

Санкт-Петербургский Политехнический университет Петра Великого  
Институт прикладной математики и механики  
Кафедра «Прикладная математика»

Курсовая работа  
по дисциплине  
"Методы оптимизации"

Выполнили студенты гр. 33631/2:

Лебедь П.С.

Туников Д.А.

Федоров Д.А.

Преподаватель:

Родионова Е. А.

Санкт-Петербург, 2019 г.

## Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>2</b>
<b>2</b>	<b>Формализация задачи</b>	<b>2</b>
<b>3</b>	<b>Анализ задачи</b>	<b>2</b>
<b>4</b>	<b>Алгоритм</b>	<b>3</b>
<b>5</b>	<b>Обоснование алгоритма</b>	<b>3</b>
<b>6</b>	<b>Сложность алгоритма</b>	<b>3</b>
<b>7</b>	<b>Вычисление логического условия</b>	<b>4</b>
<b>8</b>	<b>Пример решения задачи для <math>n = 7</math></b>	<b>4</b>
<b>9</b>	<b>Программа, решающая поставленную задачу</b>	<b>5</b>
<b>10</b>	<b>Выводы</b>	<b>5</b>

## Список таблиц

1	$x_1 \Rightarrow x_2$ . . . . .	4
---	---------------------------------	---

## 1 Постановка задачи

1. Известны: набор инвестиционных проектов  $p_{1..n}$ , начальное количество денежных средств -  $s$ . Каждый проект  $p_i$  определяется инвестиционной стоимостью -  $a_i$  и прибылью -  $b_i$ . Необходимо найти максимальное по прибыли подмножество проектов  $\{p_{k_1}, \dots, p_{k_m}\}$ , в которые выгоднее всего вложить денежные средства, и для реализации которых суммарно имеется достаточно средств.
2. Необходимо, чтобы полученное подмножество удовлетворяло логическому условию вида:  
 $p_{l_1} \& \dots \& p_{l_d} \Rightarrow (p_{e_1} \vee \dots \vee p_{e_w}) \& (p_{j_1} \vee \dots \vee p_{j_z})$ ,  
где  $l_1 \dots l_d, e_1 \dots e_w, j_1 \dots j_z$  - не фиксированы (это подмножества  $1..n$ ).
3. Для поставленной задачи нужно придумать, обосновать и реализовать алгоритм решения. И представить результаты работы для  $n = 7$ .

## 2 Формализация задачи

1. Количество денежных средств положительно, т.о.  
 $a_i > 0, b_i > 0, s > 0 \forall i \in 1..n$
2. Представим множество проектов  $\{p_1, \dots, p_n\}$  как булевый вектор [1], т.е.  $p_i \in (0, 1)$ , где  $p_i = 1$ , если мы вкладываемся в проект,  $p_i = 0$  - если не вкладываемся.  $\forall i \in 1..n$ , тогда  
Стоимость проектов:  $c = \sum_{i=1}^n p_i * a_i$   
Прибыль проектов:  $nvp = \sum_{i=1}^n p_i * b_i$   
В заданных обозначениях должно выполняться  $c \leq s$ , где  $s$  - начальное количество денежных средств
3. Условия вида  
 $p_{l_1} \& \dots \& p_{l_d} \Rightarrow (p_{e_1} \vee \dots \vee p_{e_w}) \& (p_{j_1} \vee \dots \vee p_{j_z})$  должны выполняться по правилам булевой алгебры [1]

Т.о. образом, необходимо найти  $\max: nvp = \sum_{i=1}^n p_i * b_i$ , при условиях:

1.  $c = \sum_{i=1}^n p_i * a_i \leq s$ , где  $s$  - начальное количество денежных средств
2.  $p_{l_1} \& \dots \& p_{l_d} \Rightarrow (p_{e_1} \vee \dots \vee p_{e_w}) \& (p_{j_1} \vee \dots \vee p_{j_z})$

## 3 Анализ задачи

Первый пункт поставленной задачи эквивалентен "задаче об упаковке рюкзака". [1] (В этой задаче необходимо уложить в рюкзак максимальные по суммарной стоимости предметы, т.к. чтобы суммарный вес не превышал максимальной вместимости (по весу) рюкзака. В исходной задаче вместо стоимости

предметов - прибыль от инвестиции в проект, а вместо вместимости рюкзака - начальные денежные средства). Эта задача принадлежит к NP-полным, т.о. она не имеет известного решения за полиномиальное время. Задача решается следующими методами [1]:

1. С применением динамического программирования со сложностью  $O(NW)$ .
2. Перебором (например, с использованием метода ветвей и границ) со сложностью  $O(2^N)$ .

Существенным является наличие второго условия (логического). Оно не позволяет использовать метод динамического программирования, т.к. не гарантируется, что результат будет ему удовлетворять. А перерасчет максимального значения после проверки приведет к полному перебору в худшем случае. Т.о. образом необходимо предварительно найти все множества вида  $\{p_1, \dots, p_n\}$ , где  $p_i \in (0, 1)$ , которые будут удовлетворять логическому условию. Построение таблицы истинности [1] имеет сложность  $O(2^N)$ . (Фактически перебираются все возможные подмножества  $n$ -го булева пространства). Значит, решение задачи будем проводить перебором.

## 4 Алгоритм

Перебираем все подмножества булева  $n$ -го булева пространства. Для каждого булева вектора проверяем выполнения логического условия. Если условие выполняется, считаем значения стоимости проекта и прибыли проекта. Если стоимость проекта не больше начального количества денежных средств, считаем прибыль проектов и добавляем в множество возможных по логическому условию проектов. Из множества возможных по логическому условию проектов выбираем наибольшее по прибыли значение. Это значение и соответствующий ему булевый вектор - ответ. Если получившееся множество пустое, значит не существует подмножества проектов, в которые при заданных условиях можно вложиться.

## 5 Обоснование алгоритма

Т.к. перебор осуществляется по всем вариантам, из которых выбирается максимальное по прибыли значение - алгоритм решит задачу.

## 6 Сложность алгоритма

Перебор всех векторов  $n$ -го булева пространства -  $2^N$  шагов. Действительно, каждый элемент вектора может принимать одно из двух значений. Порядок следования элементов существенен. Т.о.  $\tilde{P}(2, n) = 2^n$  (Число размещений с повторениями [1]). При этом на каждом шаге считаются суммы  $c = \sum_{i=1}^n p_i * a_i$ ,  $nvp = \sum_{i=1}^n p_i * b_i$  - 2 \*  $n$  шагов. Т.о. сложность алгоритма -  $O(2^N * N)$ .

## 7 Вычисление логического условия

$$p_{l_1} \& \dots \& p_{l_d} \Rightarrow (p_{e_1} \vee \dots \vee p_{e_w}) \& (p_{j_1} \vee \dots \vee p_{j_z})$$

Логическое выражение типа следствие имеет следующую таблицу истинности

Таблица 1:  $x_1 \Rightarrow x_2$

$x_1$	$x_2$	Значение
0	1	1
0	0	1
1	0	0
1	1	1

$$p_{l_1} \& \dots \& p_{l_d} = 1 \Leftrightarrow x_i = 1 \forall i \in l_1 \dots l_d$$

$$p_{e_1} \vee \dots \vee p_{e_w} = 1 \Leftrightarrow \exists i \in e_1 \dots e_w : x_i = 1$$

Множества  $l_1 \dots l_d$ ,  $e_1 \dots e_w$ ,  $j_1 \dots j_z$  задаем как индексы исходного множества.  $\forall$  булева вектора вида  $p_1, \dots, p_n$

1. Вычисляем значение левой части выражения следствия: для этого проверяем в булевом векторе по индексам  $l_1 \dots l_d$  есть ли значение 0, если есть - выражение имеет значение 0, иначе - 1. Если значение 0, то согласно таблице истинности для следствия - ответ 1, иначе - левая часть имеет значение 1.
2. Вычисляем значение правой части выражения следствия: для этого проверяем в булевом векторе по индексам  $e_1 \dots e_w$ ,  $j_1 \dots j_z$  для каждого выражения в скобках есть ли значение 1, если есть - выражение имеет значение 1, иначе - 0. Если оба значения в скобках 1, то правая часть имеет значение 1, иначе - 0. Ответ в данном случае = выражение правой части.

Сложность вычисления логического условия не превосходит  $3 * n$  шагов.

## 8 Пример решения задачи для $n = 7$

Исходные данные: Наличных средств - 100, размерность  $n = 7$ , вектор стоимостей проектов {30, 20, 10, 10, 20, 50, 50}, вектор прибылей проектов {54, 56, 12, 1, 23, 60, 59}. Индексы в левом выражении {1, 2}, индексы первого выражения в правой части {3, 5, 6}, индексы второго выражения в правой части {7, 4}.

Условие имеет вид  $p_1 \& p_2 \Rightarrow (p_3 \vee p_5 \vee p_6) \& (p_7 \vee p_4)$

Результат {0, 1, 1, 0, 1, 1, 0}, стоимость проектов - 100, прибыль - 151. Подставляя, для проверки, результат в выражение получаем:

$$0 \& 1 \Rightarrow (1 \vee 1 \vee 1) \& (0 \vee 0) = 1$$

В результате, перебрано  $128 = 2^7$  вариантов.

## 9 Программа, решающая поставленную задачу

Работа выполнена на языке Python. Исходный код находится по адресу:  
<https://github.com/PavelLebed20/MetOptCourse/>

## 10 Выводы

Дополнительное логическое условие существенно при выборе алгоритма решения. Решение имеет большую алгоритмическую сложность  $O(2^N)$ . Так при размерности задачи  $n = 48$  на современном процессоре (тактовая частота 3.5 ГГц) вычисление займет больше сотни лет. При этом для ускорения алгоритм можно распараллелить. Задача при  $n = 7$  решается за приемлемо быстрое время. Динамическое задание условия по индексам реализовано. Выбор разных наборов индексов может существенно повлиять на ответ. При необходимости алгоритм можно изменить под любое логическое условие, достаточно уметь вычислять значение на любом булевом векторе.

## Список литературы

- [1] НОВИКОВ. *Дискретная математика: Учебник для вузов. 3-е изд. Стандарт третьего поколения.* ПИТЕР УДВ, 2019.