каждой подветви воспользуемся жадным алгоритмом. Для этого необходимо:

1) отсортировать предметы по удельной стоимости, т.е. по параметру $c = \frac{p}{w}$, где p — стоимость предмета, w — его вес;
2) класть в ранец вещи с наибольшей удельной стоимостью до тех пор пока предметы не закончатся или следующий по списку предмет не поместится;
3) той вещью, которая не поместилась (если такая есть) заполнить полностью рюкзак так, как будто вещи можно брать частично.

Алгоритм дает точное решение для дробной задачи о ранце, т.е. в задаче с условием, что предмет можно брать частично. В нашем случае полученное решение будет являться оценкой сверху для данной подветви. Действительно, если рюкзак оказывается полностью загружен вещами, удельный вес которых наибольший, то это будет лучшим решением. В нашей задаче, однако, это выполняется не всегда, поэтому реальная оптимальная общая стоимость взятых предметов может быть ниже.

Оценим сложности алгоритмов, рассмотренных в этом разделе. Пусть n — количество вещей. В случае классического полного перебора необходимо рассмотреть 2^n вариантов заполнения ранца, т. к. каждый предмет может быть либо взят, либо не взят. Следовательно, сложность полного перебора — $O(2^n)$. Сортировка вещей по удельной стоимости для жадного алгоритма выполняется за $O(n \log n)$, а непосредственно оценка за $O(n)$. Сортировку достаточно провести один раз до составления ветвей. Сложность жадного алгоритма — $O(n \log n)$ или $O(n)$, если учитывать, что сортировка была проведена заранее. Метод ветвей и границ по сути является методом полного перебора с тем отличием, что мы исключаем заведомо неоптимальные ветви дерева полного перебора. Способность метода ветвей и границ уменьшать ко-

личество вариантов перебора сильно опирается на входные данные. Если удельные стоимости вещей сильно разнятся, то есть шанс найти оптимальное решение при составлении первых ветвей и убрать из рассмотрения значительную часть вариантов. В общем случае, если никаких ветвей из рассмотрения не удастся убрать, придется просмотреть 2^n вариантов. Если принять, что ЭВМ способна обработать 10^8 вариантов в секунду, то на 20 вещей потребуется доля секунды, на 40 вещей — чуть больше 3-х часов, а на 80 вещей — порядка 30 миллионов лет. Ясно, что для решения практических задач с большим количеством случайных входных данных использование такого метода является неразумным.

6. Приближенная схема полностью полиномиального времени

Так как точного алгоритма решения задачи за полиномиальное время не было найдено, появилась задача получить за полиномиальное время решение с некоторой гарантированной точностью $\varepsilon > 0$. Для этого существует целый ряд приближённых схем полностью полиномиального времени [1], то есть со сложностью, являющейся полиномом от количества вещей n и точности $\frac{1}{\varepsilon}$.

Идея, стоящая за реализуемой нами схемой, заключается в снижении точности, с которой заданы ценности предметов [4]. Сначала рассмотрим случай, когда стоимости всех предметов выражены целыми числами.

Пусть i-ый предмет имеет стоимость p_i и вес w_i. Обозначим через S сумму всех предметов, а через P — максимальную среди этих предметов стоимость, т. е.

$$S = \sum_{i=1}^{n} p_i, \qquad P = \max_{i \in 1..n} p_i.$$

Пусть W — вместимость рюкзака. Через $w(k,\, p)$ обозначим минимальный вес, необходимый для того, чтобы уложить предметы с номерами, не превосходящими $k \in [0..n]$, общей стоимостью не менее $p \leq S$.

Составим таблицу размером $n \times S$, где на пересечении k-ой строки с p-ым столбцом поставлен $w(k, p)$. Заполнять её можно по столбцам:

$$w(0,\, 0) = 0; \qquad w(0,\, p) = \infty,\ p > 0;$$
$$w(k+1,\, p) = \min\{w(k,\, p), w(k,\, p - p_{k+1}) + w_{k+1}\}.$$

Под значением ∞ в ячейке $(k,\, p)$ понимается, что предметами с номерами $1..k$ нельзя

получить общую стоимость p ни при какой их укладке. Выражение для определения ячейки $w(k+1,\, p)$ означает, что минимальный вес будет равен или $w(k,\, p)$, если мы не берем $(k+1)$-ый предмет, или $(w(k,\, p-p_{k+1})+w_{k+1})$, в случае если мы берем $(k+1)$-ый предмет.

Таким образом, максимальное p, для которого верно $w(n,\, p) \leq W$, и будет максимально возможной стоимостью укладки предметов. Если при этой реализации еще сохранять номера предметов, реализующих данные $w(n,\, p)$, то будет получено решение. Алгоритм работает за $O(nS)$, или $O(n^2P)$, т. к. $S \leq nP$.

Теперь воспользуемся описанным алгоритмом для решения нашей задачи с вещественными стоимостями. При помощи ранее описанного жадного алгоритма (без последнего шага с частичным взятием предмета) получим нижнюю оценку P_{min} суммарной стоимости оптимального решения задачи. Итак, пусть задано число $\varepsilon > 0$, определяющее, с какой точностью мы хотим найти ответ. Обозначим

$$K_\varepsilon = \frac{P_{min}}{\left(1+\frac{1}{\varepsilon}\right) n}. \tag{1}$$

Поделим все имеющиеся стоимости p_i на K_ε и округлим результат:

$$p'_i = \left[\frac{p_i}{K_\varepsilon}\right]. \tag{2}$$

Далее, на новом наборе стоимостей p'_i реализуем алгоритм для целых стоимостей и получим некоторое решение. Осталось доказать, что полученный алгоритм дает необходимую нам точность.

Пусть A_ε — общая стоимость набора предметов, который вернул алгоритм. Заметим, что алгоритм для целых стоимостей находит точное решение, поэтому отклонение для вещественных стоимостей могло появиться только при округлении. При округлении могло потеряться не более 1 единицы на каждом из n предметов, эта потеря домножается на K_ε при обратном переходе от p'_i к p_i. Таким образом, $A_\varepsilon \geq A_{opt} - nK_\varepsilon$, где A_{opt} — оптимальная стоимость для данной задачи. Ясно также, что $A_{opt} \geq P_{min}$, т. е. оптимальная стоимость взятых предметов не меньше чем стоимость, уже полученная с помощью жадного алгоритма. Тогда

$$\frac{A_\varepsilon}{A_{opt}} \geq \frac{A_{opt} - nK_\varepsilon}{A_{opt}} = 1 - \frac{nP_{min}}{A_{opt} n \left(1+\frac{1}{\varepsilon}\right)} \geq 1 - \frac{1}{1+\frac{1}{\varepsilon}} = \frac{1}{1+\varepsilon} > 1-\varepsilon.$$

Таким образом, получена оценка

$$A_\varepsilon > (1-\varepsilon) A_{opt}.$$

Время работы алгоритма для предметов с вещественной стоимостью с учетом (1) и (2) не превосходит $n^2 \max_{i \in 1..n} p'_i \leq n^3 \left(1 + \frac{1}{\varepsilon}\right)$. Отметим, что при увеличении точности множитель K_ε уменьшается, а стоимости p_i увеличиваются, тем самым таблица $w(k, p)$ становится шире, что помимо увеличения времени работы, увеличивает и требуемую память. Таким образом, приближенная схема полностью полиномиального времени позволяет за время полинома от n и $\frac{1}{\varepsilon}$ получить решение, отличающееся от оптимального не более чем на заданную величину.

7. Реализация метода ветвей и границ

7.1. Особенности реализации на языке C++

Была реализована базовая проверка на правильность введенных данных: вес и стоимость предметов должны быть положительными числами. Предметы были реализованы структурой `struct Item`, содержащей три поля: `int num` — номер предмета, `T weight` — его вес, `T price` — его стоимость, где `T` — псевдоним типа (задан как `double`). Для реализации метода ветвей и границ использовалась рекурсивная функция `find_best()`. В каждом узле эта функция выбирает между двумя решениями, полученными с учетом того, взяли следующий предмет в подветви или нет. Для удобства исполнения алгоритма была создана структура `struct Solution`, хранящая номера предметов `std::list<int> item_numbers` некоторого решения и его суммарную стоимость `T total_price`. Перед нахождением решений массив предметов `std::vector<Item> items` сортировался по удельной стоимости. Это с одной стороны необходимо для выполнения оценки с помощью жадного алгоритма, а с другой стороны, при условии, что в рекурсии сначала рассматривается вариант «взять предмет», это позволяет быстрее найти оптимальное решение и, тем самым, убрать из рассмотрения больше вариантов. Функция `greedy_bound()` осуществляет оценку сверху с помощью жадного алгоритма. Создан класс `Backpack`, реализующий рюкзак и использовавшийся в рекурсивной функции для работы с выбираемыми в каждой ветви предметами. Функция `branch_and_bound()` выполняет поставленную задачу методом ветвей и границ и возвращает номера предметов, реализующих оптимальное решение.

7.2. Особенности реализации в системе компьютерной алгебры

Для решения задачи о ранце в системе компьютерной алгебры нами использовался пакет прикладных программ MATLAB, т. к. он имеет широкий спектр функций

и предоставляет большие возможности по работе с матрицами. В целом, синтаксисы языка в MATLAB и C++ имеют много общего, поэтому сложностей в реализации алгоритмов в MATLAB после успешного написания их на языке C++ не возникло. Была написана функция `sort_by_specific_price()`, выполняющая сортировку выбором по удельной стоимости предметов, а также функции `greedy_bound()` и `find_best()`, аналогичные функциям в программе на C++, отвечающим за оценку сверху с помощью жадного алгоритма и поиска лучшего решения в подветви соответсвенно. Для всех перечисленных функций мы использовали m-файлы. Функция `branch_and_bound()` выполняет поставленную задачу методом ветвей и границ и возвращает номера предметов, реализующих оптимальное решение.

8. Реализация приближенной схемы полностью полиномиального времени

8.1. Особенности реализации на языке C++

При нахождении решения задачи о ранце с помощью приближенной схемы полностью полиномиального времени может потребоваться большое количество дополнительной памяти на составление описанной ранее таблицы $w(k, p)$. Памяти будет требоваться тем больше, чем большую точность `double eps` мы задаем. Функция `polynomial_time_scheme()` выполняет поставленную задачу, реализуя приближенную схему полностью полиномиального времени, и возвращает номера предметов, реализующих полученное приближенное решение. Функция `greedy_price_min()` осуществляет оценку снизу решения задачи о ранце. Она осуществляет жадный алгоритм и нужна для нахождения коэффициента P_{min}. Для удобства составления таблицы $w(k, p)$ использовалась структура `struct Pseudo_solution`, хранящая номера предметов `std::list<int> item_numbers` некоторого решения и минимальный вес, необходимый для того, чтобы уложить предметы с номерами, не превосходящими k, общей стоимостью не менее $p \leq S$, где k и p — строка и столбец таблицы.

8.2. Особенности реализации в системе компьютерной алгебры

При написании алгоритмов в MATLAB была полезна информация из книги [5]. Представление системой компьютерной алгебры MATLAB данных в виде таблиц оказалось вполне удобным при реализации задачи о ранце в этом пакете. Для хранения предметов использовалась структура из трех ячеек, аналогичная структуре с тремя полями в реализации на C++. Была написана функция `greedy_price_opt()`,

которая, используя жадный алгоритм, находит оценку снизу для заданного набора вещей. Функция `polynomial_time_scheme()` выполняет поставленную задачу, реализуя приближенную схему полностью полиномиального времени, и возвращает номера предметов, реализующих полученное приближенное решение.

9. Сравнение методов

Выбор того или иного выбора метода здесь целиком опирается на специфику конкретной задачи, а точнее от её входных данных. Отметим выявленные в ходе работы недостатки и преимущества обоих методов.

Метод ветвей и границ.

Преимущества метода:

- Дает точное решение задачи.
- Интуитивно понятен и просто реализуем.
- Не требует дополнительного выделения памяти.
- Для некоторых задач способен быстро выдавать ответ, несмотря на большое количество входных данных.

Недостатки метода:

- Время работы программы, использующей такой метод, в общем случае стремительно растет при увеличении входных данных, поэтому использование этого алгоритма на большом количестве случайных входных данных ($n \gtrsim 40$) становится неразумным.

Как уже отмечалось, количество убранных из рассмотрения вариантов с помощью метода ветвей и границ сильно зависит от входных данных. Его целесообразно применять в том случае, когда удельные ценности предметов отличаются значительно [2].

Приближенная схема полностью полиномиального времени.

Преимущества метода:

- Решает задачу за полиномиальное время.
- Известна оценка точности для полученного этим методом решения, более того эта оценка может быть задана.

Недостатки метода:

- Находит только приближенное решение.
- Требует дополнительного выделения памяти для таблицы.

Отметим, что для малого количества предметов ($n \lesssim 20$) методы работают одинаково быстро. В целом, метод ветвей и границ нам кажется более интуитивно понятным и просто реализуемым методом, по сравнению с приближенной схемой полностью

полиномиального времени. В частных случаях метод ветвей и границ способен выдавать ответ быстрее второго метода. Конкретные примеры рассмотрим далее.

10. Примеры решения модельных задач

На некоторых примерах сравним рабому алгоритмов с помощью метода ветвей и границ и приближенной схемы полностью полиномиального времени. Результаты показаны для реализаций на языке C++. Точность задавалась равной $\varepsilon = 0{,}1$, то есть приближенный алгоритм должен найти решение, отличающее от оптимального не более чем на 10%.

Входные данные показаны в виде считываемого файла. Формат ввода следующий:

```
n          // количество предметов
w1 p1      // вес и стоимость 1-го предмета
w2 p2      // вес и стоимость 2-го предмета
...
wn pn      // вес и стоимость n-го предмета
W          // вместимость ранца
```

Пример 1. Работа методов при малом количестве предметов $n = 10$. Вместимость ранца $W = 43$. Результат отработки программы показан в таблице 1. Методы выдали разные решения, однако суммарная стоимость этих решений совпала.

Файл ввода	Результат			
	Метод ветвей и границ		ПСППВ	
	Вывод	Время (сек)	Вывод	Время (сек)
10 15 14 13 15 2 12 7 8 13 4 5 7 7 7 11 8 6 16 8 4	1 2 3 4 9	0,013	2 3 4 6 7 9	0,109
43	Точность решения 100%			

Таблица 1. Пример работы алгоритмов при $n = 10$

Пример 2. Работа методов при сравнительно большом количестве случайных предметов $n = 50$. Были заданы случайным образом 50 предметов с вещественными стоимостями и весами. Результат отработки программ показан в таблице 2.

Метод	Стоимость	Вес	Время (сек)	Точность
Метод ветвей и границ	$6{,}96 \cdot 10^8$	$7{,}94 \cdot 10^5$	1373	100%
ПСППВ	$6{,}94 \cdot 10^8$	$7{,}31 \cdot 10^5$	0,6	99,65%

Таблица 2. Пример работы алгоритмов при $n = 50$

Пример 3. Использовался такой же набор предметов, как в примере 2, но вместимость указана так, что все предметы помещаются в ранец. В этом случае метод ветвей и границ сразу найдет оптимальное решение и будет убирать из рассмотрения все оставшиеся ветви. Результат отработки программ показан в таблице 3.

Метод	Стоимость	Вес	Время (сек)	Точность
Метод ветвей и границ	$25{,}84 \cdot 10^8$	$36{,}66 \cdot 10^8$	0,03	100%
ПСППВ	$25{,}76 \cdot 10^8$	$4{,}79 \cdot 10^8$	0,5	99,7 %

Таблица 3. Ситуация, когда все предметы помещаются в ранец

11. Заключение

В результате выполнения практического задания мы познакомились с задачей о ранце и реализовали алгоритмы «метод ветвей и границ» и «приближённая схема полностью полиноминального времени» для решения этой задачи комбинаторной оптимизации. Для реализации этих методов были использованы язык программирования `C++` и система компьютерной алгебры MATLAB. При решении задачи мы исследовали и сравнили два метода на предмет эффективности, выявили их преимущества и недостатки. По завершении ознакомительной практики 4-го семестра мы овладели навыками разработки программ на `C++`, реализующих заданные алгоритмы, а также закрепили знания о работе с пакетами прикладных программ.

Литература

1. Задача о рюкзаке // Википедия. Свободная энциклопедия.
   URL: https://ru.wikipedia.org/wiki/Задача_о_рюкзаке
   (дата обращения: 25.05.2020).
2. Методы решения задач динамического программирования: задача о рюкзаке // Платформа YouTube.
   URL: https://www.youtube.com/watch?v=AlLqrMJkqEY&t=535s
   (дата обращения: 25.05.2020).
3. Информатика. Алгоритм «укладки рюкзака». Центр онлайн-обучения «Фоксфорд» // Платформа YouTube.
   URL: https://www.youtube.com/watch?v=HtrgxH3feME&t=196s
   (дата обращения: 25.05.2020).
4. «Эффективные алгоритмы» Лекция 3: Приближенные алгоритмы.
   URL: https://compsciclub.ru/media/courses/2007-autumn/spb-efficientalgorithms/slides/efficientalgorithms_lecture_071007.pdf
   (дата обращения: 25.05.2020).
5. Кетков Ю.Л., Кетков А.Ю., Шульц М.М. MATLAB 7: программирование, численные методы. — СПБ.: БХВ-Петербург, 2005. — 752 с.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Фундаментальные науки
КАФЕДРА Прикладная математика

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Студент Мелешко Евгений Дмитриевич

фамилия, имя, отчество

Группа ФН2-41Б

Тип практики: Ознакомительная практика

Название предприятия: НУК ФН МГТУ им. Н.Э. Баумана

Студент _____ Мелешко Е.Д.
подпись, дата *фамилия и.о.*

Руководитель практики _____ Савельева И.Ю.
подпись, дата *фамилия и.о.*

Оценка _____

2020 г.