УДК 519.85

Б. И. Юхименко, Н. П. Волкова
Одесский национальный политехнический университет

ПРИБЛИЖЕННЫЕ АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ О МНОГОМЕРНОМ РАНЦЕ

Работа посвящена созданию комбинаторных алгоритмов решения многомерной задачи о ранце. Показана актуальность проблемы. Дан небольшой исторический анализ исследований и публикаций комбинаторных алгоритмов дискретной оптимизации. Обращено внимание на сложность вычислений при решении такого рода задач. Предположено использовать приближенные алгоритмы. В работе приведены три способа получения приближенных решений, разработанных на идеях жадного алгоритма, генетического алгоритма муравьиной колонии, а также некоторого комбинированного подхода. Сущность алгоритмов состоит в том, что конкретизация компонент вектора решений следует сформированной приоритетной очереди. Согласно ей присваивается значение «1» пока это допустимо. Получаемая последовательность зависит от использованной идеи. Значение целевой функции (рекорд) полученного решения является основой отсеивания вариантов при его улучшении. Само улучшение осуществляется через двойственный подход комбинаторных алгоритмов. Приведен числовой пример.
MSC: 68W25, 90C59.
Ключевые слова: комбинаторные методы, приближенное решение, многомерная задача о ранце, жадный алгоритм, муравьиная колония, рекорд, отсеивание.

Введение. Дискретные оптимизационные задачи находят широкое применение в различных областях, где используются математические методы для анализа происходящих там процессов. Необходимость решения таких задач предъявляет требования к разработке эффективных методов решения, позволяющих оперативно получить достаточно хорошее, а при возможности и оптимальное решение.

В настоящее время разработаны современные методы и алгоритмы решения задач дискретного программирования. Им посвящено множество научных разработок, их изучают в высших учебных заведениях. Разработаны пакеты прикладных программ, позволяющих решить ряд задач дискретного программирования. Наиболее удачные алгоритмы и программы решений задач целочисленного линейного программирования.

Современные методы точного или приближенного решения задач дискретной оптимизации достаточно сложные. В их основе лежат задачи из классов NP полных. Переход из класса NP полных к P полным – один из моментов, чему уделяется достойное внимание. Модификация существующих классических алгоритмов, улучшающих скорость сходимости, наиболее распространенный прием поиска быстродействующих алгоритмов.

Переход от алгоритмов поиска точного решения к алгоритмам поиска приближенного решения представляется очень важным приемом, поскольку не всегда поиск математически точного решения является необходимым в реальных

Получена 10.09.2017

условиях. Понятие тонкостей вычислительной реализации комбинаторных алгоритмов позволяет их «направлять» на нужное русло при поиске особенностей решаемых задач. Однако трудности, связанные с перебором большого количества вариантов, остается на повестке дня при совершенствовании алгоритмов. Разработка правил отсеивания неперспективных подмножеств вариантов одна из основных проблем совершенствования комбинаторных алгоритмов.

Несмотря на большие возможности персональных компьютеров и сервиза программного обеспечения, вопрос оперативного получения достаточно хороших решений в дискретной оптимизации актуален. В данной статье предлагается ряд способов получения приближенных решений, основанных на идее жадных алгоритмов, алгоритмов муравьиной колонии, а также комбинированные подходы имеющихся алгоритмов. Основной целью является получение алгоритма, предопределяющего вариант решения близкого к оптимальному за приемлемое время и сложности вычислений.

Важным моментом в комбинаторных методах дискретной оптимизации является процесс отсеивания так называемых неперспективных вариантов. Чем меньше вариантов пересматривается, тем скорость получения хорошего решения больше. Неперспективными вариантами решения принято называть те, которые заведомо, по каким-то причинам, не являются оптимальными вариантами. Наличие рекордного значения близкого к оптимальному позволяет с ним сравнивать прогнозное значение целевой функции варианта и принимать решение о его неперспективности. Работа фактически и посвящена формированию рекордных значений целевой функции.

В работе также описан способ улучшения полученного приближенного решения, а в некоторых случаях и получения оптимального. Способ основывается на двойственных алгоритмах [12], когда формирование решения начинается не с нулевого вектора, как обычно, а с вектора, компоненты которого единичные. Замена единиц нулями начинается с наименьшего значения компонента целевой функции. В предполагаемой работе, эта замена не зависит непосредственно от структуры целевой функции, а от способа, согласно которому, она появилась в варианте решения.

Одним из основных комбинаторных алгоритмов, использующих конечность множества вариантов дискретной оптимизации, является метод ветвей и границ. Начало такого рода алгоритмов было предложено Лэнд и Дойг [1] для решения задач целочисленного линейного программирования (ЦЛП). Однако главный толчок развития метода ветвей и границ дал метод Литла и др. [2] для решения задачи о коммивояжере. Структура метода такова, что выделяются моменты, которых можно модифицировать, учитывая особенности реальных задач. Метод вызвал живой интерес для разработки новых и новых подходов. К примеру [3,4].

Несколько другое направление в комбинаторной алгоритмизации дало появление алгоритма Балаша [5], который породил новое, так называемое одностороннее выявление при решении задач линейного программирования с булевыми переменными. На сегодняшний день есть алгоритмы, навеяны идеями Балаша (к примеру [6,7]). Смысл алгоритма с односторонним ветвлением состоит в том, что конкретизируются только единичные компоненты вектора решений. Невозможность поставить число «1» как компоненту вектора решений, предопределяет получение варианта решений. Полученный вариант либо совершенствуется с целью получения лучшего, либо является оптимальным, либо получили приближенное решение.

Приближенные методы широко применяются при решении ЦЛП, так как для точного нахождения решения может понадобиться значительные вычислительные средства. Современные приближенные методы обычно являются комбинированными, содержащие элементы различных методов. В приближенных методах решение задачи проводится в два этапа: построение начального решения и его улучшения. На первом этапе широко используются различные эвристические приемы, построение на правдоподобиях о свойствах оптимального решения задачи.

Среди приближенных методов ведущее место занимают так называемые жадные методы, применяемые как самостоятельно, так и в сочетании с другими подходами. Их несомненное преимущество состоит в том, что применение, скажем к задаче о ранце, их трудоемкость в основном имеет линейный вид по отношению к числу искомых величин. Жадные метода не гарантируют получение оптимального решения, но позволяют получить оценки вариантов так, что отклонение приблизительного решения от оптимального не велико. Согласно жадным алгоритмам предпочтение отдается предметам с наибольшей количественной их оценки. Обзор по самим алгоритмам приведен [8]. Жадный алгоритм для решения задачи о ранце описан в [9].

Новый перспективный подход к оптимизации базируется на имитации поведения колонии муравьев. Интенсивным результатом кооперативного поведения биологических муравьев является нахождение кратчайшего маршрута от источника пищи к гнезду. Алгоритмы оптимизации, имитирующие такое поведение муравьев, предложены в начале девяностых в Италии [10].

Популяризация алгоритмов муравьиной колонии происходило довольно быстро. Хорошие результаты получены для таких комбинаторных задач как задача о коммивояжере, раскраске графа, календарного планирования. Подробнее см.[11].

В сущности сама алгоритмизация производится через поведение муравьев при перемещении и передачи информации, через так называемые стигмержи. Биологически, стигмержи осуществляют через феромоны – специальный секрет, откладывается след при перемещении муравья. Чем выше концентрация феромонов по тропе, тем больше муравьев будет по ней двигаться.

В последнее время широко распространены приближенные алгоритмы, являющиеся модификациями точных методов, изначально ориентированы на поиск приближенного решения с оценкой отклонения от оптимального. В целом, приближенным методом посвящена обширная литература. В работе [12] приведены описания многих подходов к нахождению приближенных решений, а также дан список работ по данным вопросам.

Основные результаты. Многомерная задача о ранце имеет многочисленные приложения и является одной из основных модульных задач комбинаторной оптимизации. С точки зрения теории сложности эта задача NP-сложная. Поэтому в последнее время основное внимание уделяется приближенным методом решения данной задачи. Следует отметить, что эта тенденция характерна для комбинаторной оптимизации.

Известны различные подходы нахождения как точных, так и приближенных. При последовательном построении решения и компоновке варианта очередность конкретизации компонент вектора решения значительной степени предопределяет эффективность работы алгоритма метода ветвей и границ. Экспериментально это было показано в [7]. Одностороннее выявление исключает выбор приоритетного подмножества вариантов, содержащего компоненту с единичным и подмножества с нулевым значением конкретизируемой переменной. В связи с этим, очередность конкретизации еще более важный момент. Ниже предлагаются некоторые способы выбора переменной, которой присваивается значение «1».

Итак, рассматривается задача ЦЛП в постановке

$$Z = \max \sum_{j=1}^{n} c_j x_j$$

при ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i,\ i = \overline{1,m};$$

$$x_j \in \{0,1\},\ j = \overline{1,n}.$$

Алгоритм решения задачи состоит из трех этапов:

1. Формирование приоритетной последовательности конкретизации компонент вектора решений значением «1».
2. Формирование самого вектора решений.
3. Улучшение полученного решения.

Для формального описания перечисленных пунктов введем следующие обозначения: p_j - оценка j-ой компоненты вектора решений; $I_x = \{j_1, j_2, \ldots, j_n\}$ — последовательность индексов компонент вектора решений, предопределяющих порядок их конкретизации; s — шаг конкретизации; R — исходное значение целевой функции (рекорд),

$$K_x^s = \{j/x_j = 1\},$$

$$b_i^s = \left(b_i - \sum_{j \in K_x^s} a_{ij}\right),$$

$$V_j^s = \left\{j/b_i^s > 0,\ \ i = \overline{1,m}\ \right\}.$$

Формирование приоритетной последовательности можно осуществить несколькими способами. От этого непосредственно зависит близость получаемого решения к оптимальному решению, а также получаемое значение целевой функции — рекорд. Наличие рекорда уменьшает количество пересматриваемых вариантов при улучшении приближенного решения. Отсеивание вариантов очень важный момент при работе любых комбинаторных алгоритмов. Ниже рассматриваются способы формирования последовательности приоритетов компонент вектора решений — претендентов на получение значения «1».

Первый способ формирования последовательности приоритетов следует идеи жадного алгоритма. Ранжируемые компоненты вектора целевой функции в порядке невозрастания. Индексы компонент предопределяют приоритетную последовательность. Если $c_{j_1} \geqslant c_{j_2} \geqslant \cdots \geqslant c_{j_n}$, то множество индексов компонент вектора решения имеет ту же лексикографическую последовательность, а именно $I_x = \{j_1, j_2, \ldots, j_n\}$, согласно которой, будут присваиваться значения «1» компонентам вектора решений.

Второй способ навеян идеями алгоритмов муравьиной колонии. Феноменами, предопределяющими правильное направление движения биологических существ, предполагается считать частоту появления положительных значений компонент в векторах решения одномерных нецелочисленных задач о ранце.

Многомерную задачу о ранце можно рассматривать как состоящую из m-одномерных, имеющих одну и ту же целевую функцию. Оптимальное решение каждой из них является локальным по отношению к многомерной. Процедура решения одномерных задач о ранце вычислительном смысле несложная, особенно если отказаться от требований целочисленности искомых величин. Метод решения нецелочисленной задачи о ранце был предложен Данцигом еще в шестидесятых годах прошлого столетия (см. напр.[12]). Частота появления положительных значений в решениях локальных задач может рассматриваться как оценка важности каждой компоненты в векторе решений. Следуя терминологии муравьиной колонии это наличие феноменов.

Итак, для каждой i-той задачи составляются соотношения

$$\lambda_j^i = \frac{c_j}{a_{ij}}, \;\; j = \overline{1,n}; \quad i = \overline{1,m},$$

λ_j^i упорядочиваются в порядке не возрастания для каждого i в отдельности. Места расположения индекса j предопределяют очередность присвоения положительных значений компонент вектора локальных решений. Первые *(k-1)* компонент принимают значения «1». Имеем

$$x_{j_1}^i = x_{j_2}^i = \cdots = x_{j_{k-1}}^i = 1,$$
$$x_{j_{k+2}}^i = x_{j_{k+2}}^i = \cdots = x_n^i = 0,$$

k-ая компонента определяется как дробное значение, вычисляемое по формуле

$$x_{j_1}^i = \frac{b_i - \sum_{l=1}^{k-1} a_{ij_l}}{a_{ik}}.$$

Следует отметить, что k различное для каждого $i = \overline{1,m}$. Приоритетная оценка каждой j-ой компоненты определяется величиной

$$p_j = \sum_{i=1}^{m} x_j^i.$$

Ранжированная последовательность p_j в порядке невозрастания определяет множество $I_x = \{j_1, j_2, \ldots, j_n\}$.

Третий способ предполагает комбинирование идеи жадного алгоритма с учетом возможностей присвоения значения «1» большему количеству компонент вектора решений. Другими словами, оценка определяется по формуле

$$p_j = c_j * \sum_{i=1}^{m} (b_i - a_{ij}), \quad j = \overline{1,n}.$$

Ранжировка в порядке невозрастания оценок p_j предопределяет $I_x = \{j_1,j_2,\ldots,j_n\}$.

Формирование варианта решения. Изначально берем вектор $x = (x_1,x_2,\ldots,x_n)$, компоненты которого имеют значение «0». Согласно последовательности индексов I_x присваиваются значения

$$x_{j_1} = x_{j_2} = \cdots + x_{j_{k-1}} = 1, \quad x_{j_k} = 0,$$

где k определяется из соотношения $\left(b_i - \sum_{l=1}^{k-1} a_{ij_l}\right) \geqslant 0$ для $i = \overline{1,m}$.

Если существует такой $j_r(r > k)$, что $(b_i^r - a_{ij_r}) \geqslant 0$ при $i = \overline{1,m}$, то $x_{j_r} = 1$. Следует отметить, что таких r может быть больше, чем один.

Улучшение полученного приближенного решения предлагается осуществить через способ двойственности при формировании решения [12]. В называемом прямом способе конкретизации переменных вектор решений $x = (x_1,x_2,\ldots,x_n)$ содержит все нулевые компоненты ($x_j = 0,\ \ j = \overline{1,n}$). Конкретизируются значения «1» согласно приоритетной очереди I_x. В двойственном способе считается, что исходное положение вектора таково, что $x_j = 1$ ($j = \overline{1,n}$). Определяется, какие компоненты x_j следует сделать нулевыми, чтобы довести вектора до допустимого решения, удовлетворяющего системе ограничений.

В рассматриваемом случае имеется решение, в котором несколько компонент имеется со значением «1». Проверяется возможность улучшить имеющиеся решение, если «1» заменить на «0».

Пусть имеется приближенное решение $X = (x_1,x_2,\ldots,x_n)$, в котором $x_q = 1$. Принимается $x_q = 0$; формируется K_x^{s-1}; а также $b_i^{s-1} = (b_i^s + a_{iq})$, $i = \overline{1,m}$; определяется содержание V_j^s. Если $V_j^{s-1} \neq \varnothing$, то проверяется возможность улучшения имеющегося рекорда R. А именно, если

$$\left(\sum_{j \in K_x^{s-1}} c_j + \sum_{j \in V_x^{s-1}} c_j\right) > R,$$

то формируется новый вектор $X = (x_1,x_2,\ldots,x_q = 0,\ldots x_n)$ на основе последовательности I_x. В противном случае улучшить решение нельзя, заменяя значение x_q на нулевое.

Возможны следующие случаи:

1. Получили решение лучше предыдущего, получаем новый рекорд.
2. Убедились, то улучшение быть не может.
3. Следует выбрать следующую компоненту со значением «1» в приближенном решении.

Таким образом, поочередно компоненты со значением «1» меняют значение «0» и каждый раз проверяются на возможность улучшения.

Пример. Дана задача о многомерном ранце

$$Z = \max(4x_1 + 2x_2 + 5x_3 + 3x_4 + 6x_5 + x_6 + 7x_7)$$

при ограничениях

$$\begin{cases} 2x_1 + 3x_2 + 4x_3 + 6x_4 + 2x_5 + x_6 + 7x_7 \leqslant 10 \\ 4x_1 + 6x_2 + 3x_3 + 2x_4 + x_5 + 3x_6 + 6x_7 \leqslant 10 \\ 6x_1 + 2x_2 + 4x_3 + 3x_4 + 5x_5 + 2x_6 + 4x_7 \leqslant 10 \end{cases}$$

$$x_j \in \{0{,}1\}\,,\; j=\overline{1{,}7}.$$

Согласно первому способу формирования приоритетов, имеем $I_x = \{7,\,5,\,3,\,1,\,4,\,2,\,6\}$. Последовательность соответствует ранжированной последовательности коэффициентов целевой функции.

Вариант решения согласно полученной последовательности определяется следующим образом

$$x_7 = 1;\quad 10-7=3;\; 10-6=4;\; 10-4=6;\quad R=7;$$

$$x_5 = 1;\quad 3-2=1;\; 4-1=3;\; 6-5=1;\quad R=7+6=13$$

Больше единичных значений присвоить нельзя. Вариант

$$x=\{0,\,0,\,0,\,0,\,1,\,0,\,1\}\,;\quad R=13.$$

Определим вариант согласно второму способу формирования приоритетов.

$$\lambda_j^1 ::= \frac{4}{2};\frac{2}{3};\frac{5}{4};\frac{3}{6};\frac{6}{2};\frac{1}{1};\frac{7}{7};$$

Расположение индексов j согласно ранжировке λ_j^1 следующее $\{5,\,1,\,2,\,4,\,3,\,7,\,6\}$; $x_5=x_1=x_2=1$; $x_5=\frac{3}{6}=\frac{1}{3}$; $Z_1=13{,}0$;

$$\lambda_j^2 ::= \frac{4}{4};\frac{2}{6};\frac{5}{3};\frac{3}{2};\frac{6}{1};\frac{1}{3};\frac{7}{6};$$

Расположение индексов j $\{5,\,3,\,4,\,7,\,1,\,6\}$; $x_5=x_3=x_4=1$; $x_7=4/6=2/3$; $Z_2=18{,}6$;

$$\lambda_j^3 ::= \frac{4}{6};\frac{2}{2};\frac{5}{4};\frac{3}{3};\frac{6}{5};\frac{1}{2};\frac{7}{4};$$

Расположение индексов j следующее: $\{7,\,5,\,3,\,2,\,4,\,1,\,6\}$; $x_7= x_5=1$; $x_3=1/4$; $Z_3=14{,}2$;

Локальные решения следующие:

$$\begin{array}{cccccc}
1 & 1 & 0 & 1/3 & 0 & 0\\
0 & 0 & 1 & 1 & 0 & 2/3\\
0 & 0 & 1/4 & 0 & 0 & 1
\end{array}$$

Приоритетные оценки p_j для каждого j следующие:

$$P_1=1;\quad P_2=1;\quad P_3=1\frac{1}{4};\quad P_4=1\frac{1}{3};\quad P_5=3;\quad P_6=0;\quad P_7=1\frac{2}{3};$$

Ранжированная последовательность приоритетов $I_x=\{5,\,7,\,4,\,3,\,1,\,2,\,6\}$. Приближенное решение $x=\{0,\,0,\,0,\,0,\,1,\,0,\,1\}\,;\; R=13$;

Согласно третьему — комбинированному способу — приоритетные оценки определяются с учетом суммарных невязок в системе ограничений.

$$P_1=\{(10-2)+(10-4)+(10-6)\}\cdot 4=72;\; P_2=19\cdot 2=38;$$

$$P_3 = 19 \cdot 5 = 95;\ P_4 = 19 \cdot 3 = 57;\ P_5 = 22 \cdot 6 = 132;\ P_6 = 24 \cdot 1 = 24;$$

$$P_7 = 13 \cdot 7 = 91.$$

Последовательность приоритетов $I_x = \{5,\, 3,\, 7,\, 1,\, 4,\, 2,\, 6\}$. Приближенное решение $x = \{0,\, 0,\, 1,\, 0,\, 1,\, 0,\, 0\}$; $R = 11$.

Проведем улучшение полученных приближенных решений. Для первого способа имеем

$$X = \{0,\, 0,\, 0,\, 0,\, 1,\, 0,\, 1\};\ I_x = \{7,\, 5,\, 3,\, 1,\, 4,\, 2,\, 6\};\quad R = 13.$$

Предполагается $x_5 = 0$;

$$b_i^1 = \begin{pmatrix} 3 \\ 4 \\ 6 \end{pmatrix};\quad V_j^1 = \{1{,}6\};\ \sum_{j \in v_j^1} c_j = 5;\quad 5 + 7 < 13.$$

Улучшения не может быть.

Предполагается $x_7 = 0$;

$$b_i^1 = \begin{pmatrix} 8 \\ 9 \\ 5 \end{pmatrix};\quad V_j^1 = \{2,\, 3,\, 4,\, 6\};\ \sum_{j \in v_j^1} c_j = 9 + 6 = 15.$$

Согласно последовательности I_x принимаем $x_3 = 1$;

$b_i^2 = \begin{pmatrix} 4 \\ 6 \\ 1 \end{pmatrix}$; больше единичных значений нельзя поставить, т.е. $V_j^3 = \varnothing$.

Имеем $Z = 6 + 5 = 1 < 13$. Улучшение нет.

Для второго способа имеем

$$X = \{0,\, 0,\, 0,\, 0,\, 1,\, 0,\, 1\};\ I_x = \{5,\, 7,\, 4,\, 3,\, 1,\, 2,\, 6\};\quad R = 13.$$

Предполагаем $x_5 = 0$;

$$b_i^1 = \begin{pmatrix} 3 \\ 4 \\ 6 \end{pmatrix};\quad V_j^1 = \{1,\, 6\};\ \sum_{j \in V_j^1} c_j = 5;\qquad 5 + 7 = 12 < 13;$$

Аналогично, как и в первом случае, улучшения не может быть.

Предполагаем $x_7 = 0$;

$$b_i^1 = \begin{pmatrix} 8 \\ 9 \\ 5 \end{pmatrix};\quad V_j^1 = \{2,\, 3,\, 4,\, 6\};\ \sum_{j \in V_j^1} c_j = 9;\qquad 9 + 6 = 15;\quad 15 > R = 13;$$

Проверяем: $x_4 = 1$; $b_i^2 = \begin{pmatrix} 2 \\ 7 \\ 2 \end{pmatrix}$; $V_j^2 = \{6\}$; $x_6 = 1$; $b_i^3 = \begin{pmatrix} 1 \\ 4 \\ 0 \end{pmatrix}$; $V_j^3 = \varnothing$;

$Z = 6 + 3 + 1 = 10$; $10 < 13$; Улучшения нет.

Для третьего способа имеем

$$X = (0,\,0,\,1,\,0,\,1,\,0,\,0)\,;I_x = \{5,\,3,\,7,\,1,\,4,\,2,\,6\}\,.$$

Полагаем $x_3 = 0$;

$$b_i^1 = \begin{pmatrix} 8 \\ 9 \\ 5 \end{pmatrix};\;\; V_j^1 = \{2{,}4{,}6{,}7\}\,;\;\; \sum_{j\in V_j^1} c_j = 13;\;\; Z = 13 + 6 = 19; \quad 19 > R = 11;$$

Проверяем: $x_7 = 1$; $b_i^2 = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$; $V_j^3 = \varnothing$; $Z = 13 > 11$. Вариант решения $X = (0{,}0{,}0{,}0{,}1{,}0{,}1)\,; R = 13$; Полагаем $x_5 = 0$;

$$b_i^1 = \begin{pmatrix} 6 \\ 7 \\ 6 \end{pmatrix};\;\; V_j^2 = \{1{,}2{,}4{,}6\}\,;\;\; \sum_{j\in V_j^2} c_j = 12;\;\; Z = 5 + 12 = 17 > 11;$$

Проверяем: $x_1 = 1$; $b_i^2 = \begin{pmatrix} 4 \\ 3 \\ 0 \end{pmatrix}$; $V_j^2 = \varnothing$; $Z = 4 + 5 = 9 < 13$. Приближенное решение

$$X = (0{,}0{,}0{,}0{,}1{,}0{,}1)\,; \quad R = 13.$$

Заключение. В работе приведено построение двухэтапных алгоритмов решения многомерной задачи о ранце. На первом этапе рассматриваются способы построения ранжированных последовательностей индексов компонент вектора решений, согласно которым осуществляется конкретизация переменных x_j со значением «1». Использованы эвристические подходы, основанные на правдоподобиях, но не имеющих строгих доказательств.

Предложены три способа построения последовательностей. Первый основан на идее жадного алгоритма, третий комбинированный способ, и именно, идея жадного с учетом величины суммарной невязки в системе ограничений, практически предопределяющих количество единичных компонент в векторе решений и, в какой-то степени, значение целевой функции. Второй алгоритм навеян идеями метаэвристических алгоритмов. Здесь используется поведение биологических существ в поиске пищи. Другими словами, в поиске оптимального варианта достижения цели — накормиться.

Существует множество алгоритмов оптимизации основанных на идее муравьиной колонии. В работе дан новый подход поиска определения компонент, претендующих на значение «1». На основе частоты появления положительных значений компонент в векторах, решения локальных одномерных нецелочисленных задач о ранце составляется последовательностью претендентов. Разделение многомерной задачи о ранце на множество одномерных давно используется для получения оценки множества вариантов в методе ветвей и границ при двухстороннем ветвлении [12].

Вторая часть алгоритмов предполагает улучшение полученного приближенного решения. Эта часть использует идею двойственности при формировании самого решения. Проверяется возможность улучшить решение за счет поочередной замены компонент со значением «1» на значение «0». Есть случаи, что это удается. Однако главную роль играет построение приоритетной последовательности конкретизации. Приведенный пример как-то демонстрирует возможности алгоритмов. Проводится большой эксперимент. Более весомое слово будет сказано позже.

1. **Land A. H.** An automatic method of solving discrete programming probems / A. H. Land, A. G. Doig // Econometrica. – 1960. – V. 28, №3. – P. 497–520.
2. **Little J. D. C.** An algorithm for the traveling salesman problem / J. D. C. Little, K. G. Murty, C. Karel // Operat. Res. – 1963. – V. 11, №6. – P. 972-989.
3. **Меламед И. И.** Задача коммивояжера. Точные алгоритмы / И. И. Меламед, С. И. Сергеев, И. Х. Сигал // Автоматика и телемеханика. – 1989. – №10. – с. 3-29.
4. **Меламед И. И.** Задача коммивояжера. Приближенные алгоритмы / И. И. Меламед, С. И. Сергеев, И. Х. Сигал // Автоматика и телемеханика. – 1989. - №11. – с. 3–20.
5. **Balas E.** An additive algorithm for soving linear programs with zero-one variables // Operat. Res. – 1965. – V. 13, №4. – P. 517–546.
6. **Юхименко Б. И.** Ускоренный алгоритм метода ветвей и границ для решения задачи целочисленного линейного программирования // Тр. Одес. политехн. ун-та. – 2004. – Вып. 2. – с. 223–226.
7. **Юхименко Б. И.** Сравнительная характеристика алгоритмов метода ветвей и границ для решения задач целочисленного линейного программирования / Б. И. Юхименко, Ю. Ю. Козина // Тр. Одес. политехн. ун-та. – 2005. – Вып. 2. – с. 199–204.
8. **Karte B.** Kombinatorische Optimierung und algorithmische Principien // Okonomische Prognose, Entscheidungs und Gleichgewichtsmodelle. Weinheim : VCH Verlangsgesellschschact, 1986. – P. 286–341.
9. **Дюбин Г. Н.** Жадные алгоритмы для задачи о ранце: поведение в среднем / Г. Н. Дюбин, А. А. Корбут // Сибирский журнал индустриальной математики. Том II, №2(4). – 1999. – с. 68–93.
10. **Dorigo M.** Optimization, Learning and Natural Algorithms // PhD Thesis. Dipertamento di Ellettronica, Politechnico Di Milano. Italy. – 1992. – 140 p.
11. **Штобва С. Д.** Муравьиные алгоритмы: Теория и применение // Программирование. – 2005, №4. – с. 1–16.
12. **Сигал И. Х.** Введение в прикладное дискретное программирование / И. Х. Сигал, А. П. Иванова. – М. : Физматлит, 2007. – с. 304.

Юхименко Б. І., Волкова Н. П.
Наближені алгоритми розв'язання задачі про багатовимірний ранець

Резюме

Робота присвячена створенню комбінаторних алгоритмів розв'язання багатовимірної задачі про ранці. Показана актуальність проблеми. Дан невеликий історичний аналіз досліджень і публікацій комбінаторних алгоритмів дискретної оптимізації. Звернуто

увагу на складність обчислень при вирішенні такого роду завдань. Припущено використовувати наближені алгоритми. В роботі наведені три способи отримання наближених рішень, розроблених на ідеях жадібного алгоритму, генетичного алгоритму мурашиної колонії, а також деякого комбінованого підходу. Сутність алгоритмів полягає в тому, що конкретизація компонент вектора рішень слідує сформованої пріоритетної черзі. Відповідно до неї присвоюється значення «1» поки це допустимо. Отримана послідовність залежить від використаної ідеї. Значення цільової функції (рекорд) отриманого рішення є основою відсіювання варіантів при його поліпшенні. Саме поліпшення здійснюється через подвійний підхід комбінаторних алгоритмів. Наведено числовий приклад.
Ключові слова: комбінаторні методи, наближене рішення, багатовимірна задача про ранці, жадібний алгоритм, мурашина колонія, рекорд, відсіювання .

Yukhymenko B. I., Volkova N. P.
Approximate algorithms for solving the multidimensional knapsack problem

Summary

This paper is devoted to creating combinatorial algorithms for solving the multidimensional knapsack problem. The urgency of the problem is shown. A small historical analysis of researches and publications of the combinatorial algorithms for discrete optimization is given. Attention is drawn to the computational complexity in solving such kind of problems. It is expected to use approximation algorithms. The paper gives three ways of obtaining approximate solutions developed on the ideas of a greedy algorithm, the genetic algorithm of ant colony, and also some combined approach. The essence of algorithms is that the specification of the solution vector components follows the formed queue priority. It is assigned the value "1" as long as this is allowed. The obtained sequence depends form the used idea. The value of the objective function (record) of the obtained solution is the basis for screening out options as it improves. The improvement is realized through the dual approach of combinatorial algorithms. A numerical example is given.
Key words: combinatorial methods, approximate solution, multidimensional knapsack problem, greedy algorithm, ant colony, record, screening out.

References

1. Land, A. H. & Doig, A. G. (1960). An automatic method of solving discrete programming probems, *Econometrica*, vol. 28, №3, pp. 497–520.
2. Little, J. D. C., Murty & K. G., Karel, C. (1963). An algorithm for the traveling salesman problem, *Operat. Res.*, vol. 11, №6, pp. 972–989.
3. Melamed, I. I., Sergeev, S. I. & Sigal, I. H. (1989). Zadacha kommivoyazhera. Tochnyie algoritmyi, *Avtomatika i telemehanika*, no. 10, pp. 3–29.
4. Melamed, I. I., Sergeev, S. I. & Sigal, I. H. (1989). Zadacha kommivoyazhera. Priblizhennyie algoritmy, *Avtomatika i telemehanika*, no. 11, pp. 3–20.
5. Balas, E. (1965). An additive algorithm for soving linear programs with zero-one variables, *Operat. Res.*, vol. 13, no. 4, pp. 517–546.
6. Yuhimenko, B. I. (2004). Uskorennyiy algoritm metoda vetvey i granits dlya resheniya zadachi tselochislennogo lineynogo programmirovaniya, *Tr. Odes. politehn. un-ta*, issue 2, pp. 223–226.

7. Yuhimenko, B. I. & Kozinam, Yu. Yu. (2005). Sravnitelnaya harakteristika algoritmov metoda vetvey i granits dlya resheniya zadach tselochislennogo, lineynogo programmirovaniya, *Tr. Odes. politehn. un-ta*, issue 2, pp. 199–204.

8. Karte, B. (1986). Kombinatorische Optimierung und algorithmische Principien, *Okonomische Prognose-, Entscheidungs- und Gleichgewichtsmodelle*, Weinheim : VCH Verlangsgesellschschact, pp. 286–341.

9. Dyubin, G. N. & Korbut, A. A. (1999). Zhadnyie algoritmyi dlya zadachi o rantse: povedenie v srednem, *Sibirskiy zhurnal industrialnoy matematiki*, vol. II, no. 2(4), pp. 68–93.

10. Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*, PhD Thesis. Dipertamento di Ellettronica, Politechnico Di Milano, Italy, 140 p.

11. Shtobva, S. D. (2005). Muravinyie algoritmyi: Teoriya i primenenie, *Programmirovanie*, no. 4, pp. 1–16.

12. Sigal, I. H. & Ivanova, A. P. (2007). *Vvedenie v prikladnoe diskretnoe programmirovanie*, Fizmatlit, Moscow, 304 p.