

МЕТОД ТАБЛИЦ ДОПУСТИМЫХ РЕШЕНИЙ В ЗАДАЧЕ О РАНЦЕ

В.Н. Бурков¹, В.О. Корепанов¹, А.Р. Кашенков²

¹ Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, Россия,

² Вологодский государственный университет, г. Вологда, Россия

Предложен новый подход к получению верхних и нижних оценок для задачи о ранце, в основе которого лежат операции «склеивания» решений одного слоя (k -слоем называется множество решений, у которых определены первые k компонент, соответствующих предметам). Рассмотрены две операции склеивания. При первой операции склеивания не теряется ни одно из допустимых решений, но могут появиться недопустимые. При этом мы получаем верхнюю оценку решений задачи. Во второй операции остаются только допустимые решения, но не все. При этом мы получаем нижнюю оценку решений задачи. В статье представлены результаты численных экспериментов по оценке времени и точности предлагаемого подхода в зависимости от объёма входных данных, объёма рюкзака и величины склеивания.

Ключевые слова: задача о ранце, верхняя и нижняя оценки решений, вычислительный эксперимент.

Введение

Задача о ранце относится к числу широко известных задач дискретной оптимизации. Её популярность, прежде всего, вызвана большим количеством приложений [1–5]. Методам решений задачи о ранце посвящены сотни публикаций (см. ссылки в [6, 7]). Тем не менее, поиск новых подходов к её решению или к получению оценок (верхних и нижних) продолжается. В статье предлагается новый подход к получению оценок в задаче о ранце, в основе которого лежат операции «склеивания» решений одного слоя (k -слоем называется множество решений, у которых определены первые k компонент, соответствующих предметам). Данный подход является модификацией метода допустимых решений, предложенного в [8], в части механизма «склеивания» решений. Представлены результаты вычислительных экспериментов о зависимостях времени поиска и точности решений от величины склеивания (под величиной склеивания понимается максимальная разность весов склеиваемых решений k -слоя).

1. Описание метода

Идею метода таблиц допустимых решений со склеиванием по разнице весов поясним на примере одномерной задачи о ранце: определить $x_i \in \{0,1\}$, $i = 1, \dots, n$, при которых

$$P(x) = \sum_i p_i x_i \rightarrow \max \quad (1)$$

при ограничении

$$\sum_i w_i x_i \leq C, \quad (2)$$

где $p_i \geq 0, w_i \geq 0, i = 1, \dots, n$ – ценности и веса предметов; C – объём (максимальный вес) рюкзака.

Без потери общности рассматриваем естественную очередность переменных в порядке нумерации $1, 2, \dots, n$. У метода есть параметр склеивания – максимальная разница весов, при которой происходит склеивание решений, обозначим его $s \in \mathbb{N}$. Последовательно для каждой переменной строим *таблицы допустимых подмножеств решений* (ДПР).

Сначала формируем тривиальную табл. 1 (которую можно опустить в дальнейшем) для первой переменной.

Таблица 1

ДПР для первой переменной

Вариант	0 ($x_1 = 0$)	1 ($x_1 = 1$)
Эффект	0	p_1
Вес	0	w_1

Далее строим табл. 2 для 1 и 2 переменной, где столбцы соответствуют вариантам значений первой переменной, а строки – второй; на их пересечении первое число соответствует эффекту (ценности), а второе – весу решения.

Таблица 2

ДПР для первых двух переменных

1	$p_2; w_2$	$p_1 + p_2; w_1 + w_2$
0	0; 0	$p_1; w_1$
2 1	0	1

Правило удаления недопустимых решений. Если для какой-то ячейки значение веса больше C , то мы этот вариант исключаем, так как это недопустимое решение. В итоге в таблице остаются только допустимые решения (остальные переменные можем считать равными 0), но поскольку это не все допустимые решения, то мы и называем их таблицами подмножеств допустимых решений.

Пусть $w_1 + w_2 > C$, тогда в табл. 2 для 1 и 2 переменной у нас осталось 3 допустимых решения. Далее строим табл. 3 ДПР для 1, 2 и 3 переменных.

Таблица 3

ДПР для 3 переменных

1	$p_3; w_3$	$p_1 + p_3; w_1 + w_3$	$p_2 + p_3; w_2 + w_3$
0	0; 0	$p_1; w_1$	$p_2; w_2$
3 1, 2	0	1	2

Далее удаляем недопустимые решения и переходим к формированию таблицы решений для переменных 4, 5 и т. д.

Формирование таблиц можно наглядно представить также в виде построения дерева слева направо по слоям, в котором вершинам дерева соответствуют решения (рис. 1), каждый слой соответствует рассмотрению новой переменной, а дуга вверх к следующему слою – это выбор значения 1 новой переменной, дуга вниз – выбор значения 0. В вершинах дерева нарастающим итогом пишем эффект (первое число) и вес (второе число).

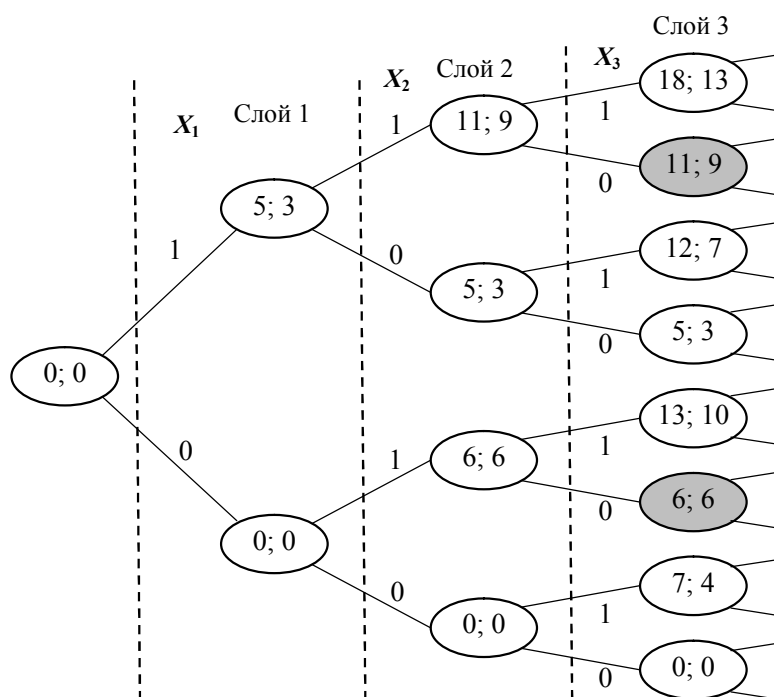


Рис. 1. Дерево решений

На рис. 1 приведён пример такого дерева для $p_1 = 5, w_1 = 3, p_2 = 6, w_2 = 6, p_3 = 7, w_3 = 4$. Ясно, что значениям i -й таблицы соответствуют вершины i -го слоя.

Эффект нарастающим итогом для k -й вершины i -го слоя дерева (соответственно, i -й таблицы) будем обозначать P_k^i , а вес нарастающим итогом – W_k^i . Без потери оптимальных решений для каждой таблицы/слоя дерева можно ввести правило доминирования.

Правило доминирования. Если для двух вершин k и q i -го слоя имеет место

$$P_k^i \geq P_q^i, W_k^i \leq W_q^i,$$

то вершину q исключаем. Иными словами, если упорядочить вершины слоя по возрастанию весов, то нужно удалить вершины, которые приводят к убыванию эффекта вершин в этом упорядочении.

Так, для вершин (11; 9) и (12; 7) (серые вершины третьего слоя на рис. 1) имеем: $12 > 11$, а $7 < 9$. Поэтому вершину (11; 9) исключаем. Аналогично исключаем вершину (6; 6), поскольку она доминируется вершиной (7; 4). Остаётся 6 недоминируемых решений.

Мы построили допустимые недоминируемые решения для первых трёх переменных.

Определение. Будем говорить, что решение $(x_1, x_2, \dots, x_k, 0, \dots, 0)$ на слое k *входит* в решение $(x_1', x_2', \dots, x_t', 0, \dots, 0)$ слоя t ($t > k$), если $x_i = x_i'$, для всех $1 \leq i \leq k$.

Наконец введём два правила «склеивания» вершин по разнице весов.

Правило склеивания 1. Склеиваем вершины с разностью вторых чисел меньше или равной s . При этом эффект склеенной вершины равен максимальному из эффектов, а вес – минимальному из весов склеиваемых вершин.

Склеенную вершину будем называть *проблемной*, из-за которой появляются *проблемные переменные* – так как мы соединяем два пути дерева, то в этих путях хотя бы одна переменная имеет разные значения – в одном 0, в другом 1. Одна проблемная вершина может порождать несколько проблемных переменных.

Таким образом, при склеивании по правилу 1 мы разным решениям назначаем одинаковый эффект и вес, причём для каждого из решений или увеличиваем его эффект, или уменьшаем вес, поэтому если это решение входит в другие допустимые решения, то оно не сделает их недопустимыми. При таком правиле склеивания мы не теряем ни одного допустимого решения, хотя могут появиться недопустимые решения (если вес уменьшился, и появилась возможность добавить другие переменные)¹.

Правило склеивания 2. Склеиваем вершины с разностью вторых чисел меньше или равной s . При этом эффект и вес склеенной вершины равен максимальному из эффектов и весов склеиваемых вершин соответственно.

При таком правиле склеивания мы можем терять допустимые решения, но недопустимых не появляется.

Пусть $s = 2$. На третьем слое дерева рис. 1 вершины (5; 3) и (7; 4) подходят под правило склеивания 1. Склеивая эти вершины, получаем вершину (7; 3). Действуя таким образом, мы получаем дерево (или последовательность таблиц), ширина каждого слоя которого (или размер каждой таблицы) не превышает C/s .

Теорема 1. Максимальный эффект вершин последнего слоя дерева (последней таблицы) при использовании правила склеивания 1 дает оценку сверху для исходной задачи.

Доказательство следует из отмеченного выше факта, что мы при правиле склеивания 1 не теряем ни одного допустимого решения.

Теорема 2. Максимальный эффект вершин последнего слоя дерева (последней таблицы) при использовании правила склеивания 2 дает оценку снизу для исходной задачи.

Доказательство следует из отмеченного выше факта, что мы при правиле склеивания 2 можем терять допустимые решения.

Пример 1. Рассмотрим задачу с шестью предметами, данные о которых приведены в табл. 4. Пусть $B = 14$, а $s = 1$.

¹ Вообще говоря, после применения правила склеивания таблицу 1 уже некорректно называть таблицей подмножеств допустимых решений, но, понимая происходящее, не будем вводить новое название.

Таблица 4

Исходные данные примера 1

i	1	2	3	4	5	6
a_i	5	6	7	9	12	11
b_i	3	6	4	5	8	6

1 шаг. Формируем таблицу допустимых решений для предметов 1 и 2 (табл. 5).

Таблица 5

ДПР для 1 и 2 предметов

Вариант	0	1	2	3
Эффект	0	5	6	11
Вес	0	3	6	9

2 шаг. Формируем таблицу допустимых решений для первых трех предметов. Решение приведено в табл. 6.

Таблица 6

ДПР для 3 предметов

1	7; 4	12; 7	13; 10	18; 13
0	0; 0	5; 3	6; 6	11; 9
3 1, 2	0	1	2	3

Как отмечалось выше, доминируемые решения (6; 6) и (11; 9) исключаем. Результаты сведены в табл. 7.

Таблица 7

Удаление доминируемых решений

Вариант	0	1	2	3	4	5
Эффект	0	5	7	12	13	18
Вес	0	3	4	7	10	13

Склеиваем решения (5; 3) и (7; 4) в (7; 3) по правилу склеивания 1. Получаем итоговую таблицу размера 5 (табл. 8).

Таблица 8

ДПР после операции склеивания

Вариант	0	1	2	3	4
Эффект	0	7	12	13	18
Вес	0	3	7	10	13

3 шаг. Формируем таблицу допустимых решений для первых четырех предметов. Решение приведено в табл. 9.

Таблица 9

ДПР для 4 предметов

1	9; 5	16; 8	21; 12	—	—
0	0; 0	7; 3	12; 7	13; 10	18; 13
4 1–3	0	1	2	3	4

Имеются доминируемые решения (13; 10) и (18; 13). Результаты сведены в табл. 10.

Таблица 10

Удаление доминируемых решений

Вариант	0	1	2	3	4	5
Эффект	0	7	9	12	16	21
Вес	0	3	5	7	8	12

Склеиваем решения (12; 7) и (16; 8), получаем (16; 7). Получаем итоговую табл. 11.

Таблица 11

ДПР после операции склеивания

Вариант	0	1	2	3	4
Эффект	0	7	9	16	21
Вес	0	3	5	7	12

4 шаг. Формируем таблицу допустимых решений для первых пяти предметов. Решение приведено в табл. 12.

Таблица 12

ДПР для 5 предметов

1	12; 8	19; 11	21; 13	–	–
0	0; 0	7; 3	9; 5	16; 7	21; 12
5 1–4	0	1	2	3	4

Решение (16; 7) доминирует (12; 8), а (21; 12) доминирует (21; 13). Склеиваем решения (19; 11) и (21; 12), получаем (21; 11). Оставшиеся решения приведены в табл. 13.

Таблица 13

ДПР после операции склеивания

Вариант	0	1	2	3	4
Эффект	0	7	9	16	21
Вес	0	3	5	7	11

5 шаг. Формируем табл. 14 подмножеств допустимых решений для всех шести предметов.

Таблица 14

ДПР для 6 предметов

1	11; 6	18; 9	20; 11	27; 13	–
0	0; 0	7; 3	9; 5	16; 7	21; 11
6 1–5	0	1	2	3	4

Решение (21; 11) доминирует (20; 11). Это последний шаг, поэтому склеивание смысла не имеет. Итоговые решения приведены в табл. 15.

Таблица 15

Итоговая таблица решений

Вариант	0	1	2	3	4	5	6	7
Эффект	0	7	9	11	16	18	21	27
Вес	0	3	5	6	7	9	11	13

Лучшее решение (27; 13). Этому решению соответствует решение $x_1 = x_3 = x_4 = x_6 = 1$. Но это решение является недопустимым и, следовательно, мы получили только оценку сверху.

Применим метод ветвей и границ.

Проблемными являются переменные x_1 , x_3 и x_4 . Делим множество всех решений на два подмножества, в первом $x_3 = 1$, во втором $x_3 = 0$.

Оценка первого подмножества ($x_3 = 1$). Если $x_3 = 1$, то решение (16; 7) табл. 11 шага 3 превращается в решения (16; 9) и (12; 7), для первого $x_5 = 0$, $x_6 = 0$, а эффект 16, для второго $x_5 = 0$, $x_6 = 1$, а эффект 23. Оптимальным становится второе решение.

Оценка второго подмножества ($x_3 = 0$). В табл. 8 решение (7; 3) превращается в решение (5; 3), приводящее к решению (25; 14).

Выбираем второе подмножество. Ему соответствует решение ($x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 1$, $x_5 = 0$, $x_6 = 1$) с эффектом 25.

2. Результаты экспериментов

Цель каждого эксперимента – оценить зависимость *измеряемого результата* (например, время работы, точность решения, см. далее) работы алгоритма от изменения одного из параметров входных данных – параметра эксперимента.

Набор экспериментов следующий:

Эксперимент оценки времени работы алгоритма точного решения в зависимости от объёма входных данных, при фиксированном объёме рюкзака.

Эксперимент оценки времени работы алгоритма точного решения в зависимости от объёма входных данных, при изменяемом объёме рюкзака.

Эксперимент оценки времени работы алгоритма точного решения в зависимости от объёма рюкзака.

Эксперимент времени работы алгоритмов со склеиванием в зависимости от величины параметра склеивания.

Эксперимент потери точности алгоритмов со склеиванием в зависимости от величины параметра склеивания.

2.1. Модель данных

В данном разделе вводятся необходимые обозначения, описание процесса эксперимента далее в разделе 2.2.

Входные данные эксперимента это набор $G = (N, n, P, W, c, t)$ и множество значений A_x параметра эксперимента, где параметр эксперимента x – один из набора $\{n, c, s\}$:

– N – определяет количество запусков алгоритма в одном эксперименте и, следовательно, определяет количество случайных наборов входных данных для алгоритма и число результатов работы алгоритма, по умолчанию $N^0 = 100$;

– n – «объём входных данных», количество предметов во входных данных для задачи рюкзака, по умолчанию $n^0 = 700$;

– $P = \{p_{\min}, p_{\min} + 1, \dots, p_{\max}\} \subset \mathbb{N}$ – множество возможных значений стоимостей предметов, по умолчанию $P^0: p_{\min}^0 = 100, p_{\max}^0 = 2000$;

– $W = \{w_{\min}, w_{\min} + 1, \dots, w_{\max}\} \subset \mathbb{N}$ – множество возможных значений весов предметов, по умолчанию $W^0: w_{\min}^0 = 10, w_{\max}^0 = 20\,000$;

– $c \in (0, 1)$ – доля объёма рюкзака от суммы весов предметов, по умолчанию $c^0 = 0,3$;

– t – количество величин параметра склеивания, по умолчанию $t^0 = 20$.

При заданных параметрах G генерируются *входные данные* для экспериментов $D(G) = ((p^i, w^i))_{i=1}^N, C, A_s^0$:

1. Генерация N наборов предметов: $\{(p^i, w^i)\}_{i=1}^N$, сортированных по возрастанию весов w^i , где $p^i \in P^n$ – вектор стоимостей предметов, каждый элемент вектора – реализация дискретной случайной величины, равномерно распределённой на P ; $w^i \in W^n$ – вектор весов предметов,

каждый элемент вектора – реализация дискретной случайной величины, равномерно распределённой на W .

2. Расчёт единого объёма рюкзака для N наборов предметов – доля c от математического ожидания суммы n равномерно распределённых случайных величин на W :

$$C = \lfloor c n(w_{\max} - w_{\min})/2 \rfloor.$$

3. Расчёт множества значений параметра склеивания $A_s^0(m, w^1, \dots, w^N) = \{s_1, \dots, s_m\}$:

$$dw^i = \{w_2^i - w_1^i, \dots, w_n^i - w_{n-1}^i\};$$

$$a^i = Q(dw^i; 0, 1), \quad b^i = \max(dw^i),$$

где $Q(x; 0, 1)$ – квантиль уровня 0,1 от эмпирического распределения x .

$$\bar{a} = \frac{1}{N} \sum_{i=1}^N a^i, \quad \bar{b} = \frac{1}{N} \sum_{i=1}^N b^i,$$

$$s_k = \bar{a} + (k-1) \frac{\bar{b}-\bar{a}}{m-1}, \quad k = 1, \dots, m.$$

Опишем *результаты работы алгоритмов*.

Точный алгоритм возвращает параметры решения (стоимость P , вес W) и время работы алгоритма T по заданным векторам цен p , весов w и объёму рюкзака C :

$$(P, W, T) = KSolver(p, w, C);$$

$$KSolver: \mathbb{N}^n \times \mathbb{N}^n \times \mathbb{N} \rightarrow \mathbb{R}_+^3.$$

Алгоритм поиска верхних и нижних оценок возвращает по сути те же значения, а на вход получает дополнительно параметр величины склеивания s :

$$(uP, uW, uT) = UpKSolver(p, w, C, s);$$

$$(dP, dW, dT) = DownKSolver(p, w, C, s);$$

$$(Up|Down)KSolver: \mathbb{N}^n \times \mathbb{N}^n \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}_+^3.$$

Значения $P, W, T, uP, uW, uT, dP, dW, dT$ являются *измеряемыми результатами* работы алгоритмов в экспериментах.

2.2. Процесс эксперимента

Для каждого значения параметра эксперимента N раз запускается алгоритм решения на случайных данных, в итоге получаем данные, с помощью которых можем оценить зависимость измеряемых результатов от изменения параметра эксперимента.

Для каждого значения a параметра эксперимента x вычисляется результат $R(a, D(G(a)))$ как набор N измеряемых результатов работы алгоритма.

Например, пусть P измеряемый результат, n – параметр эксперимента, тогда $R(a, D(G(a)))$ рассчитывается следующим образом:

1) выбор очередного значения параметра $n = a$;

2) генерация входных данных $D(G(a))$;

3) запуск алгоритма на входных данных и получение измеряемых результатов $(P^i, W^i, T^i) = KSolver(p^i, w^i, C)$, $i = 1, \dots, N$;

4) Результат $R = \{P^1, \dots, P^N\}$.

Результатом эксперимента обозначим набор данных $\{(a, R(G(a)))\}_{a \in A_n}$.

Измеряемый результат алгоритма – T – время работы алгоритма точного решения задачи о ранце. Параметр эксперимента – объём входных данных n .

2.3. Эксперимент № 1

Объём рюкзака фиксирован (не доля от суммы весов предметов) и одинаков для всех запусков алгоритма: $C^1 = 2\,500\,000$.

Множество A_n выбрано как $\{300, 320, \dots, 1200\}$, а набор входных данных эксперимента имеет вид

$$\{(N^0, 300, S_p^0, S_w^0, C^1, 0), (N^0, 320, S_p^0, S_w^0, C^1, 0), \dots, (N^0, 1000, S_p^0, S_w^0, C^1, 0)\}.$$

Схема эксперимента изображена на рис. 2.

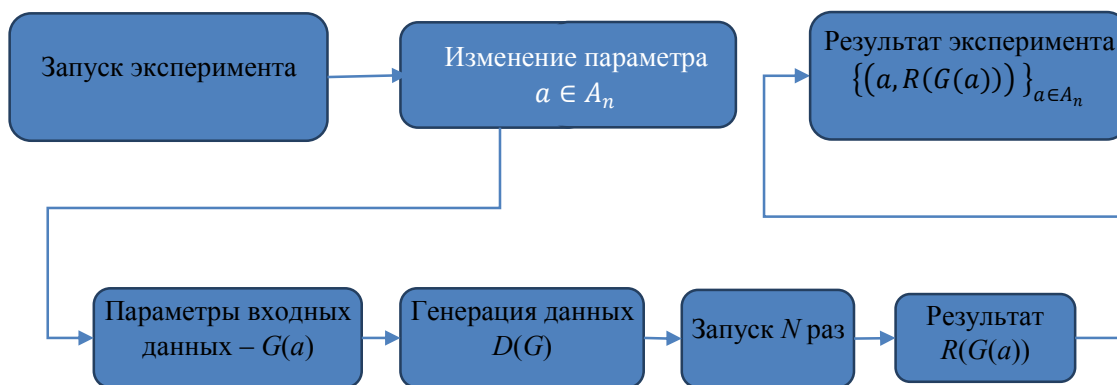


Рис. 2. Схема эксперимента

Результат эксперимента представлен на рис. 3 в форме «ящик с усами».

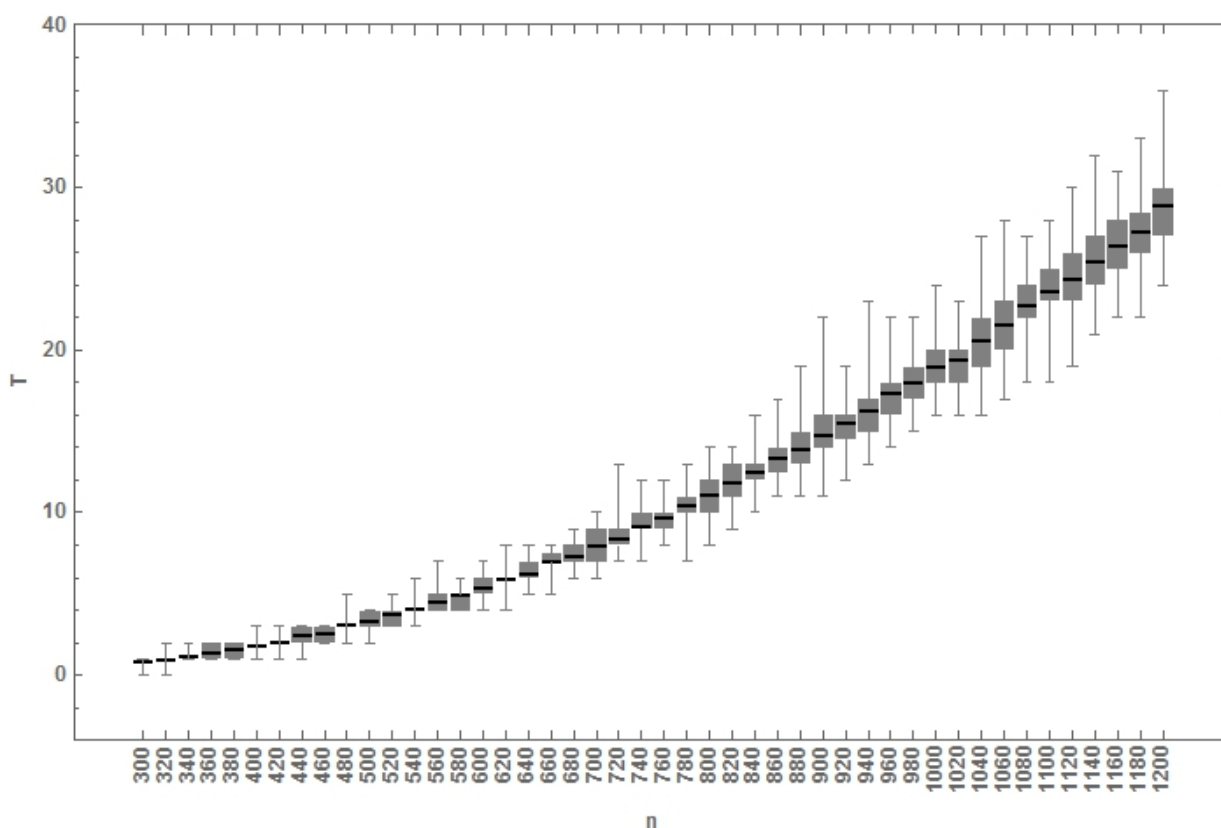


Рис. 3. Диаграмма зависимости времени работы алгоритма точного решения задачи о ранце от объема входных данных

Теоретически сложность алгоритма при данных условиях должна быть линейна по n : $T = O((2n - 1)C)$, так как таблиц будет $2n - 1$ и в каждой не больше C решений. Что подтверждается данным экспериментом как визуально – кривизна в начале соответствует только моменту, когда дополнительные вычисления (например, выделение памяти) соотносимы по величине с основными, так и регрессией – лучший коэффициент детерминации (0,99) у нелинейной регрессии гиперболой с тремя параметрами a , b и c :

$$T = b\sqrt{n^2/a + 1} - c,$$

которая в асимптоте стремится к линейной функции.

Коэффициент детерминации так же высок и у нелинейной регрессии экспонентой, для уточнения модели важно будет провести оптимизацию по памяти предложенного алгоритма, с тем, чтобы время на дополнительные вычисления сократилось.

2.4. Эксперимент № 2

Измеряемый результат алгоритма – T – время работы алгоритма точного решения задачи о ранце.

Параметр эксперимента – объём входных данных n .

Множество A_n выбрано $\{300, 320, \dots, 1200\}$, а набор входных данных эксперимента имеет вид

$$\{(N^0, 300, S_p^0, S_w^0, c^0, 0), (N^0, 320, S_p^0, S_w^0, c^0, 0), \dots, (N^0, 1200, S_p^0, S_w^0, c^0, 0)\}.$$

Отличие данного эксперимента от предыдущего состоит в том, что здесь объём рюкзака меняется в зависимости от n в соответствии с долей c^0 . Результат эксперимента представлен на рис. 4 в форме «ящик с усами».

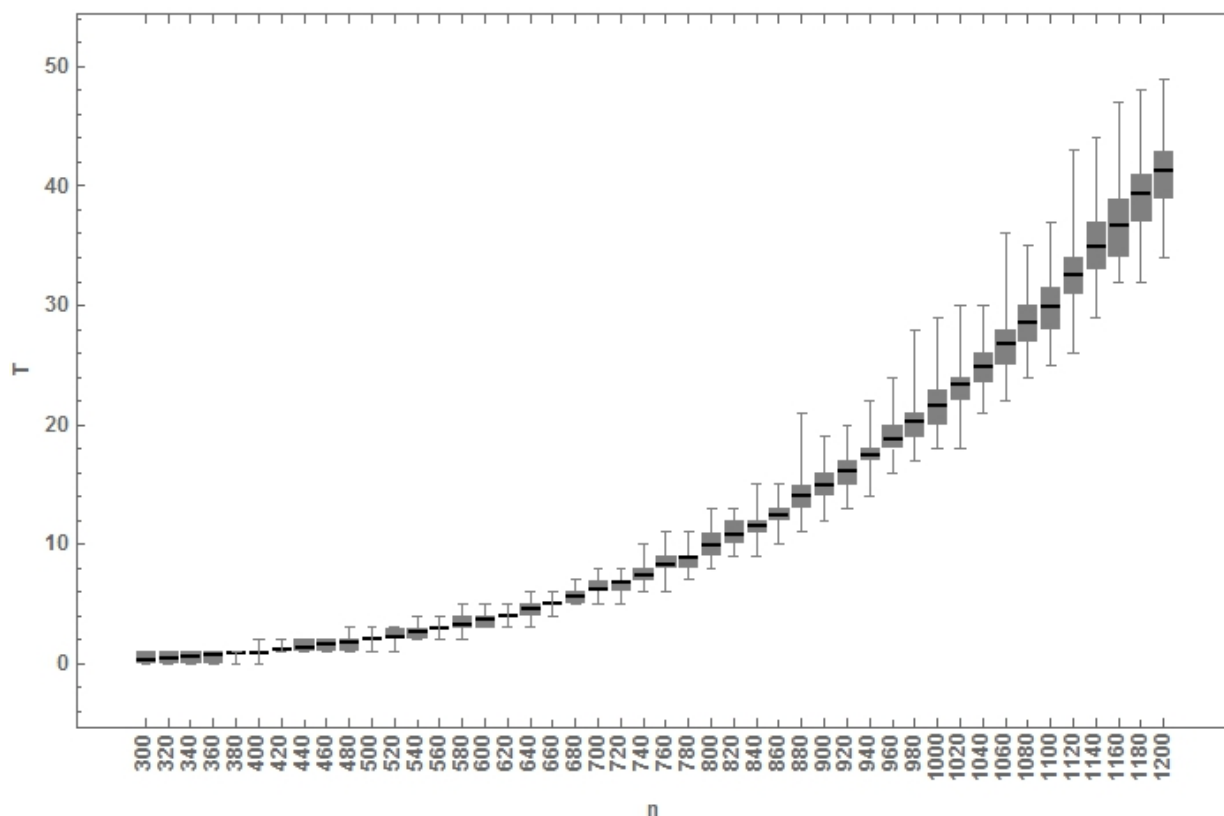


Рис. 4. Диаграмма зависимости времени работы алгоритма точного решения задачи о ранце от объёма входных данных при относительной доле объёма рюкзака

Теоретически сложность алгоритма при данных условиях должна быть квадратична по n : $T = O((n-1)n c^0 (w_{\max}^0 + w_{\min}^0)/2)$. Эта оценка подтверждается: линейная регрессия полиномом 2 степени с тремя параметрами a, b и c показывает большой коэффициент детерминации (0,98):

$$T = a + bn + cn^2.$$

2.5. Эксперимент № 3

Измеряемый результат алгоритма – T – время работы алгоритма точного решения задачи о ранце.

Параметр эксперимента – объём рюкзака c в процентах от суммарного веса предметов.

Множество A_c выбрано $\{5, 10, \dots, 95\}$, а набор входных данных эксперимента примет вид

$$\{(N^0, 900, S_p^0, S_w^0, 0.05, 0), (N^0, 900, S_p^0, S_w^0, 0.1, 0), \dots, (N^0, 900, S_p^0, S_w^0, 0.95, 0)\}.$$

Результат эксперимента представлен на рис. 5.

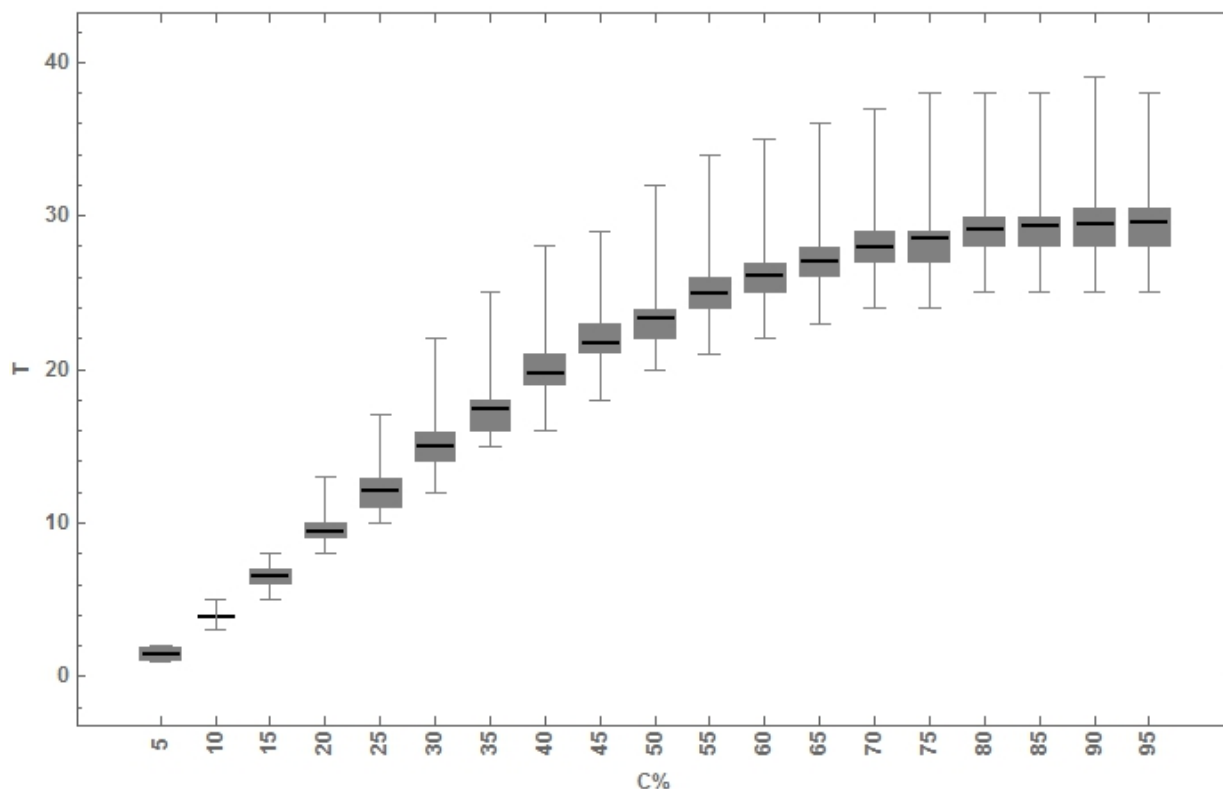


Рис. 5. Результат эксперимента № 3

Интересно, что рост времени поиска решения замедляется после $c = 50\%$, а при максимальных значениях объёма рюкзака время поиска решений практически не меняется. Можно предположить, что при больших значениях происходит практически полный перебор всех вариантов.

Следующий эксперимент посвящён алгоритмам получения верхних и нижних оценок. Поскольку ясно, что алгоритмы при значении величины склеивания $= 0$ будут вести себя так же, как и точный алгоритм, то выбирать параметром эксперимента n и C нет смысла – нужно смотреть на изменение измеряемого результата алгоритма в зависимости от величины склеивания.

2.6. Эксперимент № 4

Измеряемый результат алгоритма – T^U, T^D – время работы алгоритмов получения верхней и нижней оценки решения задачи о ранце.

Параметр эксперимента – величина склеивания s .

Множество $A_s = S(m, w^1, \dots, w^N)$.

Множество A_c имеет вид $\{5, 10, \dots, 95\}$, а набор входных данных эксперимента примет вид

$\{(N^0, 900, S_p^0, S_w^0, 0.05, 0), (N^0, 900, S_p^0, S_w^0, 0.1, 0), \dots, (N^0, 900, S_p^0, S_w^0, 0.95, 0)\}$.

Результат эксперимента представлен на рис. 6: для каждого параметра склеивания рядом расположены результаты верхней (серый цвет) и нижней оценки (красный цвет).

Кажется очевидным, что для обоих алгоритмов результат практически одинаков – медианы везде равны, а границы интервалов равны почти везде. Примем гипотезу, что время выполнения верхней и нижней оценки в данном тесте одинаково (данные результаты есть реализации одной случайной величины) $T^U = T^D$, тогда можно построить график для совмещённых результатов (рис. 7).

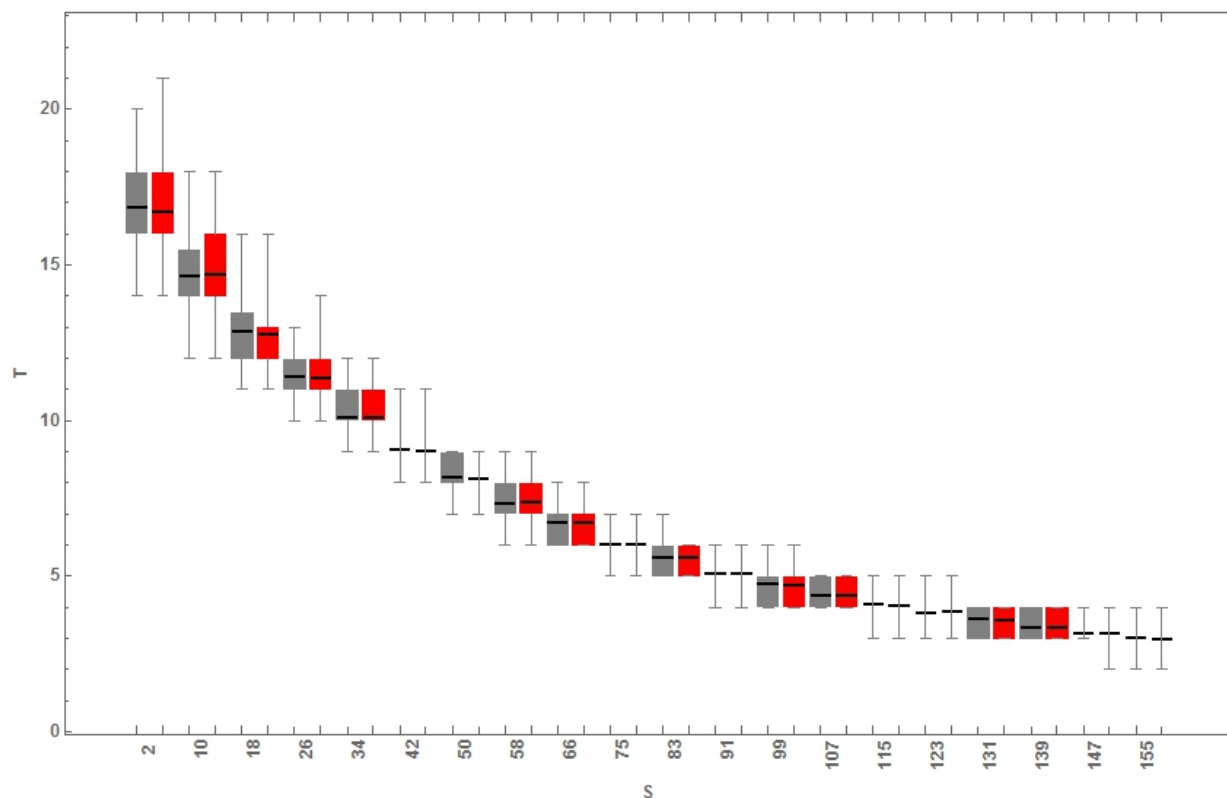


Рис. 6. Зависимость времени получения верхней (серый) и нижней (красный) оценки от величины склеивания

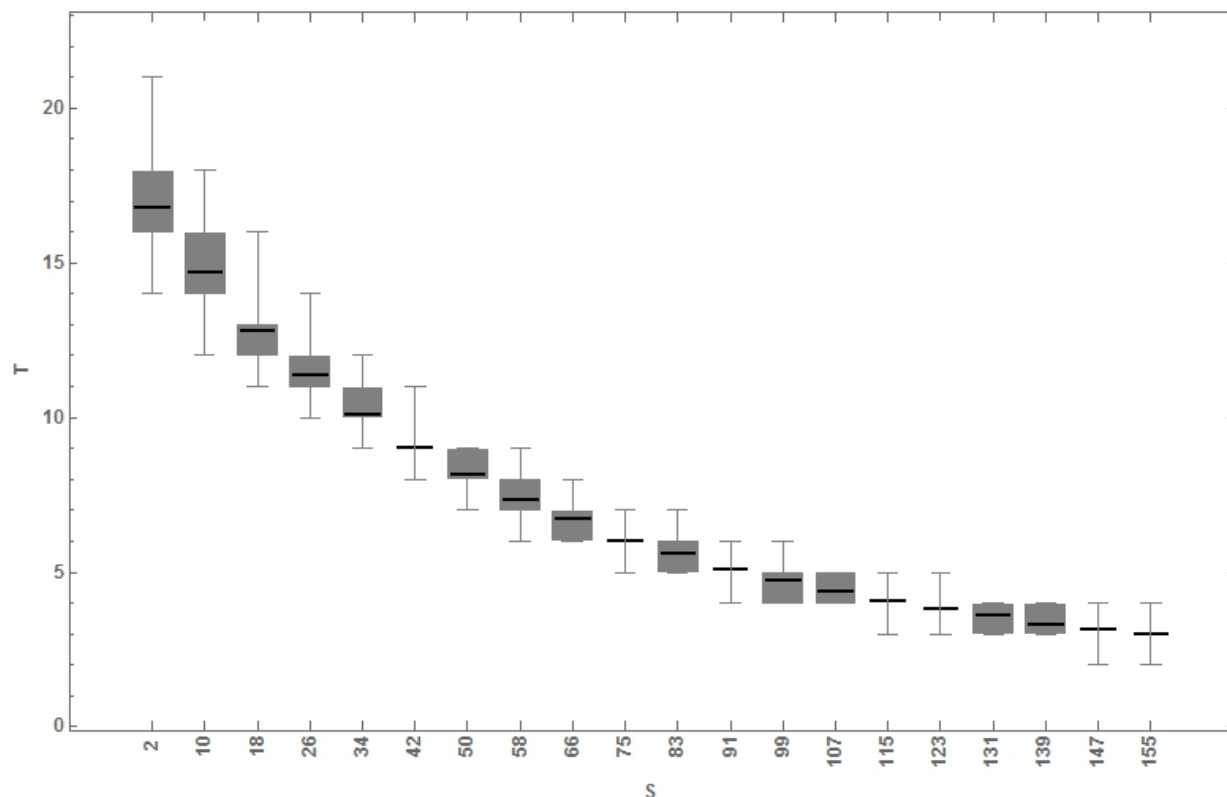


Рис. 7. Зависимость времени получения верхней и нижней оценки от величины склеивания

Теоретически скорость убывания времени работы алгоритма будет гиперболической по s : $T = O((n - 1) C/s)$. Нелинейная регрессия гиперболой действительно даёт большой коэффициент детерминации 0,99.

2.7. Эксперимент № 5

Измеряемый результат алгоритма – потери точности алгоритмов получения верхней и нижней оценки решения задачи о ранце.

Параметр эксперимента – величина склеивания s .

Множество $A_s = S(m, w^1, \dots, w^N)$.

Введём обозначения

$$(P^i, W^i, t^i) = KSolver(p^i, w^i, C), i = 1, \dots, N;$$

$$(uP_j^i, uW_j^i, uT_j^i) = UpKSolver(p^i, w^i, C, s_j^i) \in \mathbb{R}_+^3, i = 1, \dots, N, j = 1, \dots, m;$$

$$(dP_j^i, dW_j^i, dT_j^i) = DownKSolver(p^i, w^i, C, s_j^i) \in \mathbb{R}_+^3, i = 1, \dots, N, j = 1, \dots, m.$$

Определим величину «потеря точности верхней/нижней оценки» решения как отношение разницы между верхней/нижней оценкой эффекта рюкзака и эффектом рюкзака в точном решении к этому же эффекту рюкзака в точном решении. Обозначим потерю точности верхней и нижней оценки на j -м параметре склеивания для i -го набора предметов соответственно muP_j^i и mdP_j^i . Тогда:

$$muP_j^i = (uP_j^i - P^i)/P^i \geq 0;$$

$$mdP_j^i = (P^i - dP_j^i)/P^i \geq 0.$$

Результат для потери точности нижней оценки представлен на рис. 8.

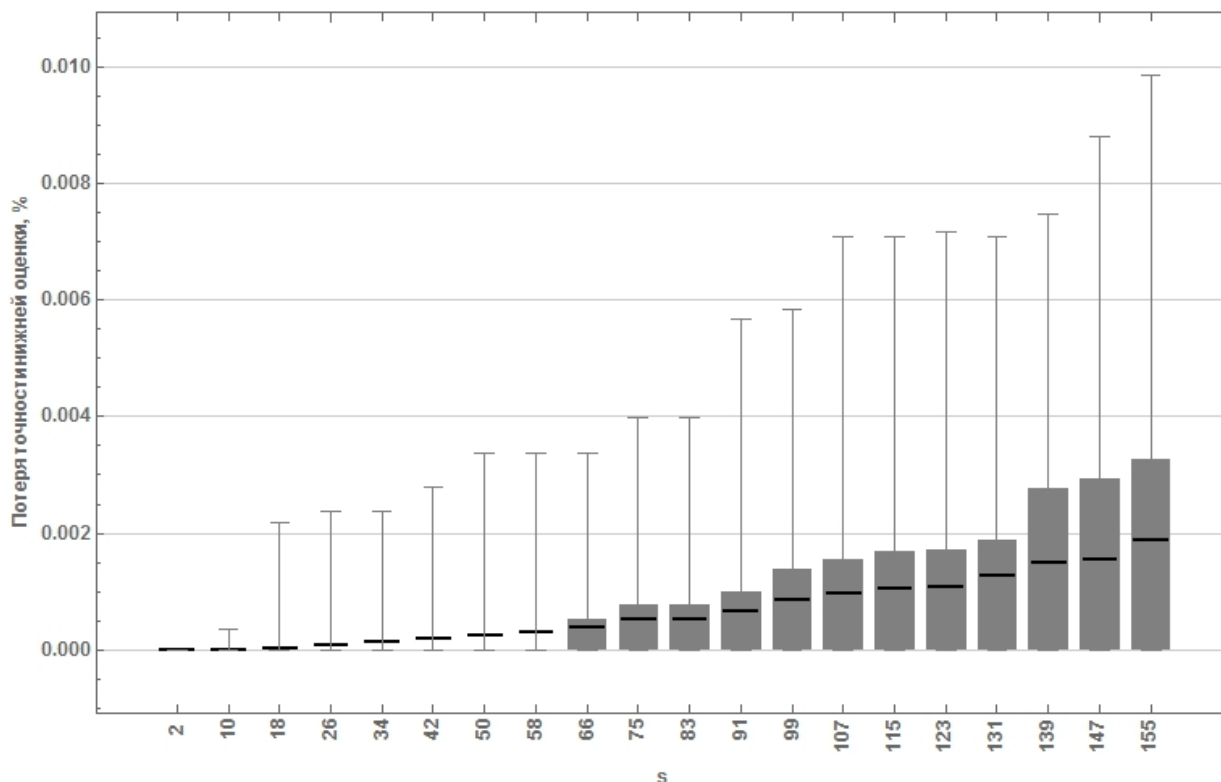


Рис. 8. График потери точности нижней оценки рюкзака

Очевидно, потеря точности для данных значений параметра склеивания слишком мала, поэтому мы увеличили множество A_s до значения $s = 500$ (почти в 2,5 раза), на рис. 9 и 10 представлены результаты для потери точности верхней и нижней оценки соответственно.

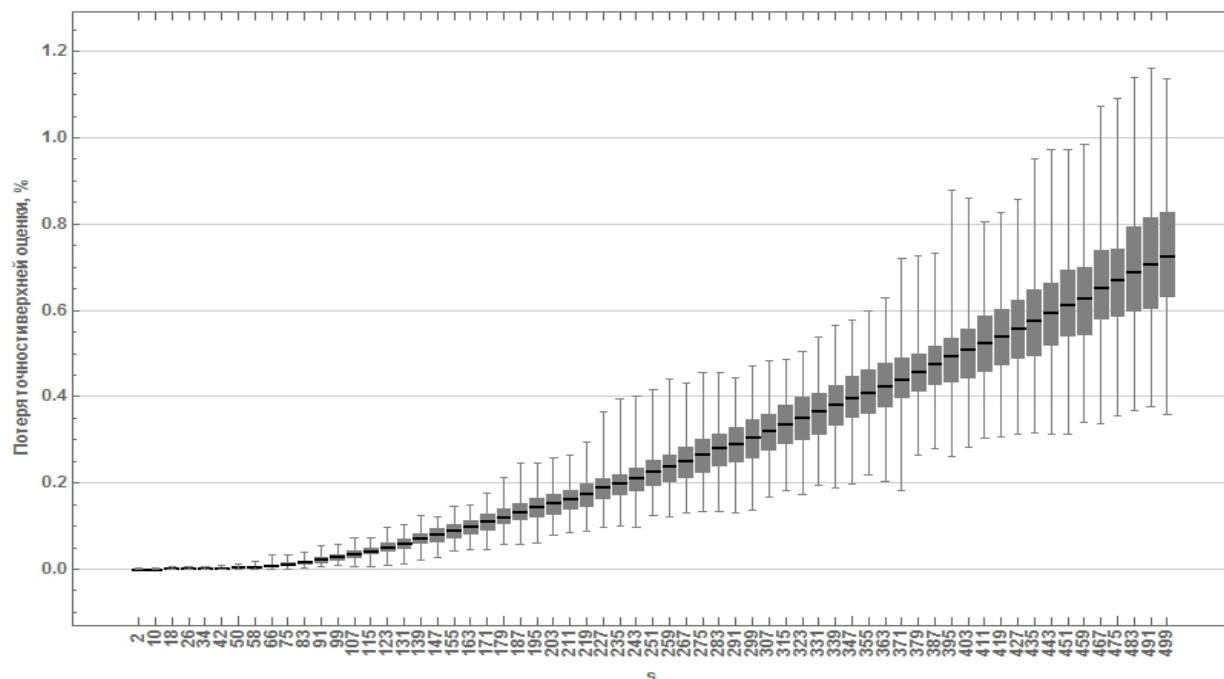


Рис. 9. График потери точности верхней оценки рюкзака

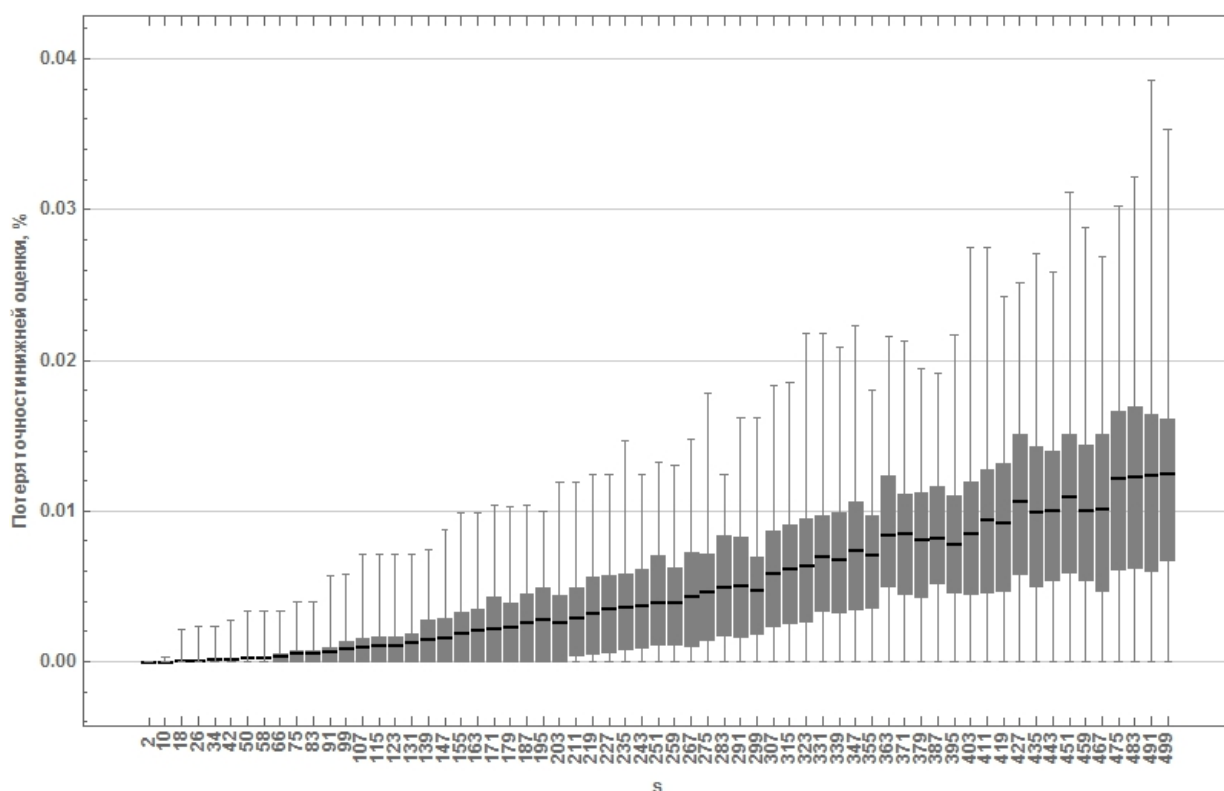


Рис. 10. График потери точности нижней оценки рюкзака

Для данных экспериментов пока не получена теоретическая оценка зависимости роста потери точности от величины склеивания.

Для построения регрессии потери точности верхней оценки использовались линейная регрессия полиномом 2 степени и нелинейные регрессии экспонентой и гиперболой, все регрессии с тремя параметрами. Регрессия экспонентой с тремя параметрами a , b и c имеет вид:

$$T^U = e^{b(x-c)} - a.$$

Все регрессии показывают высокий коэффициент детерминации, который приблизительно равен 0,91; 0,96; 0,96 для полинома, экспоненты и гиперболы соответственно.

Для нижней оценки более адекватны модели экспоненты и гиперболы, но коэффициенты детерминации низки: приблизительно равны 0,46; 0,68; 0,68 соответственно.

Попробуем сравнить потерю точности с уменьшением времени работы алгоритмов верхней и нижней оценки. Данные экспериментов № 2 и 4 говорят о том, что при изменении величины склеивания от 0 до 155 время работы алгоритма уменьшилось приблизительно в 3 раза. В то же время результат эксперимента № 5 говорит, что потеря точности верхней оценки составляет 0,1 %, а нижней – 0,01 %. При этом величина склеивания 155 есть $\approx 0,006$ % от средней величины рюкзака $\frac{(w_{\max}-w_{\min})}{2}c^0n = 9995 \cdot 0,3 \cdot 900 \approx 2,7 \cdot 10^6$. Таким образом, незначительным изменением величины склеивания можно добиться кратного уменьшения времени работы алгоритмов при потере точности 0,1 %.

Следует однако заметить, что порядок отношения потери точности (0,1 %) к величине склеивания в процентах от объема рюкзака (0,006 %) составляет 10^2 , что с учетом регрессионных моделей говорит о быстрой потере точности при дальнейшем увеличении параметра склеивания. Будущие исследования могут быть посвящены в том числе данному вопросу.

Заключение

В работе предложен новый метод получения верхних и нижних оценок для задачи о ранце, представлен пример решения простой задачи с использованием верхней оценки и метода ветвей и границ.

Проведенный анализ результатов экспериментов с предложенным алгоритмом получения верхних и нижних оценок подтверждает теоретические оценки зависимости измеряемых результатов алгоритма решения от изменения параметров: объем данных, объем рюкзака, величина склеивания. Получены данные экспериментов для оценки зависимости роста потери точности от величины склеивания, для чего пока нет теоретической оценки.

Полученные данные говорят о том, что можно сократить время расчёта оценок в разы при потере точности 0,1 %, хотя потеря точности растёт достаточно быстро при дальнейшем увеличении величины склеивания, а время расчёта убывает по гиперболическому закону (медленнее).

Дальнейшие работы могут быть направлены на получение оценки зависимости потери точности от величины склеивания и на оптимизацию использования памяти в предложенных алгоритмах для уточнения анализа асимптотических оценок.

Работа выполнена при частичной поддержке грантов РФФИ №15-07-07790 и РФФИ №17-20-05216.

Литература

1. Diakaki, C. Towards a multi-objective optimization approach for improving energy efficiency in buildings / C. Diakaki, E. Grigoroudis, D. Kolokotsa // *Energy and Buildings*. – 2008. – Vol. 40, no. 9. – P. 1747–1754. DOI: 10.1016/j.enbuild.2008.03.002
2. Dynamic placement for clustered web applications / A. Karve, T. Kimbrel, G. Pacifici et al. // *Proceedings of the 15th international conference on World Wide Web, WWW '06*. – 2006. – P. 595–604. DOI: 10.1145/1135777.1135865
3. Gatzianas, M. Control of wireless networks with rechargeable batteries / M. Gatzianas, L. Georgiadis, L. Tassiulas // *IEEE Transactions on Wireless Communications*. – 2010. – Vol. 9, no. 2. – P. 581–593. DOI: 10.1109/TWC.2010.080903
4. Новый эффективный алгоритм решения задачи об инвестициях / Е.Р. Гафаров, А. Долгий, А.А. Лазарев, Ф. Вернер // *Автоматика и телемеханика*. – 2016. – № 9. – С. 150–166.
5. Загурских, Е.А. Разработка и реализация криптографической системы с открытым ключом, основанной на обобщении задачи о ранце для кольца многочленов с коэффициентами из поля: выпускная квалификационная работа магистра / Е.А. Загурских. – Новосибирск: Новосибирский государственный университет, 2017. – 24 с.
6. Martello, S. New trends in exact algorithms for the 0–1 knapsack problem / S. Martello, D. Pisinger, P. Toth // *European Journal of Operational Research*. – 2000. – Vol. 123, no. 2. – P. 325–332. DOI: 10.1016/s0377-2217(99)00260-x

7. Kellerer, H. *Knapsack Problems* / H. Kellerer, U. Pferschy, D. Pisinger. – Springer-Verlag Berlin Heidelberg, 2004. ISBN 3-540-40286-1. MR 2161720. DOI: 10.1007 / 978-3-540-24777-7

8. Метод допустимых решений в многомерной задаче о ранце / В.Н.Бурков, И.В. Буркова, Б.К. Уандыков, Д.С. Чу // Экономика и менеджмент систем управления. – 2015. – Т.18, № 4.1. – С. 136–144.

Бурков Владимир Николаевич, д-р техн. наук, профессор, заведующий лабораторией 57, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва; vlab17@bk.ru.

Корепанов Всеволод Олегович, канд. техн. наук, старший научный сотрудник, лабораторией 57, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва; vkorepanov@ipu.ru.

Кашенков Александр Рудольфович, канд. техн. наук, доцент, Вологодский государственный университет, г. Вологда.

Поступила в редакцию 1 ноября 2017 г.

DOI: 10.14529/ctcr180204

THE METHOD OF TABLES OF FEASIBLE SOLUTIONS FOR THE KNAPSACK PROBLEM

V.N. Burkov¹, vlab17@bk.ru,

V.O. Korepanov¹, vkorepanov@ipu.ru,

A.R. Kashenkov²

¹ V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation,

² Vologda State University, Vologda, Russian Federation,

The paper proposes a new approach to obtaining upper and lower bounds for the knapsack problem, which is based on the operations of "gluing" together solutions of one layer (the k-layer is the set of solutions for which the first k components corresponding to items are defined). Two operations of gluing are considered. The first gluing operation does not lose any of the acceptable solutions, but it may appear inadmissible. Wherein, we obtain an upper bound for the solutions of the problem. In the second operation, only feasible solutions remain, but not all. In this case we obtain a lower bound for the solutions of the problem. The article presents the results of numerical experiments to estimate the time and accuracy of the proposed approach, depending on the input data size, the volume of the knapsack size and the volume of the gluing.

Keywords: knapsack problem, upper and lower bounds for solutions, computational experiment.

References

1. Diakaki C., Grigoroudis E., Kolokotsa D. Towards a Multi-Objective Optimization Approach for Energy Efficiency in Buildings. *Energy and Buildings*, 2008, vol. 40, no. 9, pp. 1747–1754. DOI: 10.1016/j.enbuild.2008.03.002.

2. Karve A., T. Kimbrel, G. Pacifici et al. Dynamic Placement for Clustered Web Applications. *Proceedings of the 15th International Conference on World Wide Web, WWW '06*, 2006, pp. 595–604. DOI: 10.1145/1135777.1135865.

3. Gatzianas M., Georgiadis L., Tassioulas L. Control of Wireless Networks with Rechargeable Batteries. *IEEE Transactions on Wireless Communications*, 2010, vol. 9, no. 2, pp. 581–593. DOI: 10.1109/TWC.2010.080903.

4. Gafarov E.R., Dolgii A., Lazarev A.A., Werner F. A new effective dynamic program for an investment optimization problem. *Automation and Remote Control*, 2016, vol. 77, iss. 9, pp. 1633–1648. DOI: 10.1134/S0005117916090101

5. Zagurskikh E.A. *Razrabotka i realizatsiya kriptograficheskoy sistemy s otkrytym klyuchom, osnovannoy na obobshchenii zadachi o rantse dlya kol'tsa mnogochlenov s koeffitsientami iz polya: vypusknaya kvalifikatsionnaya rabota magistra* [Development and Implementation of a Public Key Cryptographic System Based on the Generalization of the Knapsack Problem for a Polynomial Ring with Coefficients from a Field]. Novosibirsk, Novosibirsk State University, 2017. 24 p.

6. Martello S., Pisinger D., Toth P. New Trends in Exact Algorithms for the 0-1 Knapsack Problem. *European Journal of Operational Research*, 2000, vol. 123, no. 2, pp. 325–332. DOI: 10.1016/s0377-2217(99)00260-x

7. Kellerer H., Pferschy U., Pisinger D. *Knapsack Problems*. Springer-Verlag Berlin Heidelberg, 2004. ISBN 3-540-40286-1. MR 2161720. DOI: 10.1007 / 978-3-540-24777-7

8. Burkov V.N., Burkova I.V., Uandykov B.K., Chu D.S. [The Method of Admissible Solutions in the Multidimensional Problem on the Knapsack]. *Economics and Management of Control Systems*, 2015, vol. 18, no. 4.1, pp. 136–144. (in Russ.)

Received 1 November 2017

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Бурков, В.Н. Метод таблиц допустимых решений в задаче о ранце / В.Н. Бурков, В.О. Корепанов, А.Р. Кашенков // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2018. – Т. 18, № 2. – С. 38–53. DOI: 10.14529/ctcr180204

FOR CITATION

Burkov V.N., Korepanov V.O., Kashenkov A.R. The Method of Tables of Feasible Solutions for the Knapsack Problem. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2018, vol. 18, no. 2, pp. 38–53. (in Russ.) DOI: 10.14529/ctcr180204
