
ОПТИМИЗАЦИЯ КОЛИЧЕСТВА ЗАЯВОК ПРИ ЗАДАННОМ БЮДЖЕТЕ ИННОВАЦИЙ¹

Картушин Дмитрий Юрьевич, бакалавр мат.

Максименкова Анжела Руслановна, бакалавр прикл. мат.

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

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

Цель: статья посвящена решению 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 на тестовых примерах.

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

DOI: 10.17308/meprs.2016.6/1464

1. Введение

В связи с необходимостью реализации программ инновационного развития перед государственными корпорациями стоит задача отбора заявок на реализацию инновационных проектов. Несомненно, эта задача актуальна и для частных компаний, нуждающихся в инновационном развитии. При этом бюджет развития всегда ограничен, особенно в сложившихся макроэкономических условиях.

Проблему отбора заявок на инновации естественно формализовать как задачу о рюкзаке [1], в результате решения которой отбираются заявки с лучшей потенциальной эффективностью, суммарно удовлетворяющие бюджетным ограничениям. Хотя эта задача хорошо известна, представляется целесообразной разработка программного комплекса, реализующего различные алгоритмы ее решения с учетом их сравнительной эффективности для разных условий и обеспечения дружественного пользовательского интерфейса.

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

Можно выделить две основные группы методов решения задачи о рюкзаке: алгоритмы дискретной оптимизации [2] и генетические алгоритмы [3], которые взаимно дополняют друг друга. Имеется ряд критериев эффективности алгоритмов, допускающих их сравнительный анализ [4].

Оставшаяся часть статьи организована следующим образом. В разделе 2 описываются алгоритмы дискретной оптимизации для решения задачи о рюкзаке. Аналогично раздел 3 посвящен генетическим алгоритмам решения этой задачи. В разделе 4 проводится сравнительный анализ эффективности указанных алгоритмов применительно к модельной ситуации отбора заявок на реализацию инновационных проектов. В разделе 5 приводятся заключительные соображения.

2. Методы дискретной оптимизации решения задачи о рюкзаке

Найти $S \subset N$ – множество финансируемых заявок, обеспечивающих максимальную суммарную эффективность при заданном бюджете развития (задача о рюкзаке):

$$\sum_{i \in S} p_i \rightarrow \max, \sum_{i \in S} c_i \leq B, \quad (1)$$

где $N = \{1, \dots, n\}$ – множество заявок на инновации; B – бюджет инновационного развития организации; c_i – стоимость реализации i -й заявки; p_i – потенциальная эффективность реализации i -й заявки.

- Жадный алгоритм

Жадный алгоритм делает на каждом шаге локально оптимальный выбор, и в дальнейшем этот выбор не отменяется. Пусть выполнено:

$$\frac{p_1}{c_1} \geq \frac{p_2}{c_2} \geq \dots \geq \frac{p_n}{c_n},$$

то есть все заявки упорядочены по отношению эффективности к стоимости [10]. Вначале рюкзак пуст. Последовательно перебираем элементы от 1 до n . Очередную заявку помещаем в рюкзак в случае, если ее стоимость не превышает заданный бюджет инновационного развития. Соответственно, при помещении заявки в рюкзак бюджет организации уменьшается на стоимость текущей заявки.

- Эвристический метод

В основе эвристических алгоритмов лежат идеи, основанные на интуиции, жизненном опыте [10]. Эвристические алгоритмы обычно имеют малую сложность вычислений и, как правило, выдают хорошие, близкие к оптимальным решения. Поэтому они часто применяются для решения трудных задач. В основу данного метода положим «жадные» эвристики. Сначала применяется жадный алгоритм, а затем в цикле выполняются следующие действия:

1. Выбирается заявка с наибольшей стоимостью из массива уже занесенных в рюкзак заявок и временно уменьшается бюджет на стоимость выбранной заявки.

2. Перебираем оставшиеся заявки. Если при добавлении новой заявки ее стоимость не превышает заданный бюджет, то помещаем ее в рюкзак. Так продолжается до тех пор, пока стоимость всех заявок не превысит бюджет.

3. Проверяем, больше ли суммарная эффективность новых добавленных заявок удаленной заявки с максимальной стоимостью. Если условие выполняется, то удаляем заявку с максимальной стоимостью и выходим из цикла.

4. Если все новые заявки рассмотрены, то возвращаем стоимости добавленных заявок в бюджет и выходим из цикла.

5. Иначе делаем проверку. Если новая заявка не помещается в рюкзак, то сравниваем эффективность данной заявки с эффективностью последней добавленной. Если эффективность новой заявки больше, то заменяем последнюю добавленную заявку текущей.

Данная логика реализована, чтобы максимально приблизиться к выполнению условий (1).

- Полный перебор

Так как множество возможных решений, конечно, можно перебрать все из них и найти лучшее. Как правило, на практике полный перебор невозможно осуществить из-за огромного числа рассматриваемых вариантов. Однако этот метод может применяться в случае малых размеров входных данных или в виде частичного перебора в других алгоритмах. Таким образом, для реализации полного перебора нужна процедура генерации всех допустимых решений. Речь идет о переборе всех перестановок целых чисел из некоторого диапазона, генерации всех подмножеств конечного множества. Соответственно, каждая заявка может быть либо помещена в рюкзак, либо нет. Каждая переменная X_j может принимать одно из двух значений – 1 и 0. Любому решению соответствует вектор из n нулей и единиц. Всего таких векторов 2^n [10].

- Метод ветвей и границ

Данный метод основан на идее улучшенного перебора. Одним из вариантов ускорения служит выполнение условия: если при добавлении заявки ее стоимость превышает бюджет, то далее ветка этого решения не рассматривается, так как рюкзак будет перегружен [6]. За счёт данной процедуры ускоряется решение задачи. Лишь в отдельных случаях метод ветвей и границ не будет эффективнее, чем метод перебора.

- Метод динамического программирования

Иногда перебор допустимых решений удастся существенно сократить применением методов динамического программирования [10]. Динамическое программирование обычно применяется к задачам, в которых искомый ответ состоит из частей, каждая из которых, в свою очередь, дает оптимальное решение некоторой подзадачи. При этом каждая из подзадач ре-

шается один раз, ее решение запоминается на случай, если та же подзадача встретится вновь. В основе алгоритма обычно лежит рекуррентное соотношение, связывающее оптимальные значения целевых функций подзадач. Например, алгоритм может быть реализован следующим образом: хранение наилучших частичных решений-наборов в хэш-таблице, т. е. для каждой стоимости, если существует набор с такой стоимостью, хранится максимальная эффективность. Стартовав с пустого множества частичных наборов и добавляя по одному заявки, в каждый момент имеем не более B «лучших» частичных наборов, помещающихся в рюкзак. В конце остается только выбрать самый эффективный из них. Таким образом, хотя сложность этого алгоритма $O(B * n)$ есть экспоненциальная функция количества заявок, при ограниченных размерах рюкзака алгоритм может быть полезен и эффективен [11].

3. Генетические алгоритмы решения задачи о рюкзаке [5, 6, 7, 8]

Решения P_i кодируются в следующем виде: $P_i = (x_1, \dots, x_n)$, где n – количество заявок на инновации, а каждая координата вектора, называемая далее геном, принимает значение 1 (если соответствующая заявка включена в выборку) или 0 (иначе).

Каждое поколение P^t состоит из фиксированного количества решений $P_i, i = 1, n$, называемых далее особями. Число t – номер генерации.

Функция приспособленности (Fitness) служит основным способом отбора более подходящих и жизнеспособных особей. Функция не имеет общего вида, применимого к любому генетическому алгоритму. Ее вид выбирается индивидуально, например, исходя из экспертных оценок, опытов или математических соображений. В случае задачи о рюкзаке функцию приспособленности удобно считать суммой потенциальных эффективностей, то

$$\text{есть } F(P_i) = \sum_{j=1}^n p(x_j), \text{ где } p(x_j) = \begin{cases} p_j, x_j = 1 \\ 0, x_j = 0 \end{cases}.$$

При решении задачи о рюкзаке имеет смысл вести контроль вместимости, чтобы не сохранять потенциальные решения, стоимость которых превышает бюджет. Этот контроль выполняется следующим образом.

- 1) Проверить выполнение неравенства $\sum_{j=1}^n c(x_j) \leq B$, где $c(x_j) = \begin{cases} c_j, x_j = 1 \\ 0, x_j = 0 \end{cases}$.

- 2) Если неравенство выполняется, то перейти к шагу 5, иначе продолжить.

- 3) Выбрать случайный ген особи.

- 4) Если значение гена равно 1, то изменить его на 0 и перейти к шагу 1, иначе перейти к шагу 3.

- 5) Завершить работу.

Условиями остановки, как правило, служат ограничение на макси-

мальное число шагов функционирования алгоритма (или ограничение на количество поколений) и определение его сходимости (обычно путем сравнения приспособленности популяции нескольких последних шагов) и остановки при стабилизации этого параметра. Условием остановки в данном случае будут:

а) ограничение максимального количества поколений во избежание заикливания;

б) наличие у заданного процента особей поколения одинакового значения функции приспособленности как признак того, что алгоритм сошелся к определенному результату.

Для сравнения приведем два генетических алгоритма:

- Репродуктивный план Холланда (классический генетический алгоритм).

1) Случайным образом сформировать начальную популяцию, состоящую из m особей: $P^0 = \{P_1, P_2, \dots, P_m\}$.

2) Вычислить приспособленность каждой особи $F(P_i), i = \overline{1, m}$.

3) Если выполнилось условие остановки, то завершить работу, иначе переход на шаг 3.

4) С помощью оператора «рулетки» выбрать особь P_q из популяции.

5) Аналогично выбрать вторую особь $P_r \neq P_q$ из популяции и применить оператор одноточечного кроссовера, получив новую особь P_s .

6) С определенной вероятностью выполнить оператор мутации на особи P_s .

7) С определенной вероятностью выполнить оператор инверсии на особи P_s .

8) Выполнить контроль вместимости рюкзака.

9) Поместить полученную хромосому в новую популяцию P^{t+1} .

10) Выполнить операции, начиная с шага 4, m раз.

11) Увеличить номер текущего шага $t := t + 1$.

12) Перейти на шаг 2.

В данном алгоритме используются операторы одноточечного кроссовера, мутации и инверсии. Оператор одноточечного кроссовера выполняется следующим образом:

1) Выбрать случайное натуральное число из отрезка $[1, n]$.

2) В особях $P_i = (x_1^i, \dots, x_n^i), P_j = (x_1^j, \dots, x_n^j)$ установить «точку разреза» слева от соответствующего разряда.

3) Скопировать в новую особь гены первой особи, расположенные слева от «разреза», и гены второй особи, расположенные справа от «разреза».

4) Завершить работу.

Оператор мутации выполняется следующим образом:

- 1) Выбрать случайное натуральное число из $[1, n]$.
- 2) В особи $P_i = (x_1, \dots, x_n)$ сменить значение соответствующего гена на противоположное.

3) Завершить работу.

Оператор инверсии меняет порядок следования хромосом на обратный.

- Жадный генетический алгоритм с элитизмом.

1) Случайным образом сформировать начальную популяцию, состоящую из m особей: $P^0 = \{P_1, P_2, \dots, P_m\}$.

2) Вычислить значение функции приспособленности для каждой особи $F(P_i), i = \overline{1, m}$.

3) Отсортировать популяцию по убыванию функции приспособленности.

4) Случайным образом выбрать особь P_q из популяции.

5) Случайным образом выбрать вторую особь $P_r \neq P_q$ из популяции и применить оператор жадного кроссовера, получив новую особь P_s .

6) Подсчитать ее функцию приспособленности и поместить в популяцию на соответствующее место, исключив последнюю особь.

7) Выполнить оператор мутации на особи P_s .

8) Выполнить оператор транспозиции на особи P_s .

9) Подсчитать функцию приспособленности получившейся особи P^t и поместить ее в популяцию на соответствующее место, исключив последнюю особь.

10) Увеличить номер текущей эпохи $t := t + 1$.

11) Если выполнилось условие остановки, то завершить работу, иначе переход на шаг 2.

В данном алгоритме используются операторы жадного кроссовера, мутации и транспозиции.

Оператор жадного кроссовера выполняется следующим образом:

1) Выбрать случайное натуральное число из $[1, n]$.

2) В особях $P_i = (x_1^i, \dots, x_n^i), P_j = (x_1^j, \dots, x_n^j)$ установить «точку разреза» слева от соответствующего разряда.

3) Вычислить значение функции приспособленности для левых и правых частей обеих особей.

4) Скопировать в новую особь левую и правую части с большими значениями.

5) Завершить работу.

Оператор транспозиции выполняется следующим образом:

1) Выбрать три натуральных числа $n_1, n_2, n_3 \in [1, n]$ так, чтобы $n_1 < n_2 < n_3$.

- 2) Установить три точки разреза слева от соответствующих разрядов.
- 3) Блок генов между вторым и третьим разрезами вставляется на место первого разреза с обратным порядком следования.
- 4) Завершить работу.

4. Сравнительный анализ эффективности алгоритмов

При выборе алгоритма необходимо руководствоваться критериями, которые позволяют оценить точные и приближенные методы. Если крайне критично время выполнения и ответ не обязательно нужен лучший (достаточно точен некоторый ответ, удовлетворяющий условиям и не последний по стоимости), то можно использовать жадный алгоритм. Перебором можно воспользоваться тогда, когда нужно оперативное решение задачи. Если есть немного больше времени, можно улучшить перебор, превратив его в метод ветвей и границ. Если есть ещё больше времени и критично время выполнения, то оптимально динамическое программирование. Если же есть время сейчас, но в будущем нужно будет искать ответ в определённых временных рамках, то удобен генетический алгоритм [6].

Критерии оценки эффективности для генетических алгоритмов подобрать сложнее, поскольку они изначально ограничены по времени работы (количеству итераций), а также используют множество случайных величин, которые от запуска к запуску могут давать разные результаты. Для того чтобы оценить эффективность работы генетических алгоритмов, требуются статистические данные, полученные при многократном запуске алгоритма с одинаковыми входными данными. На их основе предлагается рассмотреть следующие критерии:

- 1) Процент найденных глобальных экстремумов.
- 2) Среднее расстояние до глобального экстремума.
- 3) Среднеквадратическое отклонение от среднего расстояния до глобального экстремума.

Для того чтобы эти критерии дали достаточно точные результаты, запуск проводился 20 раз на двух наборах стартовых данных разных размеров. Статистические данные к каждому из двух примеров приведены в таблицах. В них приводятся результаты работы репродуктивного плана Холланда (РПХ) и жадного алгоритма с элитизмом (ЖА). Рассмотрим применение реализованных методов на некоторых задачах.

1. Взят условный набор заявок с вместимостью рюкзака 80 и следующими показателями затрат и эффективности.

Стоимость	15	20	25	22	20
Эффективность	100	120	150	130	90

Точное решение задачи – это набор заявок со стоимостями 15, 20, 25, 20. Их суммарная эффективность равна 460.

Жадный алгоритм, как и эвристический метод, при данном наборе дадут точное решение, и на небольших входных данных они работают оди-

наково быстро. Например, жадный алгоритм поместит в рюкзак те заявки, которые не превысят заданный бюджет. Четвертую заявку со стоимостью 22 он пропустит, иначе бюджет станет равным 82.

Покажем, что эвристический метод в других случаях дает лучшее решение, чем жадный.

Бюджет	Метод	Решение	Суммарная эффективность	Неиспользуемый бюджет
80	Точный	(1,1,1,0,1)	460	0
	Жадный	(1,1,1,0,1)	460	0
	Эвристический	(1,1,1,0,1)	460	0
62	Жадный	(1,1,1,0,0)	370	2
	Эвристический	(1,0,1,1,0)	380	0
45	Жадный	(1,1,0,0,0)	220	10
	Эвристический	(1,0,1,0,0)	250	5

Заметим, что если сократить бюджет, то при бюджете 62 жадный алгоритм позволяет найти решение, суммарная эффективность которого равна 370. Эвристический алгоритм, благодаря дополнительным условиям и проверкам, поместит в рюкзак заявки с суммарной эффективностью 380. Аналогично можно рассмотреть данную задачу в случае бюджета, равного 45. Жадный алгоритм даст суммарную эффективность 220, эвристический – 250.

Рассмотрим результаты работы генетических алгоритмов. Аналогично, размер популяции был ограничен 5 особями, а максимальное количество итераций 5 и 10 для репродуктивного плана Холланда и жадного алгоритма с элитизмом соответственно.

Результаты двадцати запусков имеют следующий вид:

№	Вид решения		Целевая функция		Расстояние до глобального экстремума		Суммарный вес	
	РПХ	ЖА	РПХ	ЖА	РПХ	ЖА	РПХ	ЖА
1	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
2	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
3	(1 1 1 0 1)	(1 0 1 1 0)	460	380	0	80	80	62
4	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
5	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
6	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
7	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
8	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
9	(1 1 1 0 1)	(1 1 0 1 1)	460	440	0	20	80	77
10	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
11	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
12	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
13	(1 1 0 1 1)	(1 1 1 0 1)	440	460	20	0	77	80
14	(0 1 1 1 0)	(1 1 1 0 1)	400	460	60	0	67	80
15	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
16	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
17	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
18	(0 1 1 1 0)	(1 1 1 0 1)	400	460	60	0	67	80
19	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80
20	(1 1 1 0 1)	(1 1 1 0 1)	460	460	0	0	80	80

Таким образом, при первом наборе стартовых данных критерии оценки эффективности каждого из алгоритмов принимают значения [9]:

Репродуктивный план Холланда:

- 1) Процент найденных глобальных экстремумов = 85%.
- 2) Среднее расстояние до глобального экстремума = 7 (1,522%).
- 3) Среднеквадратическое отклонение от глобального экстремума = 19,493 (5,131%).

Жадный алгоритм с элитизмом:

- 1) Процент найденных глобальных экстремумов равен 90%.
- 2) Среднее расстояние до глобального экстремума равно 7 (1,522%).
- 3) Среднеквадратическое отклонение от глобального экстремума равно 18,439 (5,423%)

2. Взят условный набор заявок с вместимостью рюкзака 120 и следующими показателями затрат и эффективности.

Стоимость	38	17	23	4	31	35	18	21	30	20
Эффективность	200	96	121	20	170	220	85	105	182	100

Результаты жадного и эвристического метода представлены в таблице:

Бюджет	Метод	Решение	Суммарная эффективность	Неиспользуемый бюджет
120	Точный	(1100010010)	698	0
	Жадный	(0101110010)	688	3
	Эвристический	(0001110011)	692	0

Несмотря на то, что эвристический метод не дает точного решения, он максимально близок к получению наилучшей эффективности, чего нельзя сказать о жадном алгоритме, который не позволяет потратить весь заданный бюджет.

Рассмотрим результаты работы генетических алгоритмов. Размер популяции был ограничен 12 особями, а максимальное количество итераций 15 и 50 для репродуктивного плана Холланда и жадного алгоритма с элитизмом соответственно.

Результаты двадцати запусков имеют следующий вид:

№	Вид решения		Целевая функция		Расстояние до глобального экстремума		Суммарный вес	
	РПХ	ЖА	РПХ	ЖА	РПХ	ЖА	РПХ	ЖА
1	(1100010010)	(1100010010)	698	698	0	0	120	120
2	(1100010010)	(1100010010)	698	698	0	0	120	120
3	(1100010010)	(0010110010)	698	693	0	5	120	119
4	(1011010001)	(0101110010)	661	688	37	10	120	117
5	(1100010010)	(1100010010)	698	698	0	0	120	120
6	(1100010010)	(1100010010)	698	698	0	0	120	120
7	(1011010001)	(0010110010)	661	693	37	5	120	119
8	(1100010010)	(0101110010)	698	688	0	10	120	117
9	(0001110011)	(0010110010)	692	693	6	5	120	119
10	(1100010010)	(1100010010)	698	698	0	0	120	120
11	(1100010010)	(1100010010)	698	698	0	0	120	120

№	Вид решения		Целевая функция		Расстояние до глобального экстремума		Суммарный вес	
	12	(0010110010)	(1100010010)	693	698	5	0	119
13	(1100010010)	(1100010010)	698	698	0	0	120	120
14	(0001110011)	(0000110110)	692	677	6	21	120	117
15	(1100010010)	(1101100010)	698	668	0	30	120	120
16	(0001110011)	(0010110010)	692	693	6	5	120	119
17	(0100011011)	(0101110010)	683	688	15	10	120	117
18	(1100010010)	(1100010010)	698	698	0	0	120	120
19	(1100010010)	(0001110011)	698	692	0	6	120	120
20	(0001110011)	(1100010010)	692	698	6	0	120	120

Таким образом, при втором наборе стартовых данных критерии оценки эффективности каждого из алгоритмов принимают значения:

Репродуктивный план Холланда:

- 1) Процент найденных глобальных экстремумов равен 60%.
- 2) Среднее расстояние до глобального экстремума равно 5,9 (0,845%).
- 3) Среднеквадратическое отклонение от глобального экстремума равно 12,514 (1,793%).

Жадный алгоритм с элитизмом:

- 1) Процент найденных глобальных экстремумов равен 50%.
- 2) Среднее расстояние до глобального экстремума равно 5,35 (0,773%).
- 3) Среднеквадратическое отклонение от глобального экстремума равно 9,426 (5,423%)

Подводя итоги, полученные результаты можно представить в виде сравнительной таблицы [6, 12]:

Метод	Тип алгоритма	Плюсы	Минусы	Сложность	Среднее отклонение от экстремума
Жадный алгоритм	Приближенный	Высокая скорость; может работать с большими значениями n ; простота реализации	Неточное решение	$O(n \cdot \log(n))$	10
Эвристический метод	Приближенный	Высокая скорость; может работать с большими значениями n ; возможность доработки алгоритма	Неточное решение	Зависит от набора данных	6
Полный перебор	Точный	Простота реализации; точное решение	Небольшие входные данные; временная сложность	$O(n!)$	0
Метод ветвей и границ	Точный	Возможно значительное сокращение времени	Работает как полный перебор	$O(n!)$	0
Метод динамического программирования	Точный	Независимость от вида исходных данных; точное решение	Большой объем вычислительной работы	$O(B \cdot n)$	0
Репродуктивный план Холланда	Приближенный	Устойчивость к предварительной сходимости;	Большие затраты оперативной памяти	Алгоритм ограничен по времени	5.9
Жадный алгоритм с элитизмом	Приближенный	Высокая скорость; решения близки к оптимальным	Неустойчивость к предварительной сходимости;	Алгоритм ограничен по времени	5.35

5. Заключение

Любая задача оптимизации с конечным множеством допустимых решений может быть решена методом полного перебора [10]. Однако на практике этот метод применим только для решений задач малой размерности. Часто при организации переборных алгоритмов решения задача разбивается на подзадачи, каждая из которых решается отдельно. Эти подзадачи снова разбиваются на подзадачи и так далее, пока не будут получены подзадачи либо с малым временем решения, либо с единственным допустимым решением. Для некоторых подзадач удается использовать специальные оценки и установить, что их решение не будет лучше некоторого «рекорда». Такие подзадачи исключаются, что позволяет сократить время работы алгоритма. Методы, основанные на подобной идее, называются методами ветвей и границ.

Если в процессе решения задачи встречаются одни и те же подзадачи, то есть смысл решение каждой из них записывать в отдельную таблицу и брать значения из нее. Если получается на основе оптимальных решений подзадач построить оптимальное решение задачи в целом, то следует применять метод динамического программирования.

Для многих задач используется жадный алгоритм. Начинается процесс с пустого решения, а затем на каждом шаге происходит улучшение решения – вносятся наиболее эффективные фрагменты и впоследствии не удаляются. Такие алгоритмы обычно выполняются быстрее алгоритмов, основанных на динамическом программировании, и гораздо быстрее, чем полный перебор. Выбор того или иного метода зависит от ряда условий. Каждый метод имеет определенные недостатки. Очевидный недостаток алгоритма метода ветвей и границ при решении задач большой размерности – необходимость перебрать слишком большое количество вариантов перед тем, как будет найден оптимальный [12]. Жадный алгоритм однокритериальный, поэтому он неприменим для многомерного рюкзака. Однако следует отметить, что для задачи непрерывного рюкзака именно этот алгоритм находит оптимальное решение. Генетический алгоритм – один из самых быстрых, но его хромосомы кодируются дискретно. Соответственно, его нельзя использовать для решения непрерывных и бесконечных задач. Генетический алгоритм также 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. Но при этом, если учесть некоторые возможные варианты неоптимального результата данным методом, генетический алгоритм, как правило, дает хороший результат в кратчайшие сроки.

В работе предложены специальные критерии для сравнения алго-

ритмов. Полученные статистические и аналитические данные позволяют сделать вывод о том, что наиболее быстрым среди точных классических методов оптимизации является метод динамического программирования, в то время как жадный алгоритм дает менее точное, но намного более быстрое решение. Из двух приведенных выше генетических алгоритмов репродуктивный план Холланда чаще достигает глобального экстремума, однако остальные его решения менее точны, чем у жадного генетического алгоритма, что делает среднее отклонение более высоким.

Список источников

1. Гладков Л.А., Курейчик В.В., Курейчик В.М. *Генетические алгоритмы*. Москва, 2006.
2. Емельянов В.В., Курейчик В.В., Курейчик В.М. *Теория и практика эволюционного моделирования*. Москва, Физматлит, 2003.
3. Сигал И.Х., Иванова А.П. *Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы*. Москва, 2007.
4. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. *Алгоритмы: построение и анализ*. Москва, Вильямс, 2013.
5. Скобцов Ю.А. *Основы эволюционных вычислений*. Донецк, ДонНТУ, 2008.
6. Студенческие конференции РИНЦ: НаучФорум. Доступно: <http://nauchforum.ru/node/6006>. (дата обращения: 23.05.2016)
7. Технологии анализа данных: BaseGroupLabs. Доступно: <https://basegroup.ru/community/articles/ga-math>. (дата обращения: 23.05.2016)
8. Портал искусственного интеллекта: AIportal. Доступно: <http://www.aiportal.ru/articles/genetic-algorithms/basic-concepts.html>. (дата обращения: 23.05.2016)
9. Звонков В.Б. *Сравнительное исследование генетических алгоритмов и стайного алгоритма оптимизации*. В кн.: XIV Международная конференция «Проблемы управления и моделирования в сложных системах», Самара, 2012: материалы. Самара, Институт проблем управления сложными системами Российской академии наук (ИПУСС РАН), 2012, с. 786-797.
10. Методические указания – Обзор методов разработки алгоритмов. Доступно: <http://www.twirpx.com/file/983082>. (дата обращения: 23.05.2016)
11. Задача о рюкзаке: динамическое программирование. Доступно: http://discopal.ispras.ru/Задача_о_рюкзаке: динамическое_программирование. (дата обращения: 23.05.2016)
12. Додонова М.М. Изучение и сравнение методов решения различных модификаций задачи о рюкзаке. Сибирский федеральный университет, ИМиФИ. Доступно: <http://www.lib.tpu.ru/fulltext/c/2014/C21/198.pdf>. (дата обращения: 23.05.2015)

OPTIMIZATION OF THE QUANTITY OF APPLICATIONS WITH A GIVEN BUDGET OF 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 paper is devoted to the problem of selection of the innovative projects applications with a given budget. *Discussion:* the problem is formalized as a knapsack one and solved by methods of discrete optimizations and genetic algorithms. *Results:* we implemented some algorithms of discrete optimization and genetic algorithms of solution of the knapsack problem and analyzed its comparative efficiency by several criteria in test examples.

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

References

1. Gladkov L.A., Kureychik V.V., Kureychik V.M. *Geneticheskiye algoritmy*. [Genetic algorithms]. Moscow, 2006. (In Russ.)
2. Yemelyanov V.V., Kureychik V.V., Kureychik V.M. *Teoriya i praktika evolyutsionnogo modelirovaniya*. [Theory and practice of evolutionary modeling]. Moscow, 2003. (In Russ.)
3. Seagal I.K., Ivanova A.P. *Vvedenie v prikladnoe diskretnoye programmirovaniye: modeli vichislitelniye algoritmi*. [Introduction to applied discrete programming: models and computational algorithms]. Moscow, 2007. (In Russ.)
4. Cormen T.K., Leiserson C.I., Rivest R.L., Stein C. *Introduction to Algorithms*. Williams, 2013.
5. Skobtsov Y.A. *Osnovi evolyutsionnikh vichisleniy*. [Basics of evolutionary computation]. Donetsk, 2008. (In Russ.)
6. Studencheskiye konferentsii RINTS: NauchForum. [Student conferences RSCIScienceForum]. Available at: <http://nauchforum.ru/node/6006>. (accessed: 23.05.2016)
7. Tekhnologii analiza dannikh: BaseGroup Labs [Data analysis technologies: BaseGroupLabs]. Available at: <https://basegroup.ru/community/articles/gamath>. (accessed: 23.05.2016)
8. Portal iskusstvennogo intellekta AIportal. [Portal of artificial intelligence: AIportal]. Available at: <http://www.aiportal.ru/articles/genetic-algorithms/basic-concepts.html>. (accessed: 23.05.2016)
9. Zvonkov V.B. *Sravnitel'noye issledovaniye geneticheskikh algoritmov i staynogo geneticheskogo algoritma*. [A comparative research of genetic algorithms and optimization algorithm of schooling]. XIV International conference «Problems of management and simulation of complex systems», Samara, 2012: materials. ICCS, 2012, pp. 786–797. (In Russ.)
10. Metodicheskie ukazaniya – Obzor metodov razrabotki algoritmov. [Guidelines – Review of algorithm design techniques]. Available at: <http://www.twirpx.com/file/983082>. (accessed: 23.05.2016)

11. Zadacha o ryukzake: dinamicheskoye programmirovaniye. [Knapsack problem: dynamic programming]. Available at: http://discopal.ispras.ru/Задача_о_рюкзаке:динамическое_программирование. (accessed: 23.05.2016)

12. Dodonova M.M. *Izuchenie I sravneniye*

metodov resheniya razlichnikh modifikatsiy zadachi o ryukzake. [The study and comparison of methods for solving various modifications of the knapsack problem]. Siberian Federal University, IMFI. Available at: <http://www.lib.tpu.ru/fulltext/c/2014/C21/198.pdf>. (accessed: 23.05.2016)