Многомерная задача о рюкзаке: эффективный метод решения и возможные приложения

В.В. Топка[1]

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

Аннотация. В последнее время возникает настоятельная потребность в разработке эффективного модельного и алгоритмического аппарата для верхнего уровня системы проектного управления, на котором должна быть подсистема отбора и управления портфелем проектов, подлежащих реализации. В данной работе рассмотрен оптимальный отбор портфеля проектов и распределение ресурсов на его выполнение в виде задачи дискретной оптимизации – многомерной задачи о рюкзаке. В статье приводятся известные теоретические результаты для жадного алгоритма решения задачи об одномерном рюкзаке; необходимые и достаточные условия оптимальности; оценка погрешности алгоритма и его асимптотическая погрешность для поведения в среднем. Предлагается прямой жадный алгоритм по удельной полезности решения многомерной задачи о рюкзаке. Для повышения его эффективности применяется локальный ограниченный перебор, улучшающий жадное решение, а также дополнение и локальная оптимизация предыдущих этапов. Жадные алгоритмы являются быстрыми, с полиномиальной временной сложностью.

Ключевые слова: *задача о рюкзаке, многомерная задача о рюкзаке, портфель проектов, улучшенный жадный алгоритм, локальный ограниченный перебор, полиномиальная временная сложность.*
DOI: 10.14357/20790279190206

Введение

Все используемые в настоящее время в нашей стране методы и модели управления проектами предназначены для уровня исполнителей: руководители проекта, управляющая команда, профильные специалисты, снабженные всеми необходимыми ресурсами. Тогда как для верхних уровней управления экономикой – уровень отраслей, госкорпораций, холдингов – соответствующие модели и методы проектного управления практически не разработаны. Но это уровень принятия стратегических решений, от него зависит около 50% успеха проектной деятельности, именно на нем сосредоточены все ресурсы и принимаются важнейшие решения. В [1, с.3] отмечается, что количество неуспешных проектов по отношению к успешным достигает в нашей стране, по разным оценкам, от 40 до 60%. Поэтому, одна из основных причин неуспешности проектного управления состоит в том, что верхние уровни управления слабо вовлечены в эту деятельность, а современная методология и технология проектного управления не учитывает в должной мере их интересы.

За рубежом необходимость проектного управления исследованиями и разработками, и, в частности, его верхнего уровня осознана достаточно давно. Об этом свидетельствует создание и эксплуатация таких систем, как *PATTERN* [2], *QUEST* [3], *TORQUE* [4], *PPB* [5], *PROFILE* [6], созданных в США в конце 60-х годов. Все эти системы основаны на представлении объекта планирования в виде дерева «цели-задачи» характеризующихся различными параметрами, значения которых устанавливаются путем обработки экспертных оценок. Отличительной особенностью указанных комплексных схем планирования является использование в рамках одной модели как числовых величин (например, годовые затраты на тему), так и экспертных оценок количественных показателей. Однако это обстоятельство не препятствует широкому использованию методов указанного класса для подготовки данных при принятии решений на верхнем уровне управления по осуществлению инвестиционной политики и финансировании крупных инновационных проектов.

В 1978 г. в нашей стране для таких целей в качестве аналитического инструмента был предложен метод иерархических схем в программно-целевом планировании исследований и разработок [7]. Достоинством этого метода является хорошо разработанный аппарат использования экспертных оценок, в котором лицо, принимающее решение (ЛПР), назначает план, самостоятельно указывает критерии и дает по ним субъективные нечисловые оценки, а также корректирует полученный план. В данной методике двоичный характер оценок шкал критериев и простая структура решающего правила предоставляют ЛПР инструмент распределения ограниченных ресурсов в соответствии с заданными целями для интерактивной корректировки исходной инвестиционной политики. Однако при этом возникают ситуации, когда предлагаемое решение приводит к необходимости отказаться от финансирования некоторых низкоприоритетных проектов и изменить сложившуюся организационную структуру.

Таким образом, возникает потребность в разработке эффективного модельного и алгоритмического аппарата для верхнего уровня такой системы проектного управления, на котором должна быть подсистема отбора и управления портфелем проектов, подлежащих реализации. В данной работе рассмотрен оптимальный отбор портфеля проектов и распределение ресурсов на его выполнение в виде задачи дискретной оптимизации – многомерной задачи о рюкзаке.

В настоящее время, несмотря на технологическое и научное продвижение, в решении NP-полной задачи о рюкзаке, решение большинства таких задач размерности, взятой из реальной жизни, остается проблемой. В [8 и 9] описаны основные методы, используемые для различных вариантов задачи о рюкзаке, и приведена их классификация по типу задачи и подходу к решению. Эти методы охватывают широкий спектр процедур исследования операций: релаксации, которые обеспечивают тесные границы, полиномиальные аппроксимации, дерево поиска ветвей и границ, динамическое программирование, метаэвристики и условная оптимизация. Некоторые методы стремятся к гибридизации этих процедур. В то время как другие воспользуются более быстрыми, с большей памятью многопроцессорными вычислительными системами и применяют готовые решатели, чтобы браться за релаксации задач или применяют параллельные алгоритмы, как для классической кластерной архитектуры, так и для грид-вычислений. Несмотря на это, большинство из алгоритмов более специального назначения, не могут быть легко расширены или адаптированы для задач общего вида. Самое главное, их полезность часто оспаривается в средах, где размерности постоянно возрастают.

1. Некоторые известные результаты для задачи об одномерном рюкзаке

Жадный алгоритм впервые был предложен в [10] для решения задачи о неограниченном рюкзаке. Для решения задачи жадным алгоритмом необходимо отсортировать n проектов по их удельной полезности (ценности, доходу) (то есть отношению полезности проекта к его стоимости), и поместить в рюкзак проекты с наибольшей удельной полезностью. В [8] говорится, что время работы данного алгоритма складывается из времени сортировки и времени укладки. Сложность сортировки элементов составляет $O(n\log n)$. Далее происходит вычисление того, сколько элементов поместится в рюкзак за общее время $O(n)$. Итоговая сложность $O(n\log n)$ при необходимости сортировки и $O(n)$ при уже отсортированных данных.

Необходимые и достаточные условия оптимальности жадного алгоритма. Рассматривается целочисленная задача об одномерном рюкзаке [11]:

$$z=\sum_{j=1}^{n}c_jx_j\to\max_x,$$

$$\sum_{j=1}^{n}a_jx_j=b, \tag{1}$$

$$x_j\in Z_+,\quad j=1,\dots,n.$$

$$c_1,\dots,c_n\in Z_+,\ a_1,\dots,a_n,b\in N.$$

Идея жадного алгоритма по удельной полезности для задачи (1) состоит в последовательном отборе проектов с наибольшей удельной полезностью c_j/a_j (т.е. слева направо); при этом каждый раз проверяется допустимость текущего частичного решения. Такое решение $x^g=\{x_j^g\}$ называется *жадным* (англ. greedy) решением; соответствующее ему значение целевой функции обозначается через g.

Задачу о рюкзаке назовем «привлекательной», если жадное решение является оптимальным при любой правой части. Решение задачи «привлекательно» если оно является жадным и оптимальным одновременно. Будем считать, что параметры задачи переиндексированы по возрастанию:

$$\frac{c_j}{a_j}\le\frac{c_{j+1}}{a_{j+1}},\ j=1,2,\dots,n-1.$$

Тогда жадное решение имеет вид:

$$x_j^g=\left\lfloor\frac{b-\sum_{i=j+1}^{n}a_ix_i^g}{a_j}\right\rfloor,\quad j=n,n-1,\dots,1.$$

Счет ведется с конца. Жадный метод, идущий по точкам $\sum_{i=j}^{n} \mathrm{e}_i x_i^g,\ j = n,\ldots,1,$

где e_i - i-тый единичный вектор, есть по сути метод внутренней точки, улучшающий от точки к точки значение целевой функции. Обозначим оптимальное значение целевой функции и значение жадного решения, зависящие от параметров *n* и *b,* через $z^*(n,b)$ и $g(n,b)$ соответственно. Тогда

$$g(n,b) = \sum_{j=1}^{n} c_j x_j^g .$$

Т е о р е м а 1 [11]. *Предположим, что* $a_1 < a_2 < \ldots < a_n$ *и* $n \geq 2$. *Задача "привлекательна" тогда и только тогда, когда для каждого* $y \in \mathrm{Z}$, *такого что* $a_{n-1} + 1 \leq y \leq a_n + a_{n-1} - 1$ *имеем:* $z^*(n,y) = g(n,y)$. ■

Т е о р е м а 2 [11]. *Предположим, что* $a_1 < a_2 < \ldots < a_n$. *Пусть*

$$m = \min\left\{ y \in \mathrm{Z}_+ :\ z^*(n,y) \neq g(n,y) \right\}.$$

Определим индексы *k* и *p* следующим образом:

$$k = \max\left\{ j :\ x_j^g(n,m) > 0 \right\} \text{ и}$$

$$p = \max\left\{ j :\ \exists \mathrm{x}^*(n,m),\ \ x_j^*(n,m) > 0 \right\}.$$

Тогда есть два индекса *q* и *r*, таких что

$$p \geq q > r \geq 1, \quad x_r^g(p,a_k) > 0, \text{ и}$$

$$m = a_q + \sum_{j=q}^{p} a_j x_j^g(p,a_k). \ ■$$

Утверждение Теоремы 2 кажется более сильным, так как согласно ему, имеется только $\binom{n}{3}$ кандидатов для *m*, чей номер независим от величины коэффициентов.

Оценка погрешности жадного алгоритма. Здесь предполагается, что задача с *n* переменными "привлекательна". Добавим еще одну переменную и опишем случаи, когда задача остается "привлекательной". Предполагается, что индекс новой переменной *n*+1, а также

$$c_{n+1} \in \mathrm{Z}_+,\ \ a_{n+1} \in \mathrm{N},\ \ \frac{c_n}{a_n} \leq \frac{c_{n+1}}{a_{n+1}},\ \ a_n < a_{n+1}. \quad (2)$$

Пусть err(n,y) есть погрешность жадного решения в задаче с *n* переменными и правой частью *y*, т.е.

$$\mathrm{err(n,y)} = z^*(n,y) - g(n,y).$$

Т е о р е м а 3 [11]. *Предположим, что условия* (2) *выполняются и что задача при n переменных является "привлекательной" и рассмотрим задачу при n*+1 *переменных. Пусть i>0 – произвольное целое и* s_i *– целое, такое что*

$$i a_{n+1} \leq s_i a_n < (i+1) a_{n+1}.$$

Предположим, что для целочисленного *y* неравенство

$$i a_{n+1} \leq y < (i+1) a_{n+1}$$

выполняется и значение x_{n+1}^* *есть* 0 *в задаче при n*+1 *переменных и правой части y. Тогда*

$$\mathrm{err(n{+}1,\ s_i}a_n) \geq \mathrm{err(n{+}1,\ y)}. \ ■$$

Выше были описаны случаи, когда после добавления новой переменной к задаче она остается «привлекательной». Далее определяется ее максимальная погрешность, когда свойство «привлекательности» не выполняется.

Т е о р е м а 4 [11]. *Предположим, что условия* (2) *выполняются и функции* $z(n,\bullet)$ и $g(n,\bullet)$ *– идентичны. Пусть* $\mathrm{d=g.c.d.}(\mathrm{a_n},\ \mathrm{a_{n+1}})$ (от англ. *greatest* ***c****ommon* ***d****ivisor*) – наибольший общий делитель *и*

$$s_i = \left\lceil \frac{i a_{n+1}}{a_n} \right\rceil,\ \ t_i = s_i a_n - i a_{n+1},\quad i \in \mathrm{N}.$$

Тогда

$$\max\left\{ \mathrm{err(n{+}1,\ y)}:\ \mathrm{y} \in \mathrm{Z}_+ \right\} =$$

$$= \max\left\{ \max\left\{ 0,\ (s_i c_n - i c_{n+1} - g(n,t_i)) \right\} :\ i = 1,\ldots,(a_n/d)-1 \right\}. \ ■$$

Поведение в среднем жадных алгоритмов. Рассматривается частный случай многомерной задачи о рюкзаке – одномерная задача о рюкзаке с булевыми переменными. Она состоит в нахождении величины

$$z^* = \left\{ \max_x \sum_{j=1}^{n} c_j x_j \ \middle|\ \sum_{j=1}^{n} a_j x_j \leq b,\ \ x \in \{0,1\}^n \right\}. \quad (3)$$

Все коэффициенты в (3) положительны. Помимо этого будем предполагать, что

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \frac{c_j}{a_j} \geq \ldots \geq \frac{c_n}{a_n}, \quad (4)$$

т.е. что переменные занумерованы в порядке невозрастания их «удельных полезностей» c_j / a_j. Условие (4) обычно называется *условием регулярности*. В [12] говорится: «Среди приближенных методов ведущее место занимают жадные алгоритмы, применяемые как самостоятельно, так и в сочетании с другими подходами. Жадные методы можно трактовать как дискретные аналоги градиентных методов. Их несомненное преимущество состоит в том, что в применении к задаче (3) их трудоемкость является линейной по n (при выполнении условия регулярности). Жадные методы не гарантируют получения оптимального решения, но для них удается дать теоретические нижние оценки гарантированного функционирования, т.е. отклонения приближенного решения от оптимума». (См. обзор [13]). Идея, в определенном смысле обратная описанной ранее, состоит в последовательном отбрасывании из множества $\{1,...,n\}$ проектов с наименьшей удельной полезностью c_j / a_j до тех пор, пока суммарная стоимость остающихся проектов станет не больше b. В соответствии с принятой в теории оптимизации терминологией подобные алгоритмы естественно называть *двойственными*. Поэтому описанный выше жадный алгоритм будет далее называться *прямым*. Формально, двойственный жадный алгоритм начинает работу с недопустимого решения x=(1,…,1) и последовательно заменяет единицы нулями в порядке неубывания отношений c_j / a_j (т.е. справа налево); при этом каждый раз проверяется допустимость текущего частичного решения. Процесс завершается при получении первого допустимого решения. Это решение X^{dg} называется *двойственным жадным решением*; соответствующее ему значение целевой функции обозначается через g^d. Отметим, прежде всего, что прямое и двойственное жадные решения могут быть различными. Имеет место следующая «фольклорная теорема».

Утверждение [12]. *Двойственный жадный алгоритм для задачи* (3) *может быть произвольно плохим.*

Вместе с тем этот алгоритм успешно использовался, например, в методах декомпозиции для решения прикладных задач внедрения нововведений, где давал хорошие результаты. Это противоречие между теоретически сколь угодно плохим и практически хорошим поведением метода может быть, как обычно, разрешено путем *анализа поведения этого метода не в наихудших случаях, а в среднем.*

В таких ситуациях предполагается, что все коэффициенты c_j, a_j, j=1,…,n, – независимые случайные величины, равномерно распределенные на отрезке [0,1]. Обозначим y_j=c_j / a_j . Пусть y^k –случайная величина, равная (n-k+1) – му члену вариационного ряда, определяемого вектором случайных величин y_1,…,y_n, т.е. $y^1 \geq y^2 \geq \ldots \geq y^n$. Введем случайные величины c^k, a^k следующим образом. Если $y^k = c_{q(k)} / a_{q(k)}$, будем полагать $c^k = c_{q(k)}$, $a^k = a_{q(k)}$.

Представляет интерес поведение приближенных алгоритмов для задач с n переменными при неограниченном росте n. Пусть комбинаторная оптимизационная задача решается некоторым приближенным алгоритмом A, который дает значение целевой функции z^{A}. Будем обозначать этот алгоритм через A_n, чтобы подчеркнуть зависимость A от числа переменных задачи. Будем говорить, что алгоритм A_n имеет асимптотическую погрешность t, если

$$\mathrm{P}\left(z^* - z^{\mathrm{An}} \leq t\right) \underset{n\to\infty}{\to} 1,$$

где z^* – оптимальное значение целевой функции, t>0. В нашем конкретном случае рассматривается одномерная задача о рюкзаке с булевыми переменными, а в качестве A_n – жадные алгоритмы (прямой и двойственный). Справедлива следующая теорема.

Т е о р е м а 5. [12, с. 71]. *Если* $b = \lambda n$, *где* $\lambda > 1/2 - t/3$, *то прямой и двойственный жадные алгоритмы имеют асимптотическую погрешность t.* В условии Теоремы число ½ есть среднее равномерно распределенной случайной величины на [0,1]; коэффициент 1/3 при t также обусловлен равномерностью распределения.

Приближенная схема полностью полиномиального времени. Так как точного алгоритма решения задачи за полиномиальное время не было найдено, появился целый ряд приближенных схем полностью полиномиального времени, то есть со сложностью, являющейся полиномом от n и $\frac{1}{\varepsilon}$, при снижении точности, с которой заданы полезности (ценности) проектов. Объединяя проекты близкой ценности в одну группу, можно снизить количество разных проектов. Соответствующий алгоритм [9] выглядит следующим образом:

1. Найдем приближенное решение g при помощи жадного алгоритма. Пусть z^* — точное решение. Тогда из оценки эффективности жадного алгоритма $g \leq z^* \leq 2g$.
2. Отмасштабируем ценности следующим образом: $c_j^{/} = \left\lfloor \dfrac{c_j}{z\varepsilon / n} \right\rfloor$.
3. Алгоритмом динамического программирования для задачи с целочисленными ценностями проектов находим решение.

Существует множество различных схем аппроксимации, пригодных для задачи об одномерном рюкзаке. Среди других методов решения задачи (3) отметим метод динамического программирования, в котором для задачи включающей n элементов, число рассматриваемых подмножеств равно $O(2^n)$. Следовательно, динамическое программирование обычно приводит к экспоненциальной сложности.

Однако если рассматриваемая проблема является NP-трудной в обычном смысле, возможно, получить псевдополиномиальные алгоритмы [9,14].

В [15] рассматривается модификация алгоритма динамического программирования, который называется графическим алгоритмом. Для задачи об одномерном рюкзаке показано, что временная сложность (псевдополиномиальная) графического алгоритма меньше, чем временная сложность алгоритма динамического программирования. Более того, время работы графического алгоритма часто существенно уменьшается так же, как и количество состояний, которые должны рассматриваться на каждом этапе алгоритма динамического программирования. Кроме того, в отличие от классического динамического программирования, он также может решать задачи с нецелыми исходными данными, без необходимых преобразований соответствующей задачи.

2. Многомерная задача о рюкзаке

Многомерная задача о $\{0,1\}$ - рюкзаке (МЗР) является частным случаем общих линейных булевых $\{0,1\}$ - задач. Для МЗР были использованы в литературе несколько наименований: задача о *m*-мерном рюкзаке, многомерная задача о рюкзаке, задача о мультирюкзаке, $\{0,1\}$ - задача о рюкзаке со многими ограничениями и т. д. Для обозначения данной задачи выбрано название, введенное в [16], которое однозначно говорит, что МЗР является обобщением стандартной задачи о $\{0,1\}$ - рюкзаке $(m=1)$. Исторически сложилось, что первые образцы задачи были представлены в [17] и [18] в качестве модели бюджетирования капитала. В основном, МЗР представляет собой модель распределения ресурсов, которая может быть сформулирована следующим образом:

$$z=\sum_{j=1}^{n} c_j x_j \to \max_x, \tag{5}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i=1:m, \tag{6}$$

$$x_j \in \{0,1\}, \quad j=1:n. \tag{7}$$

где n – есть число проектов и m – количество ограничений по ресурсам в рюкзаке с объемом b_i, i=1:m. Каждый проект $j=1:n$ потребляет a_{ij} единиц ресурса в i-том рюкзаке i=1:m и дает c_j единиц полезности (ценности, дохода) при включении его в состав рюкзака.

Цель состоит в том, чтобы найти подмножество проектов, которые дают максимальную полезность (ценность, доход), не превышая ограничений по заданным объемам ресурсов. По своей природе, все параметры задачи являются неотрицательными. Точнее, предполагается, без ограничения общности, что

$$c_j>0,\ b_i>0,\ 0\le a_{ij}\le b_i \text{ и}$$

$$\sum_{j=1}^{n} a_{ij} > b_i \quad j=1:n, \quad i=1:m,$$

(так как в противном случае, некоторые или все переменные могут быть зафиксированы как 0 или 1). Кроме того, любая МЗР имеющая, по меньшей мере, один из этих параметров $a_{ij}=0$, может быть заменена эквивалентной МЗР с положительным параметром, т.е. обе задачи имеют одинаковые допустимые решения.

Эффективность метода получения оценок. В [19, с. 10] говорится, что «жадные, в качестве альтернативы называемые «близорукими» или «конструктивными/деструктивными», являются быстрыми алгоритмами (полиномиальная временная сложность), и в целом просты в реализации». Среди новых методов решения рюкзачных задач можно выделить следующие. В [20] предлагается эвристический аппроксимационный подход к решению многокритериальной блочной задаче о рюкзаке. В [21] задача выбора множества проектов при выпуске инновационной продукции сведена к решению конечного числа задач о рюкзаке. В [22] предложен метод решения многомерной задачи о ранце, в основе которого лежит построение множества вариантов и сокращение их числа путем «склеивания» пар вариантов таким образом, что все допустимые решения сохраняются. В результате получается верхняя оценка задачи. В [23] дано описание *подхода для получения оценок* в задачах дискретной оптимизации, названного методом сетевого программирования [24], в основу которого положена возможность представления функции многих переменных в виде суперпозиции более простых функций. Структура такой суперпозиции представляется в виде сети, входы которой соответствуют переменным, а выходы – функции. Показано, что если сеть является деревом, то решение задачи сводится к последовательному решению более простых задач. В общем случае предложено преобразовать сеть в дерево путем разделения вершин сети. Доказано, что решение задачи для преобразованной структуры дает *верхнюю оценку* для целевой функции исходной задачи целочисленного линейного программирования (если решается задача максимизации).

Метод ветвей и границ является вариацией метода полного перебора с той разницей, что мы исключаем заведомо неоптимальные ветви дерева полного перебора. Как и метод полного перебора, он позволяет найти оптимальное решение и поэтому относится к точным алгоритмам. В [8, с. 29, 50] говорится, что в алгоритмах типа ветвей и границ, сортируются проекты по их удельной полезности (отношению полезности к стоимости) и строится дерево полного перебора. Улучшение заключается в том, что в процессе построения дерева, для каждого узла *оценивается верхняя граница* полезности решения, и продолжается построение дерева только для узла с максимальной оценкой. Когда максимальная верхняя граница оказывается в листе дерева, алгоритм заканчивает свою работу. Способность метода ветвей и границ уменьшать количество вариантов перебора сильно основана на структуре входных данных. Его целесообразно применять только в том случае, когда удельные полезности проектов отличаются значительно.

Известно [25, с. 54], что для задачи об одномерном рюкзаке, алгоритмы ветвей и границ с полиномиальными алгоритмами расчета верхних и нижних оценок для числа вершин в дереве ветвления, т.е. числа решаемых оценочных линейных задач, которое считается *оценкой эффективности метода*, имеют асимптотику, близкую к $2^{n+3/2}/\sqrt{\pi(n+1)}$, т.е. $2^{O(n)}$ операций, где n – количество элементов. В МЗР для поиска ε – оптимального решения оценка числа ветвлений, полученная в [25, с. 156] имеет вид, $2^{m/\lambda\varepsilon}$, где m – число ограничений, ε – заданная точность, $\lambda = f(x')/\sum_{j=1}^{n} c_j$, где x' – вектор (1, 0, …, 0) или любое допустимое решение, такое, что $f(x') \geq c_1$. Для решения МЗР применяется дихотомическое ветвление, при котором на каждом шаге решается не более двух оценочных задач. Общее число решенных оценочных задач для ε – оптимального алгоритма не превосходит $2^{1+m/\lambda\varepsilon}$.

В [26] разработан графический алгоритм точного решения и основанная на нем схема аппроксимации с полиномиальным временем работы для задачи об инвестициях, которая формулируется в виде задачи о рюкзаке в более общем виде – многомерной мультивыборной задачи о рюкзаке.

3. Прямой жадный алгоритм по удельной полезности с локальным перебором решения многомерной задачи о рюкзаке

В данном разделе предлагается жадный алгоритм решения многомерной задачи о рюкзаке, который в модельных расчетах показал высокую вычислительную эффективность. Структура алгоритма задается выделением следующих основных этапов.

Этап 1. *Усреднение величин ресурсов, расходуемых на выполнение отдельных проектов.* Существенную роль в усреднении играют величины d_i, определяемые для каждого i-го ресурса, следующим образом:

$$d_i = \left(\sum_{j=1}^{n} a_{ij}\right)/b_i, \quad i = 1:m. \tag{8}$$

Можно считать величину d_i оценкой дефицитности i-го ресурса (кстати, при $d_i \leq 1$ ресурс i вообще перестает быть дефицитным и ограничение по этому типу ресурса в (4) может быть опущено). Само усреднение величины расхода ресурса для каждого j-го проекта, может осуществляться двумя основными способами. В каждом из них усредненное значение $\overline{a_{ij}}$ величин a_{ij}, $i = 1:m$ определяется следующим образом:

$$\overline{a}_j^{(1)} = \sum_{i=1}^{m} (a_{ij} d_i)^k, \tag{9}$$

$$\overline{a}_j^{(2)} = \max_{1 \leq i \leq m} \{a_{ij} d_i\}, \tag{10}$$

где величина d_i берется из (8), k – целочисленный параметр, а индексы (1) и (2) у величины a_j соответствуют двум различным способам усреднения (очевидно, при достаточно большом значении k (9) переходит (10).

Этап 2. *Определение очередности включения проектов в план финансирования.*

В соответствии с традиционными жадными алгоритмами в рассматриваемом алгоритме очередность включения проектов задается последовательностью $j_1, j_2, \ldots, j_n$, определяемой соотношениями:

$$\frac{c_{j_1}}{\overline{a}_{j_1}} \geq \frac{c_{j_2}}{\overline{a}_{j_2}} \geq \ldots \geq \frac{c_{j_n}}{\overline{a}_{j_n}}, \tag{11}$$

где величины a_{j_k} взяты из (9) или (10).

В дальнейшем будем считать, что проекты переиндексированы в соответствии с подстановкой:

$$\begin{matrix} 1 & 2 & \ldots & n, \\ j_1 & j_2 & \ldots & j_n. \end{matrix}$$

поэтому вместо (11) имеем:

$$\frac{c_1}{\overline{a}_1} \geq \frac{c_2}{\overline{a}_2} \geq \ldots \geq \frac{c_n}{\overline{a}_n}. \tag{12}$$

Этап 3. *Определение максимального числа проектов, включаемых в план.*

Жадный алгоритм включает проекты в план в порядке монотонного возрастания их номеров.

Индекс n последнего включенного проекта определяется из соотношений:

$$\sum_{j=1}^{n} a_{ij} \le b_i, \quad \sum_{j=1}^{n+1} a_{i_1 j} > b_{i_1}, \ i, i_1 \in 1:m. \qquad (13)$$

<u>Этап 4</u>. *Локальный перебор решений с целью улучшения допустимого жадного решения.*

Реализация данного этапа [27] зависит от выбора значений двух параметров l_1 и l_2, где

$$0 \le l_1 \le s, \ \ 1 \le l_2 \le n - s,$$

(заметим, что зависимость от параметра уже возникла при реализации этапа 1 – см.(8)).

Параметры l_1 и l_2 задают размеры двух групп проектов соответственно с индексами (s- l_1+1, s-l_2+2,...,s) и (s+1,s+2,...s+l_2). Первая группа из l_1 проектов исключается из жадного решения, определенного на Этапе 3, и вместе со второй группой из l_2 проектов составляет подмножество, из которых выбирается наилучшая «добавка» к проектам 1, 2, ..., $s - l_1$, «оставшимся» от каждого решения. Задача определения наилучшей добавки по структуре совпадает с исходной задачей (5)-(7) и имеет следующий вид:

$$\sum_{j=s-l_1+1}^{s+l_2} c_j x_j \to \max_x, \qquad (14)$$

при ограничении (7) и ограничениях

$$\sum_{j=s-l_1+1}^{s+l_2} a_{ij} x_j \le \Delta b_i,$$

$$\text{где } \Delta b_i = b_i - \sum_{j=1}^{s-l_1} a_{ij}, \qquad (15)$$

причем, величины Δb_i из (15) в силу (13) неотрицательны.

Задача (14),(15), (7) решается точно, причем, точный алгоритм может быть при условии небольших значений параметров l_1 и l_2, чисто переборным.

Пусть полученное решение имеет вид:

$$x_j = 1, \ j \in J_L = \{j_1, j_2, ..., j_L\}; \ x_j = 0, \ j \overline{\in} J_L \quad (16)$$

и где $J_L \in [s - l_1 + 1, \ \ s + l_2], \ \ L \le l_1 + l_2.$

В дальнейшем, будем решение МЗР идентифицировать с подмножеством индексов переменных x_j, равных 1 (в (16) решение определяется подмножеством J_L). Тогда решение, получаемое в результате выполнения Этапов 3 и 4, определяется подмножеством

$$J' = \{1, \ 2, \ ..., \ s - l_1\} \cup J_L. \qquad (17)$$

<u>Этап 5</u>. *Дополнение решения предыдущих этапов.*

Дополнения $\bar{J}'$ решения J', описанного в (17), проводится за счет возможного включения в него некоторых проектов с индексами из интервала $[s + l_2 + 1, \ ..., \ n]$. Проверка возможности включения проектов проводится в порядке увеличения их индексов, начиная с индекса (s+l_2+1) и кончая индексом n. Если $\bar{J}'$ – дополненное к моменту очередной проверки исходное решение J', то положительный результат проверки определяется выполнением следующих условий:

$$b_i - \sum_{j \in \bar{J}'} a_{ij} \ge 0, \ \ i = 1:m.$$

Полученное по окончании этапа дополнительное решение обозначается через $\bar{J}''$.

<u>Этап 6</u>. *Локальная оптимизация решения предыдущих этапов.*

Реализация данного этапа, также как и реализация Этапов 1 и 4, зависит от выбора значения некоторого параметра l_0, где

$$l_0 \le \min\left\{n - \left|\bar{J}''\right|, \ \left|\bar{J}''\right|\right\}.$$

При фиксированном значении l_0 этап осуществляется следующим образом.

Во множестве $\bar{J}''$ последовательно просматриваются все подмножества J' из l_0 элементов, и для каждого такого подмножества J' во множестве $\{1, \ 2, \ ..., \ n\} / \bar{J}''$ находится подмножество индексов J'' из (l_0+1) элемента, для которого величина

$$c'' = \sum_{j \in J''} c_j \qquad (18)$$

максимальна. Далее проверяется условие:

$$c'' > \sum_{j \in J'} c_j. \qquad (19)$$

Каждый раз при выполнении условия (19) в ходе процедуры перебора множеств J' текущее решение переформировывается по формуле:

$$\bar{J}'' := (\bar{J}'' \cup J'') / J'. \qquad (20)$$

Очевидно, в силу (19) новое решение, определяемое в (20), лучше текущего решения. Заканчивается этап формированием текущего решения $\bar{J}''$, для которого не удается выделить ни одного подмножества J', удовлетворяющего (19).

Этап 6 может проводиться для различных значений целочисленного параметра l_0, последовательно увеличивающегося со значения l_0=1 до некоторого априори, задаваемого граничного значения $l_0 = \bar{l}_0$. ■

Заканчивая на этом общее описание алгоритма решения МЗР, сделаем ряд рекомендаций по методике его применения.

1) Выбирая на Этапе 1 различные способы усреднения, можно после каждого выбора провести все остальные этапы процедуры, приняв по окончании соответствующих «прогонов» рекордное решение за приближенное решение задачи. Можно рекомендовать использование второго способа усреднения, а также первого способа при двух-трех значениях параметра k, например, $k = 1$, $k = 2$, $k \sim 10$.
2) Значения параметров l_1 и l_2 на Этапе 4 рекомендуется выбирать так, чтобы они в сумме составляли величину порядка 10. Большие значения параметров l_1 и l_2 существенно увеличивают трудоемкость реализации соответствующего этапа алгоритма.
3) На Этапе 6 рекомендуется ограничивать рост параметра l_0, значениями 2…3, поскольку дальнейшее увеличение приводит к существенному росту объема вычислений и, как показывает экспериментальный опыт, не дает сопоставимого с этим ростом увеличения точности получаемого решения.

Поскольку реализации всех этапов рассмотренного алгоритма, за исключение Этапа 4, вполне легко воспроизводимы, то мы в следующем пункте остановимся на детальном описании реализации лишь Этапа 4.

Для краткости описания алгоритма ограниченного перебора, решающего задачу (14), (15), (7), проведем применительно к исходной формулировке (5)-(7), к которой первая задача может быть сведена путем переобозначений. Рассматриваемый алгоритм состоит из следующих шагов.

Шаг 1 (начальный). Положить $j_0 := n+1$, $c_1 := 0$, $c_0 := -R$, где R – достаточно большая положительная константа, а также $y_j := 0 \quad (j = 1:n)$, $B_i := b_i$.

Шаг 2. Положить $j_0 := j_0 - 1$. В случае $j_0 > 0$, перейти к следующему шагу, в противном случае к Шагу 8.

Шаг 3. Если $y_{j_0} = 0$ перейти к Шагу 5, в противном случае к следующему шагу.

Шаг 4. Положить $y_{j_0} := 1$, $c_1 := c_1 + c_{j_0}$, а также $B_i := B_i - a_{ij_0}, i = 1:m$.

Перейти к Шагу 2.

Шаг 5. Если хотя бы для одного i, где $i = 1:m$ выполняется условие $B_i < a_{ij_0}$, то перейти к Шагу 2, в противном случае перейти к следующему шагу.

Шаг 6. Положить

$$y_{j_0} := 1,\ c_1 := c_1 + c_{j_0},\ B_i := B_i - a_{ij_0},\ j_0 := n+1$$

и перейти к следующему шагу.

Шаг 7. Если $c_1 > c_0$ перейти к Шагу 2, в противном случае положить $c_0 := c_1$, $c_1 := 0$, $x_j := y_j$ ($j = 1:n$) и перейти к Шагу 2.

Шаг 8 (заключительный). Принять текущее решение x_j; $j = 1:n$, за оптимальное решение задачи и закончить процедуру. ■

Отметим, что перебор, осуществляемый в описанном алгоритме, не является полным, поскольку в нем на Шаге 5 происходит отсев решений, не удовлетворяющих ограничениям (12). Кстати, степень неполноты перебора тем выше, чем больше коэффициенты дефицитности d_i (см. (8)) отдельных видов ресурсов.

Заключение

Задача о рюкзаке является NP – полной. Но если зафиксировать верхнюю границу для коэффициентов целевой функции, то задача о рюкзаке разрешима за полиномиальное время. Соответствующий алгоритм основан на методе динамического программирования и дает «вполне полиномиальную аппроксимационную схему» для задачи о рюкзаке [28, с.392]. Хорошо известно, что полиномиальные аппроксимационные схемы существуют для обоих случаев m=1 и $m > 1$. Первая полиномиальная схема аппроксимации дана в [29] для задачи о {0,1}-рюкзаке. Случай $m > 1$, если m – фиксированная константа, анализировали в [30], а в [31] их метод расширен до МЗР.

Предложенный в работе прямой жадный алгоритм по удельной полезности с локальным ограниченным перебором, улучшающим жадное решение, а также с дополнением и локальной оптимизацией предыдущих этапов для решения многомерной задачи о рюкзаке, является жадной эвристикой, показавшей высокую вычислительную эффективность. Его преимущество состоит, как указано в обзоре [19, с. 10] для МЗР, в том, что «жадные являются быстрыми алгоритмами (полиномиальная временная сложность), и в целом просты в реализации».

Литература

1. *Гельруд Я.Д.* Методология создания информационно-аналитической системы управления проектами на основе комплекса математических моделей функционирования стейкхолдеров. Автореф. дис... д-ра техн. наук: 05.13.10 / Челябинск, (Юж.-Ур. гос. ун-т.), 2015. 43 с.
2. *Лопухин М.М.* ПАТТЕРН – метод планирования и прогнозирования научных работ. М.: Сов. радио, 1971.
3. *Cetron M.J.* QUEST Status Report // IEEE Transactions on Engineering Management. 1967. V. 14. №1. March. 443.

4. *Nutt A.B.* Testing TORQUE – a quantitative R&D research allocation system // IEEE Trans. on Eng. Manag. 1969.Vol. EM-16. №4.
5. *The Analysis and Evaluation* of Public Expenditures: the PPB System. Washington: U.S. Govern. Print Office, 1969.
6. *Янч Э.* Прогнозирование научно-технического прогресса. М.: Прогресс, 1970.
7. *Ларичев О.И., Бойченко В.С., Мошкович Е.Н., Шепталова Л.П.* Методы иерархических схем в программно-целевом планировании научных исследований / Препринт. М.: ВНИИСИ, 1978.
8. *Martello S., Toth P.* Knapsack problems: Algorithms and computer implementations / in: Series in Discrete Mathematics and Optimization. N. Y.: Wiley Inter Sciences, 1990. 296 p.
9. *Kellerer H., Pferschy U., Pisinger D.* Knapsack Problems. Berlin / Heidelberg: Springer-Verlag, 2004. 548 p.
10. *Dantzig G. B.* Discrete-Variable Extremum Problems // Oper. Res. (INFORMS), 1957. Vol. 5, No. 2. pp. 266–277.
11. *Vizvari B.* On the Optimality of the Greedy Solutions of the General Knapsack Problems // Optimization. 1992. Vol. 23. No.2. pp. 125-138.
12. *Дюбин Г.Н., Корбут А.А.* Жадные алгоритмы для задач о ранце: поведение в среднем // Сибирский журнал индустриальной математики. 1999. Т.2. №2(4). С.68-93.
13. *Korte B.* Kombinatorische Optimierung und algorithmische Prinzipien // Ökonomische Prognose-, Enstscheidungs- und Gleichgewichtsmodelle. Weinheim: VCH Verlagsgesellschaft, 1986. pp. 286-341.
14. *Гафаров Е.Р., Лазарев А.А., Вернер Ф.* Алгоритмы решения задач максимизации суммарного запаздывания и максимизации количества запаздывающих требований для одного прибора // АиТ. 2010. № 10. С. 63–79.
15. *Lazarev A., Salnikov A., Baranov A.* Graphical Algorithm for the Knapsack Problems // Lecture Notes in Computer Science. 2011. 6873. pp. 459-466.
16. *Weingartner H.M., Ness D.N.* Method for the solution for the multidimensional 0–1 knapsack problem // Operations Research. 1967. Vol. 15. pp. 83–103.
17. *Lorie J., Savage L.J.* Three problems in capital rationing // Journal of Business. 1955. Vol. 28. pp. 229–239.
18. *Manne A.S., Markowitz H.M.* On the solution of discrete programming problems // Econometrica. 1957. Vol. 25. pp. 84–110.
19. *Freville A.* The multidimensional 0–1 knapsack problem: An overview // European Journal of Operational Research. 2004. Vol. 155. pp. 1–21
20. *Левин М.Ш., Сафонов А.В.* Эвристический алгоритм для многокритериальной блочной задачи о рюкзаке // Искусственный интеллект и принятие решений. 2009. № 4. С. 53-64.
21. *Буркова И.В., Ткаченко М.В., Чураков П.П.* Задача выбора множества проектов при выпуске инновационной продукции // Вестник Воронежского государственного технического университета. 2010. Т. 6. №9. С. 120-124.
22. *Бурков В.Н., Буркова И.В., Уандыков Б.К., Чу Д.С.* Метод допустимых решений в многомерной задаче о ранце // Экономика и менеджмент систем управления. 2015. Т.18. № 4.1. С. 136-144.
23. *Буркова И.В., Пужанова Е.О., Марин О.Л.* Задача о ранце и ее модификации // Научный вестник Воронежского государственного архитектурно-строительного университета. Серия: Управление строительством. Воронеж: ВГАСУ, 2014. Вып. № 1(6). С. 103-112.
24. *Бурков В.Н., Буркова И.В., Попок М.В., Овчинникова Т.И.* Метод сетевого программирования // Проблемы управления. 2005. №3. С. 23 – 29.
25. *Финкельштейн Ю.Ю.* Приближенные методы и прикладные задачи дискретного программирования. М.: Наука, 1976. 264 с.
26. *Гафаров Е.Р., Долгий А., Лазарев А.А., Вернер Ф.* Новый эффективный алгоритм решения задачи об инвестициях // А и Т. 2016. №9. С. 150-166.
27. *Рубинштейн М.И., Топка, В.В.* Жадный алгоритм решения многомерной задачи о рюкзаке // Труды IV-ой Всесоюзной школы-семинара «Статистический и дискретный анализ данных и экспертное оценивание» (Одесса, 1991). ОдПИ,1991. С. 273-274.
28. *Схрейвер А.* Теория линейного и целочисленного программирования. В 2-х т. Т.2. / Пер. с англ. М.: Мир, 1991. 702 с. (Schrijver, A. Theory of Linear and Integer Programming. N.Y.: John Wiley & Sons Ltd, 1986. 702 p.)
29. *Shani S.* Approximate algorithms for the 0–1 knapsack problem // Journal of Association for Computing Machinery. 1975. Vol.22. pp. 115–124.
30. *Chandra A.K., Hirschberg D.S., Wong C.K.* Approximation algorithms for some generalized knapsack problems // Theoretical Computer Sciences. 1976. No.3. pp. 293–304.
31. *Magazine M. J., Oguz O.* A fully polynomial approximation algorithm for the 0–1 knapsack problem // European Journal of Operational Research. 1981. No.8. pp. 270–273.

Топка Владимир Владимирович. Институт проблем управления им. В.А. Трапезникова Российской академии наук. Старший научный сотрудник. Кандидат технических наук. Количество печатных работ: 75. Область научных интересов: управление проектами, методы оптимизации, теория надежности. E-mail: topka3@mail.ru; topka@ipu.ru

Multidimensional Knapsack problem: an effective solution method and its applying

V.V. Topka[I]

[I]Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

Abstract. Recently, there is an urgent need to develop an effective model and algorithmic apparatus for the top level of the project management system, on which there should be a subsystem for selecting and managing the portfolio of projects to be implemented. In this paper, we consider the optimal selection of the project portfolio and the allocation of resources for its implementation in the form of a discrete optimization problem – the multidimensional Knapsack problem. The paper presents known theoretical results for a greedy heuristic for solving a one-dimensional Knapsack problem. Necessary and sufficient conditions for optimality; estimation of the error of the algorithm and its asymptotic error for the behavior in the mean. A direct greedy heuristic at unit cost is proposed for solving the multidimensional Knapsack problem. To improve its effectiveness, we use a local limited search, which improves the greedy solution, as well as the addition and local optimization of the previous stages. Greedy heuristics are fast, with a polynomial time complexity.

Keywords: *knapsack problem, multidimensional Knapsack problem, project portfolio, improved greedy heuristic, local limited search, polynomial time complexity*.
DOI: 10.14357/20790279190206

References

1. *Gelrud Ya.* D. 2015. Metodologiya sozdaniya informatsionno-analiticheskoy sistemy upravleniya proyektami na osnove kompleksa matematicheskikh modeley funktsionirovaniya steykkholderov. [Methodology for creating an information and analytical system for project management on the basis of a complex of mathematical models of the functioning of stakeholders]: Dr. Sci. Thesis. Chelyabinsk. The South Urals State University (SRU). 43 p.
2. *Lopukhin M.M.* 1971. PATTERN – metod planirovaniya i prognozirovaniya nauchnykh rabot. [PATTERN – a method of planning and forecasting research developments.] M .: Soviet radio.
3. *Cetron M.J.* 1967. QUEST Status Report. IEEE Transactions on Engineering Management. 14(1):443.
4. *Nutt A.B.* 1969. Testing TORQUE – a quantitative R&D research allocation system. IEEE Trans. on Eng. Manag. EM-16(4).
5. *The Analysis and Evaluation* of Public Expenditures: the PPB System. 1969. Washington: U.S. Govern. Print Office.
6. *Jantsch E.* 1967. Technological forecasting in perspective: a framework for technological forecasting, its techniques and organization - Paris : Organization for Economic Co-operation and Development, – 401 p.
7. *Larichev O.I., Boychenko V.S., Moshkovich E.N., Sheptalova L.P.* 1978. Metody iyerarkhicheskikh skhem v programmno-tselevom planirovanii nauchnykh issledovaniy [Methods of hierarchical schemes in goal-oriented research planning] / Preprint.[Working Paper] M .: VNIISI.
8. *Martello S., Toth P.* 1990. Knapsack problems: Algorithms and computer implementations / in: Series in Discrete Mathematics and Optimization. N. Y.: Wiley Inter Sciences, 296 p.
9. *Kellerer H., Pferschy U., Pisinger D.* 2004. Knapsack Problems. Berlin / Heidelberg: Springer-Verlag, 548 p.
10. *Dantzig G.B.* 1957. Discrete-Variable Extremum Problems. Oper. Res. (INFORMS). 5(2): 266–277.
11. *Vizvari B.* 1992. On the Optimality of the Greedy Solutions of the General Knapsack Problems. Optimization. 23(2):125-138.
12. *Dyubin G.N., Korbut A.A.* 1999. Zhadnyye algoritmy dlya zadach o rantse: povedeniye v srednem [Greedy algorithms for Knapsack problems: average behavior.]. Sibirskiy zhurnal industrial'noy matematiki. [Siberian Journal of Industrial Mathematics.]. 2 (2(4)): 68-93.
13. *Korte B.* 1986. Kombinatorische Optimierung und algorithmische Prinzipien // Ökonomische Prognose-, Enstscheidungs- und Gleichgewichtsmodelle. Weinheim: VCH Verlagsgesellschaft,. pp. 286-341.
14. *Gafarov E.R., Lazarev A.A., Werner F.* 2010. Algorithms for Some Maximization Scheduling Problems on a Single Machine . Automation and Remote Control. 71(10): 2070–2084.

15. *Lazarev A., Salnikov A., Baranov A.* 2011. Graphical Algorithm for the Knapsack Problems. Lecture Notes in Computer Science. 6873: 459-466.
16. *Weingartner H.M., Ness D.N.* 1967. Method for the solution for the multidimensional 0–1 knapsack problem . Operations Research. 15: 83–103.
17. *Lorie J., Savage L.J.* 1955. Three problems in capital rationing. Journal of Business. 28: 229–239.
18. *Manne A.S., Markowitz H.M.* 1957. On the solution of discrete programming problems. Econometrica. 25: 84–110.
19. *Freville A.* 2004. The multidimensional 0–1 knapsack problem: An overview. European Journal of Operational Research. 155: 1–21.
20. *Levin M. Sh., Safonov A.V.* 2009. Evristicheskiy algoritm dlya mnogokriterial'noy blochnoy zadachi o ryukzake [Heuristic algorithm for the multi-criteria block Knapsack problem]. Iskusstvennyy intellekt i prinyatiye resheniy. [Artificial intelligence and decision making.] 4: 53-64.
21. *Burkova I.V., Tkachenko M.V., Churakov P.P.* 2010. Zadacha vybora mnozhestva proyektov pri vypuske innovatsionnoy produktsii [The problem of selecting a set of projects for the release of innovative products]. Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta. [Newsletter of the Voronezh State Technical University.]. 6(9):.120-124.
22. *Burkov V.N., Burkova I.V., Uandykov B.K., Chu D.S.* 2015. Metod dopustimykh resheniy v mnogomernoy zadache o rantse [The method of admissible solutions in the multidimensional Knapsack problem]. Ekonomika i menedzhment sistem upravleniya. [Economics and management of control systems.]. 18(4.1): 136-144.
23. *Burkova I.V., Puzhanova Ye.O., Marin O.L.* 2014. Zadacha o rantse i yeye modifikatsii [The Knapsack problem and its modifications]. Nauchnyy vestnik Voronezhskogo gosudarstvennogo arkhitekturno-stroitel'nogo universiteta. Seriya: Upravleniye stroitel'stvom. [Scientific herald of the Voronezh State Architectural and Construction University. Series: Management of construction.] Voronezh: VGASU, 1(6): 103-112.
24. *Burkov V.N., Burkova I.V., Popok M.V., Ovchinnikova T.I.* 2005. Metod setevogo programmirovaniya [Method of network programming]. Problemy upravleniya. [Control problems.] 3: 23 – 29.
25. *Finkel'shtein Iu.Iu.* 1976. Priblizhennye metody i prikladnye zadachi diskretnogo programmirovaniia [Approximate Methods and Applied Problems of Discrete Programming]. Moscow: Nauka, 264 p.
26. *Gafarov E.R., Dolgui A., Lazarev A.A., Werner F.* 2016. A New Effective Dynamic Program for an Investment Problem. Automation and Remote Control. 77(9): 1633-1648.
27. *Rubinshteyn M.I., Topka V.V.* 1991. Zhadnyi algoritm resheniya mnogomernoy zadachi o ryukzake [Greedy algorithm for solving the multidimensional Knapsack problem]. Trudy IV-oy Vsesoyuznoy shkoly-seminara «Statisticheskiy i diskretnyi analiz dannykh i ekspertnoye otsenivaniye» [4th School-Seminar (All-Union) "Statistical and Discrete Data Analysis and Expert Evaluation" Proceedings]. Odessa. 273-274.
28. *Schrijver A.* 1986. Theory of Linear and Integer Programming. N.Y.: J. Wiley & Sons Ltd, 702 p.
29. *Shani S.* 1975. Approximate algorithms for the 0–1 knapsack problem. Journal of Association for Computing Machinery. 22: 115–124.
30. *Chandra A.K., Hirschberg D.S., Wong C.K.* 1976. Approximation algorithms for some generalized knapsack problems. Theoretical Computer Sciences. 3: 293–304.
31. *Magazine M.J., Oguz O.* 1981. A fully polynomial approximation algorithm for the 0–1 knapsack problem. European Journal of Operational Research. 8: 270–273.

Topka V.V. Ph.D., Senior Researcher Scientist. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia. E-mail: topka3@mail.ru.