
СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ О РЮКЗАКЕ ДЛЯ ОПТИМИЗАЦИИ КОЛИЧЕСТВА ЗАЯВОК НА ИННОВАЦИИ¹

Картушин Дмитрий Юрьевич, бак.

Максименкова Анжела Руслановна, бак.

Угольницкий Геннадий Анатольевич, д-р физ.-мат. наук

Южный федеральный университет, ул. Большая Садовая, 105, Ростов-на-Дону, Россия, 344006; e-mail: ougoln@mail.ru

Цель: сравнительный анализ алгоритмов решения задачи отбора заявок на инновационные проекты в условиях ограниченного бюджета. *Обсуждение:* указанная проблема формализуется как задача о рюкзаке (ранце), для решения которой применяются методы дискретной оптимизации и генетические алгоритмы. Предлагается набор критериев эффективности алгоритмов и проводится соответствующий сравнительный анализ. *Результаты:* авторами программно реализованы алгоритмы дискретной оптимизации и генетические алгоритмы для решения задачи о рюкзаке применительно к отбору заявок на инновационные проекты, проведен анализ их сравнительной эффективности по ряду критериев, приведены результаты работы алгоритмов.

Ключевые слова: генетические алгоритмы, гибридные алгоритмы, дискретная оптимизация, отбор инновационных проектов.

DOI: 10.17308/meps.2017.3/1635

Введение

Опираясь на [8], данная работа посвящена как изучению новых используемых алгоритмов, так и более детальному описанию уже представленных. Представлены результаты сравнительного анализа эффективности всех указанных алгоритмов. В разделе 2 описываются алгоритмы дискретной оптимизации для решения задачи о рюкзаке и их программная реализация. Проводится сравнительный анализ эффективности указанных алгоритмов. Аналогично раздел 3 посвящен генетическим алгоритмам решения этой задачи и сравнительному анализу их эффективности применительно к модельной ситуации отбора заявок на реализацию инновационных проектов. В разделе 4 приводятся заключительные соображения.

¹ Работа выполнена при финансовой поддержке РФФИ, проект № 15-01-00432-а.

Методы дискретной оптимизации решения задачи о рюкзаке

В статье [8] подробно описаны приближенные методы для решения задачи о рюкзаке: жадный и эвристический алгоритмы. В данной работе приведен анализ точных алгоритмов.

- Полный перебор [10, 4]

Чтобы решить задачу, необходимо составить все комбинации наборов заявок и выбрать тот набор, стоимость которого не более заданной величины B , а общая эффективность (по отношению к другим подходящим наборам) максимальна. В терминах комбинаторики это означает, что нужно рассмотреть все перестановки из множества элементов N . Очевидно, что вычислительная сложность алгоритма полного перебора для решения задачи о рюкзаке равна $O(N!)$. Недостаток способа полного перебора всех заявок состоит в том, что если количество заявок велико, то алгоритм не даст решения за приемлемое время, поскольку факториал является чрезвычайно быстро растущей функцией. Итак, алгоритм основан на переборе всевозможных наборов (перестановок) заявок для рюкзака:

– Данный метод рекурсивный.

– Сначала в наборе имеются все N предметов, затем при переходе вглубь на один уровень рекурсии один предмет удаляется.

– Выход из рекурсии произойдет, когда список предметов станет пустым.

- Метод ветвей и границ [11, 10]

Данный метод основан на идее улучшенного перебора при помощи отсева подмножеств допустимых решений, заведомо не содержащих оптимальных решений. В настоящей работе применяется следующее правило отсева неоптимальных решений: не рассматриваются решения, которые в сумме дают эффективность меньшую, чем уже достигнутая при помощи другого набора заявок. Если находится набор заявок с наибольшей суммарной эффективностью, то максимальная эффективность присваивается такому набору.

- Метод динамического программирования реализован в два этапа [6, 1, 5]

1. Строится вспомогательная матрица A .

$$A(k, 0) = 0, A(0, s) = 0.$$

Возможны 2 варианта нахождения $A(k, s)$:

Если заявка k не попала в рюкзак, тогда $A(k, s) = A(k - 1, s)$.

Если заявка k попала в рюкзак, тогда $A(k, s) = A(k - 1, s - c_k) + p_k$
то есть: $A(k, s) = \max(A(k - 1, s), A(k - 1, s - c_k) + p_k)$.

2. Восстановление набора заявок, из которых состоит максимально эффективный рюкзак.

findAns(k, s)

```

Если  $A[k][s] == 0$  return;
Если  $A[k-1][s] == A[k][s]$ 
    findAns(k-1, s);
Иначе findAns(k-1, s - w[k])
    ans.push(k).

```

Таким образом, хотя сложность этого алгоритма $O(B*n)$ экспоненциальная от количества заявок, при ограниченных размерах рюкзака алгоритм достаточно эффективен.

На примере набора из 5 заявок с бюджетом, равным 7, рассмотрим работу алгоритма.

p	10	12	15	8	11
c	5	4	10	2	3

На первом этапе строится матрица A, столбцами которой служат возможные бюджеты (s), а строками – номера заявок (k).

	0	1	2	3	4	5	6	7
k=0	0	0	0	0	0	0	0	0
k=1	0	0	0	0	0	10	10	10
k=2	0	0	0	0	12	12	12	12
k=3	0	0	0	0	12	12	12	12
k=4	0	0	8	8	12	12	20	20
k=5	0	0	8	11	12	19	20	23

На втором этапе необходимо восстановить набор заявок с наибольшей эффективностью, то есть рассмотреть матрицу начиная с элемента $A(N,B)$ и определить, входит ли элемент в искомый набор. В таблице отмечены элементы, которые участвовали в рассмотрении. Элементы матрицы $A(2,4)$, $A(5,7)$ обозначают заявки, которые поместились в рюкзак и позволили получить точное решение задачи при заданном бюджете инноваций.

	0	1	2	3	4	5	6	7
k=0	0	0	0	0	0	0	0	0
k=1	0	0	0	0	0	10	10	10
k=2	0	0	0	0	12	12	12	12
k=3	0	0	0	0	12	12	12	12
k=4	0	0	8	8	12	12	20	20
k=5	0	0	8	11	12	19	20	23

Сравнительный анализ результатов

В данной работе проведено тестирование алгоритмов на различных наборах входных данных при заданном инновационном бюджете. Характеристики компьютера, на котором выполнялось тестирование: процессор IntelCore 2 Duo 2.66 GHz 2.66 GHz, оперативная память 2 Гб, тип системы – 32-разрядная операционная система.

Метод	Входные данные			Скорость работы (мин.)
	Количество заявок	Диапазон заявок	Бюджет	
Жадный алгоритм	10	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:007
	50	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:010
	200	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:012
Эвристический метод	10	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:02
	50	Стоимость (50, 210), Эфф-ть(100, 400)	500	00:00:024
	200	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:028
Метод перебора	10	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:02:71
	11	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:32:86
	12	Стоимость (50, 210), Эфф-ть (100, 400)	500	06:02:74
Метод ветвей и границ	10	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:046
	12	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:02:56
	14	Стоимость (50, 210), Эфф-ть (100, 400)	500	03:25:92
Метод динамического программирования	10	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:002
	50	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:004
	200	Стоимость (50, 210), Эфф-ть (100, 400)	500	00:00:011

По результатам исследования выявлено, что методы перебора, ветвей и границ и динамического программирования позволяют найти точное решение задачи с глобально максимальной эффективностью. При этом динамический алгоритм дает точное решение гораздо быстрее даже на большом наборе заявок, чего нельзя сказать о методе полного перебора, так как на наборе из 12 заявок данный метод работает 6 минут. Таким образом, для крупных компаний данный метод не подойдет для частого использования. Самый быстрый и эффективный алгоритм основан на методе динамического программирования. Главный недостаток метода заключается, говоря словами Беллмана, в «проклятии размерности» – его сложность катастрофически возрастает с увеличением размерности задачи. Но в данной работе алгоритм отлично справляется и с набором из 200, 1000 и 10 000 заявок.

Гибридный генетический алгоритм решения задачи о рюкзаке [2, 3, 12]

В дополнение к представленным в [8] генетическим алгоритмам (репродуктивный план Холланда и жадный генетический алгоритм с элитизмом) в данной работе рассматривается новый гибридный алгоритм, использующий частичный перебор. Основой для него послужил репродуктивный план Холланда, однако новый генетический алгоритм перед включением новой особи в популяцию проводит локальную оптимизацию найденного решения в единичной окрестности (по метрике Холланда) на предмет наличия лучших решений.

Локальная оптимизация выполняется следующим образом:

1. Установить счетчик равным 1.
2. Проверить, равна ли нулю координата у особи-наследника, соответствующей номеру счетчика.
3. Если да, то перейти к следующему шагу, иначе увеличить счетчик на 1 и перейти к шагу 2.

4. Проверить, не превышает ли стоимость проектов бюджет при включении туда стоимости заявки, соответствующей номеру счетчика.
5. Если да, то перейти к следующему шагу, иначе увеличить счетчик на 1 и перейти к шагу 2.
6. Рассматривать полученную особь как новую особь-наследника.
7. Увеличить счетчик на 1.
8. Если счетчик превышает размер хромосомы, то выйти из цикла, иначе перейти к шагу 2, иначе выйти из цикла.

Сравнительный анализ результатов [9]

В таблице приведены результаты двадцати запусков каждого из генетических алгоритмов: репродуктивный план Холланда (РПХ), жадный генетический алгоритм (ЖА) и алгоритм локальной оптимизации (ЛО). План Холланда и локальная оптимизация запускались с максимальным количеством поколений 15. Жадный алгоритм – 50.

№	Целевая функция			Расстояние до глобального экстремума			Суммарный вес		
	РПХ	ЖА	ЛО	РПХ	ЖА	ЛО	РПХ	ЖА	ЛО
1	698	698	698	0	0	0	120	120	120
2	698	698	698	0	0	0	120	120	120
3	698	693	693	0	5	5	120	119	119
4	661	688	698	37	10	0	120	117	120
5	698	698	698	0	0	0	120	120	120
6	698	698	693	0	0	5	120	120	119
7	661	693	698	37	5	0	120	119	120
8	698	688	698	0	10	0	120	117	120
9	692	693	698	6	5	0	120	119	120
10	698	698	692	0	0	6	120	120	120
11	698	698	698	0	0	0	120	120	120
12	693	698	698	5	0	0	119	120	120
13	698	698	698	0	0	0	120	120	120
14	692	677	698	6	21	0	120	117	120
15	698	668	698	0	30	0	120	120	120
16	692	693	698	6	5	0	120	119	120
17	683	688	698	15	10	0	120	117	120
18	698	698	698	0	0	0	120	120	120
19	698	692	698	0	6	0	120	120	120
20	692	698	698	6	0	0	120	120	120

Все запуски проводились на персональном компьютере с характеристиками: процессор – AMDFX(tm)-6300 Six-Core 3.50 GHz, ОЗУ – 8 ГБ, 64-разрядная ОС.

Основываясь на статистических данных, можно привести таблицу с различными численными показателями [7] эффективности алгоритмов:

Метод	Процент нахождения глобальных экстремумов	Среднее расстояние до глобального экстремума	Среднеквадратическое отклонение	Среднее время работы алгоритма (мин.)
РПХ	60 (12/20)	5,9 (0,845%)	12,514 (1,793%)	00:02:801
ЖА	50 (10/20)	5,35 (0,773%)	9,426 (1,35%)	00:00:445
ЛО	85 (17/20)	0,8 (0,114%)	2,074 (0,297%)	00:02:807

Бесспорным лидером почти по всем численным показателям является алгоритм локальной оптимизации. На примере с двенадцатью заявками он работает примерно с той же скоростью, что и алгоритм Холланда, но имеет намного больший процент точности и меньшие отклонения от экстремума. Жадный алгоритм позволяет получать решения в среднем лучшие, чем у репродуктивного плана Холланда при более быстрой работе.

Заключение

Любая задача оптимизации с конечным множеством допустимых решений может быть решена методом полного перебора. Однако на практике этот метод применим только для решений задач малой размерности. Метод ветвей и границ, исключая заведомо неоптимальные решения, позволяет найти решение за более приемлемое время, однако для большого набора заявок также не используется. Для крупных компаний подойдет использование таких алгоритмов, как жадный, эвристический и динамического программирования. Выбор того или иного метода зависит от ряда условий. В данной работе выделяется метод динамического программирования в связи с быстрым нахождением точного решения. Однако он дает большой объем вычислений, что при очень большом наборе заявок может сказаться на скорости работы. Среди генетических алгоритмов лучшие результаты показывает гибридный алгоритм локальной оптимизации. Это показывает, что использование дискретных методов в генетических алгоритмах может резко повысить эффективность. Так как перебор идет в единичной окрестности, алгоритм не приобретает экспоненциальную сложность полного перебора.

Итоговые результаты сравнения приведены в таблице с использованием критериев из работы [1].

Метод	Тип алгоритма	Плюсы	Минусы	Сложность	Скорость работы на наборе из 12 заявок, мин.
Жадный алгоритм	Приближенный	Высокая скорость; Работа с большими значениями n ; Простота реализации;	Неточное решение	$O(n \cdot \log(n) + n)$	00:00:008
Эвристический метод	Приближенный	Высокая скорость; Работа с большими значениями n ; Возможность доработки алгоритма; Возможность получить решение лучше, чем у жадного алгоритма	Неточное решение	$O(n \cdot \log(n) + 2 \cdot n)$	00:00:022

Окончание табл.

Метод	Тип алгоритма	Плюсы	Минусы	Сложность	Скорость работы на наборе из 12 заявок, мин.
Полный перебор	Точный	Простота реализации; Точное решение;	Небольшие входные данные; Временная сложность	$O(n!)$	06:02:74
Метод ветвей и границ	Точный	Возможно значительное сокращение времени по сравнению с полным перебором;	Небольшие входные данные; Временная сложность	$O(n!)$	00:02:56
Метод динамического программирования	Точный	Точное решение; Высокая скорость;	Большой объём вычислительной работы	$O(V \cdot n)$	00:00:003
Репродуктивный план Холланда	Приближенный	Устойчивость к предварительной сходимости;	Большие затраты оперативной памяти	Алгоритм ограничен по времени	00:02:801
Жадный алгоритм с элитизмом	Приближенный	Высокая скорость; решения близки к оптимальным;	Неустойчивость к предварительной сходимости;	Алгоритм ограничен по времени	00:00:445
Алгоритм локальной оптимизации	Приближенный	Очень хороший процент сходимости к оптимальному решению;	Самая длительная работа из представленных алгоритмов;	Алгоритм ограничен по времени	00:02:807

Список источников

1. Додонова М.М. *Изучение и сравнение методов решения различных модификаций задачи о рюкзаке*. Сибирский федеральный университет, ИМИФИ. Доступно: <http://www.lib.tpu.ru> (дата обращения: 10.04.17).
2. Емельянов В.В., Курейчик В.В., Курейчик В.М. *Теория и практика эволюционного моделирования*. Москва, Физматлит, 2003.
3. Еремеев А.В. *Генетические алгоритмы и оптимизация*. Омск, Омский гос. ун-т, 2008.
4. Задача о рюкзаке. Алгоритм полного перебора. Доступно: <http://vscode.ru> (дата обращения: 10.04.17).
5. Задача о рюкзаке. Доступно: <https://neerc.ifmo.ru> (дата обращения: 10.04.17).
6. Задача о рюкзаке: динамическое программирование. Доступно: <http://discopal.ispras.ru> (дата обращения: 10.04.17).
7. Звонков В.Б. Сравнительное исследование генетических алгоритмов и стайного алгоритма оптимизации. // *XIV Международная конференция «Проблемы управления и моделирования в сложных системах»*, Самара, Институт проблем управления сложными системами Российской академии наук (ИПУСС РАН), 2012, с. 786-797.
8. Картушин Д.Ю., Максименкова А.Р., Угольницкий Г.А. Оптимизация количества заявок при заданном бюджете инноваций // *Современная экономика: проблемы и решения*, 2016, по. 6(78), с. 20-33.
9. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. *Алгоритмы: построение и анализ*. 3-е изд. Москва, Вильямс, 2013.
10. Методические указания – Обзор методов разработки алгоритмов. Доступно: <http://www.twirpx.com> (дата обращения: 10.01.17).

11. Сигал И.Х., Иванова А.П. *Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы*. Москва, 2007.

12. Технологии анализа данных: BaseGroupLabs. Доступно: <https://basegroup.ru> (дата обращения: 12.04.17).

COMPARATIVE ANALYSIS OF THE EFFICIENCY OF ALGORITHMS FOR SOLUTION OF THE KNAPSACK PROBLEM OPTIMIZING THE QUANTITY OF APPLICATIONS FOR INNOVATIONS

Kartushin Dmitry Yurievich, B. Sc.

Maximenkova Angela Ruslanovna, B. Sc.

Ougolnitsky Guennady Anatolievich, Dr. Sc. (Phys.-Math.), Prof.

Southern Federal University, Bolshaya Sadovaya St., 105, Rostov-on-Don, Russia, 344006; e-mail: ougoln@mail.ru

Purpose: the article is devoted to the comparative analysis of algorithms of solution of the task of selecting applications for innovative projects with a limited budget. *Discussion:* this problem is formalized as a knapsack problem, for which discrete optimization methods and genetic algorithms are applied. A set of criteria for the efficiency of algorithms is proposed and an appropriate comparative analysis is carried out. *Results:* the authors have implemented discrete optimization and genetic algorithms to solve the knapsack problem with regard to the selection of applications for innovative projects, an analysis of their comparative effectiveness on a number of criteria. Some results are provided.

Keywords: genetic algorithms, hybrid algorithms, discrete optimization, selection of innovative projects.

References

1. Dodonova M.M. *Izuchenie i sravnenie metodov resheniia razlichnykh modifikatsii zadachi o riukzake*. Sibirskii federal'nyi universitet, IMIFI. Available at: <http://www.lib.tpu.ru> (accessed: 10.04.17). (In Russ.)
2. Emel'ianov V.V., Kureichik V.V., Kureichik V.M. *Teoriia i praktika evoliutsionnogo modelirovaniia*. Moscow, Fizmatlit, 2003. (In Russ.)
3. Ereemeev A.V. *Geneticheskie algoritmy i optimizatsiia*. Omsk, Omskii gos. un-t, 2008. (In Russ.)
4. Zadacha o riukzake. Algoritm polnogo perebora. Available at: <http://vscode.ru> (accessed: 10.04.17). (In Russ.)
5. Zadacha o riukzake. Available at: <https://neerc.ifmo.ru/> (accessed: 10.04.17). (In Russ.)
6. Zadacha o riukzake: dinamicheskoe programmirovaniie. Available at: <http://discopal.ispras.ru> (accessed: 10.04.17). (In Russ.)
7. Zvonkov V.B. Sravnitel'noe issledovanie geneticheskikh algoritmov i stainogo algoritma optimizatsii. *XIV Mezhdunarodnaia konferentsiia «Problemy upravleniia i modelirovaniia v slozhnykh sistemakh»*. Samara, Institut problem upravleniia slozhnymi sistemami Rossiiskoi akademii nauk (IPUSS RAN), 2012, pp. 786-797. (In Russ.)
8. Kartushin D.Iu., Maksimenkova A.R., Ugol'nitskii G.A. Optimizatsiia kolichestva zaiavok pri zadannom biudzhete innovatsii. *Sovremennaia ekonomika: problemy i resheniia*, 2016, no. 6(78), pp. 20-33. (In Russ.)
9. Kormen T.Kh., Leizeron Ch.I., Rinvest R.L., Shtain K. *Algoritmy: postroenie*

i analiz. 3-e izd. Moscow, Vil'iams, 2013. (In Russ.)

10. Metodicheskie ukazaniia – Obzor metodov razrabotki algoritmov. Available at: <http://www.twirpx.com> (accessed: 10.01.17). (In Russ.)

11. Sigal I.Kh., Ivanova A.P. *Vvedenie v*

prikladnoe diskretnoe programmirovanie: modeli i vychislitel'nye algoritmy. Moscow, 2007. (In Russ.)

12. Tekhnologii analiza dannykh: BaseGroupLabs. Available at: <https://basegroup.ru> (accessed: 12.04.17). (In Russ.)