

## К ВОПРОСУ ОБ УВЕЛИЧЕНИИ СКОРОСТИ СХОДИМОСТИ АЛГОРИТМОВ МЕТОДА ВЕТВЕЙ И ГРАНИЦ

К.э.н. Б.И. Юхименко, А.А. Сиенко

Одесский национальный политехнический университет

Украина, г. Одесса

alena.sienko@yandex.ru

Дискретная оптимизация, хотя и относящаяся к молодым наукам занимает достойное место в научном и бытовом мире. Принятие оптимальных решений в различных областях деятельности человека стало классической проблемой. Однако как процедура формализации проблем принятия решений, так и разработка эффективных методов и алгоритмов оперативной реализации моделей не является совершенной. Совершенствование методов решения задач дискретной оптимизации все еще остается открытой процедурой и любая разработка в этом направлении занимает достойное место, если она приводит к новым подходам, идеям и совершенствованию того, что уже разработано.

Даная задача посвящена совершенствованию одного из наиболее приемлемых методов дискретной оптимизации - методу ветвей и границ, как комбинаторному методу направленного перебора вариантов. Сам метод, появившийся в 1960 в работе Лэнд и Дойг [4], имел свое второе рождение, связанное с появлением работы Литтла, Мурти и Кэрел [5], посвященной задаче о коммивояжере. Метод легко реализуется для решения любой дискретной оптимационной задачи. Замена полного перебора вариантов частичным перебором как бы подталкивает исследователей вносить новые идеи к совершенствованию процедуры отбора более «удачных» частей множества решений и поиска среди них оптимального решения.

Всю атрибутику метода можно разделить на три основных модуля: разбиение множества вариантов на подмножества, оценивание подмножеств и своеобразное построение признака оптимальности, полностью зависящего от количественных значений оценок подмножеств, но являю-

щимся стандартным для любых алгоритмов метода.

Скорость сходимости алгоритмов метода ветвей и границ значительно зависит от точности получения оценок т.е. от близости к значению целевой функции варианта решения. Не мало важное значение для скорости сходимости имеет и процедура ветвления. Концентрация более удачных вариантов в отдельное подмножество и дальнейшая работа с этим подмножеством несомненно приведет к меньшему количеству перебираемых вариантов. Таким образом определяется поле действия с целью ускорения работы алгоритмов метода ветвей и границ. Модификации подлежат процедура ветвления и оценивания.

В последнее время основное внимание уделяется методу последовательного построения решения [3] как способу разбиения множества вариантов на подмножества. Конкретизация компонент выбора решений позволяет выделять параметр классификации вариантов, разделяя их на подмножества с определенной количественной характеристикой. Вектор решений как бы «собирается» из отдельных компонент, значение которых определяются (конкретизируются) как числа из интервала значений, определяемого системой ограничений, т.е. математическим описанием множества вариантов. Очередность выбираемых к конкретизации компонент влияет на скорость сходимости алгоритмов [1].

Рассмотрим задачу линейного программирования с булевыми переменными

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

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = \overline{1, m}; \quad (2)$$

$$x_j \in \{0, 1\}, j = \overline{1, n} \quad (3)$$

Частичным решением длиной  $k$  будем называть такой  $\sigma_k$ , в котором

конкретизировано  $k$  компонент вектора решений  $x = \{x_1, x_2, \dots, x_n\}$ .

Оценка множества вариантов  $G$ , определяемая ограничениями (2) - (3), осуществляется путем решения  $m$  нецелочисленных одномерных задач о ранце в подстановках

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

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i$$

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

Пусть  $x_i^* = (x_{i1}^*, x_{i2}^*, \dots, x_{in}^*)$  оптимальное решение  $i$ -ой задачи. Тогда оценка множества  $G$  –  $\xi(G)$  определяется как

$$\xi(G) = \min_i \sum_{j=1}^n c_j x_{ij}^*.$$

Такой способ оценивания был разработан давно. Подробнее можно прочесть [2].

Если иметь частичное решение  $\sigma_k$  и ему соответствующее множество индексов  $I_x^k$  конкретизированных компонент, то оценка множества  $G^k$ , содержащая частичное решение  $\sigma_k$  оценивается так: решаются  $m$  задач в подстановках

$$Z_i^{k*} = \max \sum_{j \in I_x^k} c_j x_j + \sum_{j \notin I_x^k} c_j x_j$$

при ограничениях

$$\sum_{j \in I_x^k} a_{ij} x_j \leq b_i - \sum_{j \notin I_x^k} a_{ij} x_j$$

$$0 \leq x_j \leq 1, j \in I_x^k.$$

и тогда  $\xi(G^k) = \min_i Z_i^{k*}$ .

Более подробно рассмотрим процедуры разбиения множества вариантов на подмножества. Это многошаговый процесс, на каждом шаге которого выбранное подмножество разбивается на два. Оба подмножества содер-

жат частичное решение одинаковой длины, однако конкретизируемая переменная на данном шаге принимает значение «нуль» либо «единица».

При первом подходе на  $k$ -ом шаге конкретизации определяется значение  $x_k$ -ой переменной. Пусть имеем частичное решение  $\sigma_{k-1}$ . Определяются два частичные решения  $\sigma_k^1 = \{G_{k-1} \cup x_k = 1\}$  и  $\sigma_k^0 = \{G_{k-1} \cup x_k = 0\}$ , которые содержатся в соответствующих подмножествах  $G_k^1$  и  $G_k^0$ . Другими словами, порядок конкретизации переменных соответствует нумерации компонент в векторе решений.

Как обычно, процесс ветвления представляется графически в виде дерева решений. На каждом ярусе этого дерева осуществляется конкретизация одной переменной. На первом ярусе конкретизируется  $x_1$ , на втором  $x_2$  и т.д., пока не будет получено решение – пока длина частичного решения не станет равна  $n$ . Полученное решение проверяется на оптимальность. Если оно не является оптимальным, то, как обычно в методе ветвей и границ, возвращаемся к вершине с наилучшей оценкой. Она находится на определенном ярусе, скажем на  $k$ -ом. Тогда конкретизации подлежит  $k+1$ -ая переменная.

При втором подходе конкретизируемая переменная выбирается специальной процедурой. Определяется так называемое множество «хороших» компонент, то есть компонент, которые отдельно взятые согласно текущей системе ограничений могут принимать значение «единица». Полученные компоненты оцениваются исходя из того, какая неувязка в системе ограничений получается при единичном значении и какое место по величине коэффициента вектора стоимостей занимает оцениваемая переменная. Формально это представляется следующим образом. Обозначим через  $b_i^k$  ( $i = \overline{1, m}$ ) текущее значение компонент вектора ограничений на  $k$ -ом шаге ветвления. Количество значение

$$b_i^k = b_i - \sum_{j \in I_x^k} a_{ij} x_j.$$

Множество индексов хороших компонент определяется следующим образом

$$V_j^k = \{j / (b_i^k - a_{ij}) \geq 0, i = \overline{1, m}\}, j \in I_x^k.$$

Компоненты  $x_j$  для  $j \in I_x^k$  оцениваются по формуле

$$P_j = \sum_{i=1}^m (b_i^k - a_{ij}) \cdot \sqrt{c_j / \min c_j}, j \in V_j^k.$$

Конкретизации на данный момент подлежит та компонента  $j_0$ , для которой этот балл максимальный

$$P_{j_0} = \max_{j \in V_j^k} P_j.$$

Вычислительная сложность предлагаемого алгоритма, модифицирующего процедуру ветвления может быть доказана теоретически с указанием порядка числа, предопределяющего число итераций. Это дальнейшая работа - работа на инициативу. Более простой способ это экспериментальная оценка скорости сходимости. Эта работа выполняется и результаты будут обнародованы в следующей публикации.

Проведенный числовой эксперимент показал, что второй способ значительно быстрее приводит к оптимальному решению.

С целью демонстрации работы алгоритмов, отличающихся способами порядка конкретизации приведем небольшой числовой пример и дерева их решения (см. рис. 1 и рис. 2). Имеем задачу в подстановке

$$Z = \max(2x_1 + x_2 + 3x_3 + 3x_4 + 2x_5)$$

при ограничениях

$$\begin{cases} x_1 + 2x_2 + 4x_3 + 3x_4 + 5x_5 \leq 8 \\ 3x_1 + 4x_2 + 3x_3 + 4x_4 + 2x_5 \leq 9 \end{cases}$$

$$x_j \in \{0, 1\}, j = \overline{1, 5}.$$

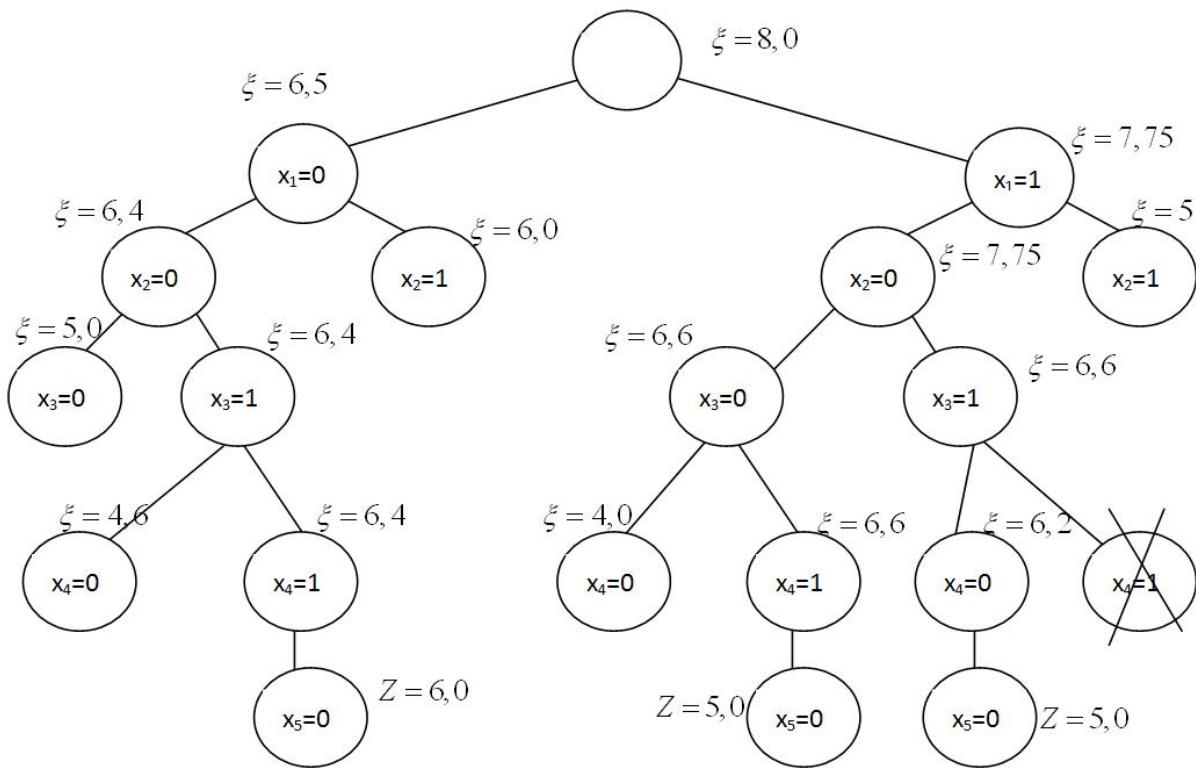


Рис. 1. Дерево рішень, представляюче перший подхід конкретизації

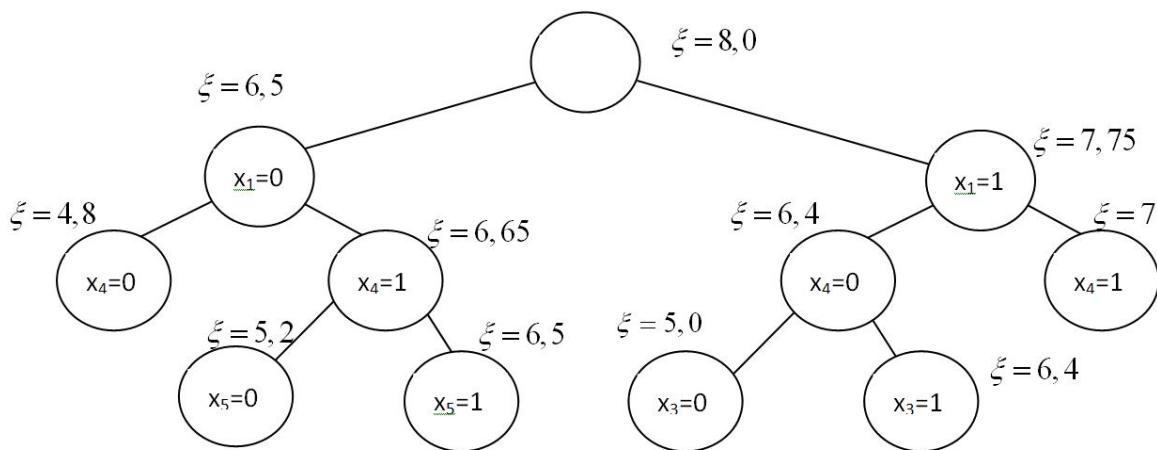


Рис. 2. Дерево рішень, определяющее второй способ конкретизации  
Действительно приведенный второй способ конкретизации переменных значительно уменьшает вычислительную сложность алгоритма метода ветвей и границ для решения задач линейного программирования с булевыми переменными. Согласно примеру при первом способе ветвления рассмотрено 19 вершин дерева решений так как при втором, разработанным

авторами, дерево решений содержит 10 вершин. Вычислительная сложность уменьшилась почти в два раза.

Необходимо провести большую экспериментальную работу, на основе которой можно будет определить теоретически вычислительную сложность предлагаемого алгоритма.

#### ИСПОЛЬЗОВАНЫЕ ИСТОЧНИКИ

1. Юхименко Б.И. Сравнительная характеристика метода ветвей и границ для решения задач целочисленного линейного программирования / Б.И. Юхименко, Ю.Ю. Козина // Труды Одесского политехнического университета. – 2005. – 2(24). – с. 199 – 204.
2. Юхименко Б.И. О некоторых алгоритмах решения задач линейного программирования с булевыми переменными // Кибернетика. – 1989. - №5. – с. 141 – 143.
3. Михалевич В.С. Последовательные алгоритмы оптимизации и их применение. – Кибернетика, 1965, № 1, с. 45 – 55.
4. Land A.H., Doig A.G., An automatic method of Solving discrete programming problems. Econometrica, 1960, 28, №3, 497 – 520.
5. Little I.D.C., Murty K.G., Sweeney D.W., Karel C., An algorithm for the travelling salesman problem. Operat. Res., 1963, 11, №6, 972 – 989.