НЕКОТОРЫЕ ИНФОРМАЦИОННЫЕ ПРОБЛЕМЫ КОМБИНАТОРНОЙ ЛИНЕЙНОЙ ОПТИМИЗАЦИИ

Юхименко Б.

Дискретная оптимизация одна из наиболее затребованных дисциплин, с которыми связана процедура принятия решений. Ее использование связано со многими проблемами, которые процедурно объемны. Построение математической модели, подбор действительно существенных факторов, приведение к типам существующих математических задач и дальнейшая реализация на компьютерной технике довольно трудоемкий процесс. Существующие математические методы как аналитического типа и комбинаторного характера все еще относятся к классу NP сложности. Хотя современная вычислительная техника сверхмощная и хорошо оснащена математическим обеспечением решения задач особенно большой размерности практически не реализуемая в случае требования – получить оптимальное решение.

Кроме того, комбинаторные методы и алгоритмы решения задач дискретной оптимизации связаны с большим количеством промежуточной информации, необходимой для получения оптимального или хотя бы достаточно хорошего решения. В этой связи, разработка новых, модифицирующих блоков и модулей, включаемых в схемы существующих алгоритмов, в настоящее время затребовано, актуально и имеет как научное, так и прикладное значение.Данная разработка представляет в некотором смысле систематизацию проблем метода ветвей и границ для решения задачи линейного программирования с булевыми переменными в основном, и многомерной задачи о ранце в частности. Перечислены основные моменты метода, которые можно модифицировать. Показана возможность подключить моменты приближенных алгоритмов подготовительная часть точного алгоритма. Решен числовой пример, демонстрирующий положительное влияние двух алгоритмических процедур на эффективность работы метода. Использование рекордного значения целевой функции уменьшает на половину число пересматриваемых вершин, а введение приоритетной последовательности компонент для конкретизации еще отсеивает значительное количество вершин.

Ключевые слова. Дискретная оптимизация, рекорд, последовательное построение решения, приближенные алгоритмы, конкретизация, приоритетная последовательность.

ВВЕДЕНИЕ

Принятие оптимальных решений в практической деятельности человека это многогранная и многоэтапная проблема. Доведение процесса от постановки задачи до получения результата решения и его внедрения встречается много различных препятствий и задач, которые необходимо решать. Многие вычислительные алгоритмы оптимизации содержат моменты, связанные с решением задач информационного характера. Эффективность работы алгоритмов комбинаторного характера не исключает проблему организации исходной, промежуточной и результирующей информации.

Метод ветвей и границ как основной комбинаторный метод дискретной оптимизации так же подвергается всеобщему обсуждению по всем его составным частям и процедурам. Ему посвящено множество публикаций, разработано немало модификаций, нацеленных на уменьшение количества перебираемых вариантов, однако, он все еще остается алгоритмом NP – сложности. Сложность вычислений увеличивается за счет обслуживания информационного обеспечения работы алгоритмов. В научной литературе этот вопрос обсуждается недостаточно хорошо как в теоретическом, так и в практическом плане. В работе [1] приведен перечень необходимых информационных массивов, целый список параметров, значения которых необходимо для формирования подзадач, используемых для получения подмножеств вариантов.

Использование идеи последовательного построения решения [2] как средство разбиения множества вариантов на подмножества, на наш взгляд, подает подсказку другого подхода к формированию и работы с массивами промежуточной информации. Практически последовательное построение решения предполагает конкретизацию значений компонент вектора решений.

Запишем этот процесс формально.

Пусть имеется задача целочисленного линейного программирования (ЦЛП) в постановке

$$Z = \max(C, X)$$

при ограничениях

$$\sum_{j=1}^{n} A_j x_j \leq B$$

$$X = (x_1, x_2, \ldots, x_n), \quad x_j \in \{0, N\},\ j = \overline{1, n}$$

Обозначим через σ^k частичное решение, в котором конкретизировано (определено) k компонент вектора решений *X*. Множество I_l^k содержит номера конкретизированных компонент соответственно. Число k предопределяет ярус дерева решений. Действительно, любой алгоритм метода ветвей и границ имеет древовидную схему получения оптимального решения. При работе алгоритма формируется само дерево, вершинами которого соответствуют подмножества вариантов, содержащих частичное решение σ_l^k . Поскольку конкретизуемая переменная может принимать не одно целочисленное значение, то на *k*-том уровне (ярусе) имеется S_k вершин ($l=\overline{1,S_k}$). Для каждой вершины формируется подзадача с целью получения оценки подмножества. Пусть G_l^k подмножество вариантов, содержащее частичное решение σ_l^k , а $\xi(G_l^k)$ - оценка подмножества. Оценка получается путем решения соответствующей подзадачи. Информационно каждая подзадача отличается одна от другой лишь компонентами двух векторов – вектора условий A_{l_0} и вектора ограничений *B*. Индекс l_0 показывает, какая компонента вектора решений конкретизирована при переходе от *k*-1-го яруса к *k*-ому.

Таким образом, на *k*-ом ярусе дерева решений достаточно иметь информацию о одной подзадаче, для остальных безболезненно формируется программно.

Экспериментально было показано [3], что в алгоритмах, реализующих идею последовательного построения решения, скорость сходимости значительно зависит от выбора компоненты, подлежащей конкретизации на очередном уровне дерева решений. Выгодно ввести процедуру $\Psi(j)$, предопределяющую номер компоненты подлежащей конкретизации на конкретном шаге работы алгоритма. Под воздействием процедуры $\Psi(j)$ останутся только те компоненты, которые имеют потенциал на более одного не отрицательного значения. Обозначим множество индексов таких компонент через V_l^k

$$V_l^k = \{j \,/\, j \overline{\in} I_{l_0}^k \,/ (b_i - \sum_{l \in I_{l_0}^k} a_{il}\sigma_{l_0}^k) \geq 0 \ \forall_i\}$$

Если $V_l^k \equiv \varnothing$, то данную ветвь можно исключить из рассмотрения.

Еще один заслуживающий внимание момент. Поскольку в процессе работы алгоритма формируется дерево решений, согласно структуре, которого содержится информация о решаемых подзадачах, то чем меньше ветвей будет порождено (особенно при первом подходе), тем больше подмножеств вариантов исключается из рассмотрения. Изначально наличие рекордного значения целевой функции R запретить порождать ветви, представляющие неперспективные подмножества. Они будут отсеены.

ОСНОВНАЯ ЧАСТЬ

Что касается решения задач с булевыми переменными, в частности, многомерной задачи о ранце, то само формирование частичного решения значительно упрощается. На каждом ярусе дерева решений имеется только два частичных решения. Конкретизируемая переменная может принимать только два значения либо «0» либо «1». Значительно уменьшается количество хранимой информации о подзадачах. Аналогично, как и в общем случае меняется только два вектора, компоненты вектора конкретизируемой переменной и вектора ограничений.

В алгоритмическом смысле увеличить количество отсеиваемых неперспективных подмножеств для алгоритма решений задач о ранце можно за счет следующих моментов:

- получение более точного рекордного значения R изначально до формирования дерева решений;

- установка приоритетной очереди номеров компонент вектора решений претендентов на значение «1»;

- формирование множества индексов компонент («хороших компонент»), имеющих потенциал на значение «1».

- проверка на наличие потенциала среди хороших компонент улучшить значение целевой функции;

- определение функции предпочтения при возврате к вершине дерева решений.

Алгоритмическая реализация некоторых из этих моментов основывается на идеях построения приближенных или вероятностно-приближенных алгоритмов. Разные приближенные алгоритмы предполагают различные подходы формирования приоритетной очереди компонент, претендентов на значение «1». Процедура $\Psi(j)$ может пошагово определять номер компоненты в зависимости от сложившейся ситуации, а может формировать всю приоритетную последовательность номеров компонент. В принципе эта идея не является новой. Алгоритм Балаша [4] , представляющий алгоритм метода ветвей и границ с односторонним ветвлением, практически состоит из процедур поиска компонент, которым присваивается значение «1». Негативный момент алгоритма то, что признак оптимальности не строгий и может привести к полному перебору вариантов. Он тоже основан на поиске компонент, которые могут иметь значение «1». Сама процедура проверки на оптимальность полученного решения более длительная, чем получение самого варианта. Он может считаться приближенным решением, хотя зачастую, он является оптимальным.

Современные приближенные алгоритмы предполагают формирование всей приоритетной очереди компонент-претендентов на значение «1». Обозначим эту последовательность номеров компонент через $\mathrm{P} = (j_1, j_2, ..., j_n)$. Место компоненты с номером j_i в разных алгоритмах определяется по-разному согласно своей методике алгоритма. Рассмотрим некоторые из них.

Пусть имеется математическая модель многомерной задачи о ранце

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

при ограничениях

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

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

- где $c_j, a_{ij}, b_i \ \forall_{ij}$ - натуральные числа.

Жадный алгоритм [5] предлагает упорядочить номера компонент j_i согласно ранжированной в порядке не возрастания компонент вектора стоимостей. Если $c_{j_1} \ge c_{j_2} \ge ... \ge c_{jn}$, то приоритетная очередь будет $P = (j_1, j_2, ..., j_n)$.

С целью получения оптимального решения нецелочисленной одномерной задачи о ранце Дацинг [1] предлагал модифицированную последовательность ранжировки. Если $\dfrac{c_{j_1}}{a_{j1}} \ge \dfrac{c_{j_2}}{a_{j2}} \ge ... \ge \dfrac{c_{jn}}{a_{jn}}$, то $(j_1, j_2, ..., j_n)$ составляет приоритетную очередь компонент вектора решений.

В некоторых комбинированных приближенных алгоритмах предполагается учитывать величину невязки в системе ограничений при подборе приоритетной компоненты. Каждая компонента оценивается величиной r_j, которая определяется так

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

Величина r_j ранжируется в порядке не возрастания и определяет приоритетную очередь.

Автор проводила такой эксперимент [3]. Увеличение эффективности работы алгоритма по сравнению с обычным жадным алгоритмом практически не наблюдается.

Генетический алгоритм [7] какойто степени учитывает «наследственность». Если получено определенное количество вариантов решения, то приоритетность компонент оценивается частотой получения значений «1» в этих вариантах. На наш взгляд, можно использовать результаты той работы, которая проделывается, получая оценки подмножеств и решая для этого подзадачи. Решается *m* одномерных нецелочисленных задач о ранце, используя метод Дацига [6]. Имеем

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

при ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \qquad x_j \in [0,1], \quad j = \overline{1,n}.$$

Имеем *m* оптимальных решений $X_i^* = (x_1^{*i}, x_2^{*i}, ..., x_n^{*i})\ (i = \overline{1,m})$. Оценка компоненты вектора решений r_j определяется числом полученных значений соответствующей компоненты в векторах X_i^*.

$$r_j = \sum_{i=1}^{m} x_j^{*i}, \ j = \overline{1,n}.$$

Величина r_j предопределяет приоритетность компонент вектора решений. В последнее время разрабатываются алгоритмы, основанные на поведении живых существ или каких-то природных явлений. Хорошо известный алгоритм муравьиной колонии [8,9] используется при решении многих оптимизационных задач. Использование такого типа алгоритмов позволяет получить достаточно хорошее приближенное, а иногда и оптимальное решение задач большой размерности [10].

Особи муравьиной колонии могут передавать информацию, действовать самостоятельно, достигая общей цели, менять направление движения в зависимости от создавшейся ситуации (обратная связь). Все это удается формализовать и создавать эффективное вероятностно-приближенные алгоритмы. Приоритетность компонент вектора решений получить значение «1» определяется величиной вероятности p_j, количественное значение которой зависит от количества информации (феромонов) при массовом движении в одном направлении, изменение направления (испарение феромонов), а так же специфических особенностей решаемой задачи. Имеем

$$p_j = f^1(\alpha, \beta) \cdot f^2(c_j, b_j),$$

где α - предопределяет количество феромонов, как информацию о повторении некоторых алгоритмических действий, β - испарение феромонов. Величина β рассматривается как параметр, от которого зависит эффективность работы алгоритма. Функция $f^2()$ формируется индивидуально создателям алгоритма и зависит от его мнения [13]. Она каким то образом увеличивает количественно значение величины p_j и дает некое предпочтение компоненте j при получении значения «1». Процедурно сам процесс присвоения положительного значения осуществляется случайным образом. Определяется значение η, как выпад равномерно распределенной случайной величины из интервала (0,1). Попадание η в определенный интервал пропорционально вероятности p_j определяет претендента на значение «1». Естественно, необходимо учесть такую возможность при наличии ограничивающих условий.

Упростить процедуру поиска приоритетной компоненты (назовем ее хорошей компонентой) можно тем, что изначально выделяются компоненты, имеющие потенциал положительного значения [12]. Пусть на каком то k-ом уровне рассматривается компонента l_0, тогда вершине соответствует множество $\mathrm{I}_{l_0}^{k}$, определяющее номера конкретизированных компонент. Множество V_l^k предопределяет компоненты, которым можно присвоить значение «1». Они определяются так

$$V_l^k = \{ j \,/\, j \overline{\in} I_{l_0}^{k} \,/ (b_i - \sum_{l \in I_{l_0}^{k}} a_{il}) > 0 \ \forall_i \}$$

В случае, если $V_l^k = \varnothing$, то продолжение по соответствующей ветви дерева решений нецелесообразно. Если $V_l^k \neq \varnothing$ т.е. существуют компоненты, которым можно присвоить значение «1» , но в сумме не могут увеличить значение целевой функции чтобы превысить имеющийся рекорд, то продолжение рассмотрения подмножеств вариантов не имеет смысла. Формально можно записать так

$$(\sum_{j \in I^k} c_j x_j + \sum_{j \in V_l^k} c_j) \leq R$$

где I^k индексы конкретизированных компонент.

Если не выполняется признак оптимальности для полученного варианта решений, имеются висящие вершины с лучшими оценками, чем рекорд *R,* то вопрос – какой вершине отдать предпочтение. Возможны такие называемые функции предпочтения выбора подмножества вариантов:

- с максимальной оценкой;
- с частичным решением, где *k* максимально;
- случайный выбор пропорционально величинам оценок;
- случайный выбор среди трех (пяти, …) вершин с максимальной оценкой;

Или еще какие-то способы по усмотрению разработчика алгоритма.

Какая функция предпочтения окажется более эффективной, уменьшить количество пересматриваемых подмножеств и тем самым уменьшить количество промежуточной информации, можно определить только экспериментально.

Использование первой функции предпочтения может привести к длительному процессу ветвления, поскольку на верхних ярусах дерева будут подмножества, содержащие частичное решение с маленьким числом конкретизованных переменных (*k* – минимальные числа). С этих позиций вторая функция предпочтения как бы эффективнее. Оценки подмножеств близки по значению к рекорду. Придется доводить до самого нижнего яруса. Оценить просто так невозможно. Необходимо экспериментировать.

Ниже приведен пример решения двухмерной задачи о ранце, с целью демонстрации влияния изначального наличия рекордного значения *R*. Приоритетная очередь конкретизации компонент вектора решений соответствует непосредственной их нумерации. Чтобы подчеркнуть влияние приоритетной очереди, определяемой согласно приближенному комбинированному алгоритму, повторно решается тот же пример.

ПРИМЕР

Приведем результаты решения многомерной задачи о ранце, используя выше сказанные соображения. Пусть дано

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

$$\begin{cases} 4x_1 + 3x_2 + 2x_3 + 4x_4 + 2x_5 + 3x_6 + 5x_7 \le 10 \\ 2x_1 + 4x_2 + 2x_3 + x_4 + 4x_5 + 4x_6 + 3x_7 \le 8 \end{cases}$$

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

Задача решалась с целью показать, как влияет наличие рекордного значения целевой функции *R* на процесс отсеивания неперспективных вариантов. И второй случай, влияние наличия приоритетной последовательности компонент вектора решений тоже на количество не рассматриваемых вариантов.

Задача решается обычным методом ветвей и границ. Оценивание подмножеств вариантов ведется с использованием подхода расширения множества вариантов. Другими словами, доведения задачи до просто решаемой модели, содержащей исходную задачу. Решается две (два ограничения)нецелочисленных одномерных задачи о ранце. Следуя методу Дацинга, оптимальное решение достигается просто. Соотношение соответствующих коэффициентов целевой функции и ограничений (c_j / a_j) предопределяет порядок присвоения значений «1» при возможности или части единицы компонентам векторов решений для одномерных задач. Первая задача имеет вид

$$Z_1 = \max(5x_1 + 3x_2 + 4x_3 + 5x_4 + 2x_5 + 4x_6 + 5x_7)$$

при ограничениях

$$4x_1 + 3x_2 + 2x_3 + 4x_4 + 2x_5 + 3x_6 + 5x_7 \le 10$$

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

Определяется $\lambda_j = \dfrac{c_j}{a_j} ::= \dfrac{5}{4}, \dfrac{3}{3}, \dfrac{4}{2}, \dfrac{5}{4}, \dfrac{2}{2}, \dfrac{4}{3}, \dfrac{5}{5};$

Упорядочив a_j в порядке не возрастания, имеем последовательность конкретизации переменных

$$P_1 = \{3, 6, 1, 4, 7, 2, 5\}.$$

Процедура получения решения следующая

$$x_3 = 1, \quad 10 - 2 = 8;$$

$$x_6 = 1, \quad 8 - 3 = 5;$$

$$x_1 = 1, \quad 5 - 4 = 1;$$

$$x_4 = 1/4.$$

Само решение $X^1 = (1, 0, 1, 1/4, 0, 1, 0)$;

Значение целевой функции $Z_1^* = 4 + 4 + 5 + 5 \cdot 1/4 = 14{,}25$.

Аналогично решается вторая одномерная задача в постановке

$$Z_2 = \max(5x_1 + 3x_2 + 4x_3 + 5x_4 + 2x_5 + 4x_6 + 5x_7)$$

при ограничениях

$$2x_1 + 4x_2 + 2x_3 + x_4 + 4x_5 + 4x_6 + 3x_7 \le 8$$

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

Последовательность присвоения положительных значений

$$P_2 = \{4, 1, 3, 7, 6, 2, 5\}.$$

Вектор решения $X^2 = (1, 0, 1, 1, 0, 0, 1)$;

Значение целевой функции

$$Z_2^* = 5 + 4 + 5 + 5 = 19{,}0.$$

Оценка множества вариантов решения

$$\xi = \min(Z_1^*, Z_2^*) = (14{,}25; 19{,}0\} = 14{,}25.$$

Разбиение множества вариантов на подмножества-ветвление организовано последовательной конкретизацией компонент вектора решений согласно их непосредственной нумерацией.

Рекордное значение целевой функции R определяется согласно приоритетной очереди компонент вектора решений, используя приближенный комбинированный алгоритм. Очередь на присвоение значения «1» следующая

$$P = \{1, 6, 3, 4, 7, 2, 5\}.$$

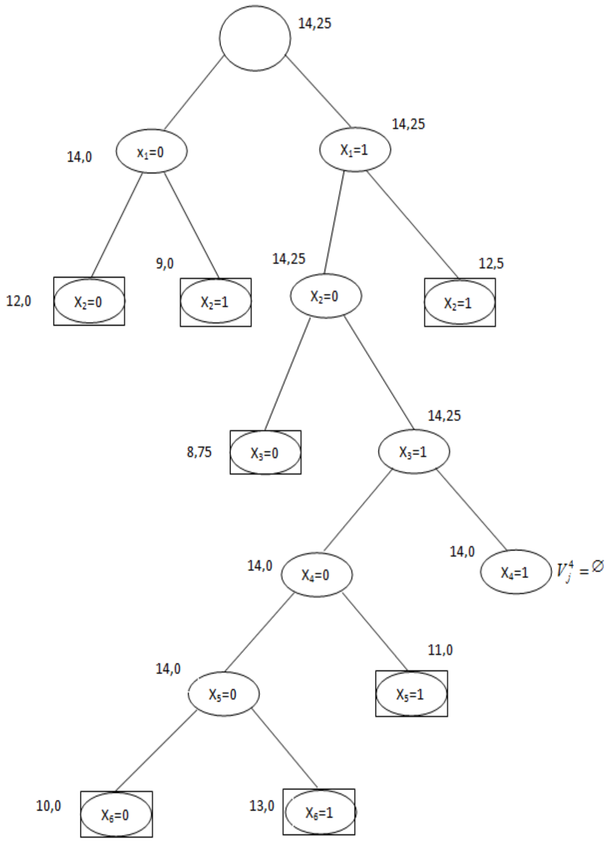

Рис.1 – Дерево решений 1

Компонентам x_1, x_6, x_3 присваивается значение «1». Больше не позволяет система ограничений. Значение целевой функции при векторе решений $X = (1, 0, 1, 0, 0, 1, 0)$ нетрудно определить $Z = 5+4+4 = 13$ оно воспринимается как изначальное рекордное значение *R*. Дерево решений, демонстрирующее влияние наличия изначального рекордного значения целевой функции приведено на рис.1. Использовано последовательное построение решение согласно нумерации компонент вектора решений. Рассмотренный пример дерева решений содержат 14 вершин. Реально рассматривалось только 6. Наличие рекордного значения целевой функции (R=13,0) позволило исключить из рассмотрения половину бесперспективных подмножеств. На рис.1 вершины дерева, подлежащие отсеиванию, отмечены. На рис.2. представлена работа алгоритма, содержащая больше модифицированных моментов.

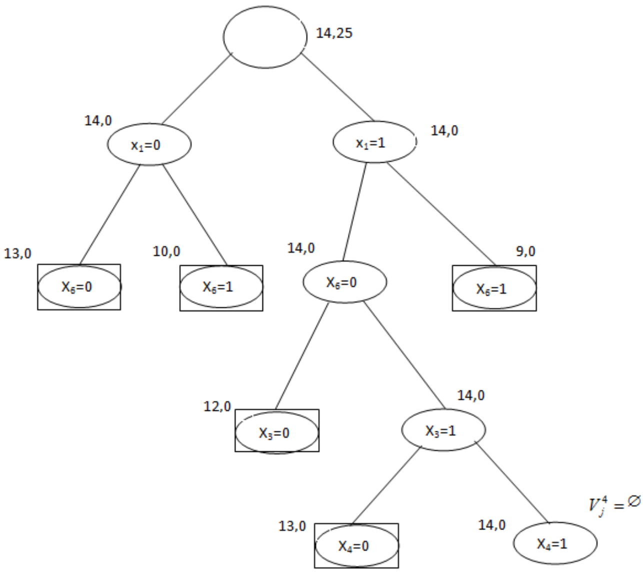

Рис. 2 – Дерево решений 2

Используется та же приоритетная последовательность компонент вектора решений при конкретизации значений, т.е. построение частичных решений ведется согласно этой очереди, а именно $P=(1, 6, 3, 4, 7, 2, 5)$. приоритетная очередь та же самая, полученная приближенным комбинированным алгоритмом, что и при определении рекордного значения. Позитивное влияние такого рода конкретизации компонент, очевидно. Количество отсеенных вершин дерева решений значительно увеличилось. Если учесть еще и момент проверки возможности определения значения «1» компоненте согласно очереди, то дерево решения имеет, лишь одну ветвь, определяющую оптимальное решение $X=(1, 0, 1, 0, 0, 1, 0)$. В первом случае, для достижения этого решения перебиралось значительно больше подмножеств, которые, в конечном счете, оказались бесперспективными. Размерность рассматриваемого примера не позволила продемонстрировать возможности функций предпочтения при поиске ветви с лучшей оценкой, чем имеющийся рекорд. Очевидно, что рекордное значение, определенное изначально при построении дерева решений, изменилось и стало равной 14,0. В принципе ветвь с оценкой 14,0 и содержащей частичное решение $\sigma^1 \equiv (0)$ можно было и не продолжать, поскольку чем ниже по дереву решений передвигаемся, тем меньше вероятность хотя бы сохранить значение оценки.

ЗАКЛЮЧЕНИЕ

Комбинаторные методы, в том числе и метод ветвей и границ, являются открытыми методами. В них можно включать различные модифицирующие блоки, увеличивающие эффективность их работы. Однако следует отметить, что точные комбинаторные алгоритмы все еще не являются быстродействующими. Они с трудом используются для решения задач большой размерности. Кроме того, комбинаторные алгоритмы имеют такую организацию работы вычислительных процедур, что необходимо хранить большое количество информации. Это тоже усложняет работу при поиске оптимального варианта. Эффективное использование алгоритмов метода ветвей и границ в практической деятельности требует либо кардинального изменения вычислительных процедур, либо такую их организацию, чтобы создать возможность получить если не оптимальное, то хотя бы приемлемое решение.

Данная работа посвящена обобщению некоторых модифицирующих моментов, увеличивающих количество отсеиваемых подмножеств вариантов. Это делается, используя различного типа разработок при создании приближенных, а так же вероятностно-приближенных. Случайный поиск компоненты подлежащей конкретизации на очередном шаге работы алгоритма сохраняет детерминированность выполнения остальных процедур, оставляя возможность получить оптимальное решение. Естественно, если была такая установка получения решения. Приведенный пример демонстрирует влияние на эффективность работы алгоритма метода ветвей и границ с использованием рекордного значения. R можно получить оперативно, используя идею любого приближенного алгоритма. Второй процедурный прием решения приведенной задачи обращает внимание на то, насколько важен прием приоритетного упорядочения компонент вектора решений. Включается в работу по совершенствованию алгоритма и момент определения множества хороших компонент. Эффективность использования функций предпочтения при возврате на вершину с оценкой больше рекорда проверить не удалось из-за маленькой размерности примера.

На одном примере небольшой размерности выводы делать нет обоснования. Однако, это так же моменты на что действительно следует обратить внимание. В дальнейших разработках следовало бы проверить предлагаемое на большом количестве примеров. Необходимо провести числовой компьютерный эксперимент. На основе результатов эксперимента создать вычислительный алгоритм решения задач большой размерности, т.е. довести до практического использования.

СПИСОК ЛИТЕРАТУРЫ

[1] Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование. – М.: Физматгиз, 2007. – С. 304.

[2] Михалевич В.С. Последовательные алгоритмы оптимизации и их применение. I, II // Кибернетика, 1965. – № 1,2.

[3] Юхименко Б.И., Козина Ю.Ю. Сравнительная характеристика алгоритмов метода ветвей и границ для решения задач целочисленного линейного программирования. Труды Одесского политехнического университета, 2005. – Вып.2. – С. 199 – 204.

[4] Balas E. An additive algorithm for soving linear programs with zero-one variables // Operat.Res, 1965. – V.13. – №4. – P. 517 – 546.

[5] 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.

[6] Юхименко Б.И., Волкова В.П. Приближенные алгоритмы решения задачи о многомерном ранце // Дослідження в математиці і механіці, 2017. – Т.22. – Вип.2 (30). – С. 104 – 115.

[7] Гладков Л.А. Куряйчик В.М. Генетические алгоритмы. – М.: ФИЗМАТЛИТ, 2006 .– 230 с.

[8] Штовба С.Д. Муравьиные алгоритмы: Теория и приложения // Программирование, 2005. – №4. – С.3 –18.

[9] Леванова Т.В., Лореш М.А. Алгоритмы муравьиной колонии и имитации отжига для задачи о p-медиане. // Автоматика и телемеханика, 2004. – №3. – С. 80 – 88.

[10] Сигал И.Х. Оценки параметров алгоритмов ветвей и границ для задач дискретной оптимизации большой размерности // Известия РАН. Теория и система управления, 2005. – №4. – С. 96 – 101.

[11] Кочетов Ю.А. Вероятностные методы локального поиска для задач дискретной оптимизации. //Дискретная математика и приложения, 2001. – С. 84 –117.

[12] Юхименко Б.И. Модификации метода ветвей и границ для решений задач целочисленного линейного программирования и их эффективность // Інформатика та математичні методи в моделюванні, 2015. – Т.5. – №.1. – С. 84 – 91.

[13] Юхименко Б.И., Ткаленко О.Ю. Алгоритм муравьиной колонии для многомерной задачи о ранце // Математичні методи обробки даних, 2019. – №2. – Т.21. – С. 3 – 12.

SOME INFORMATION PROBLEMS OF COMBINATORY LINEAR OPTIMIZATION

Yukhimenko B.

Discrete optimization is one of the most requested disciplines with which decision-making procedure is associated. Its use is associated with many problems that are procedurally voluminous. The construction of a mathematical model, the selection of really significant factors, the conversion to types of existing mathematical problems and the further computer realization is a rather laborious process. Existing mathematical methods of both analytical type and combinatorial type still belong to the complexity class NP. Although modern computing techniques is a heavy-duty and well equipped with mathematical software for solving problems of a particularly large dimension it is practically impossible to realize in the case of the requirement to get the optimal solution. In addition, combinatorial methods and algorithms for solving discrete optimization problems are associated with a large amount of intermediate information that necessary to obtain an optimal or at least sufficiently good solution. In this regard, the development of new, modifying blocks and modules to be included in the schemes, existing algorithms is currently requests, relevant and has both scientific and applied value.

This elaboration represents, in a sense, a systematization of the problems of the branch and bound method for solving the linear programming problem with Boolean variables in the main, and the multidimensional knapsack problem in particular. The main points of the method that can be modified are listed. The possibility to connect the moments of the approximate algorithms as the preparatory part of the exact algorithm is shown. A numerical example is solved that demonstrates the positive effect of two algorithmic procedures on the efficiency of the method. Using the record value of the objective function reduces by half the number of vertices being reviewed, and the introduction of a priority sequence of components for concretization still eliminates a significant number of vertices.

Keywords. Discrete optimization, the record, sequential construction of solutions, approximates algorithms, concretization, priority sequence.