УДК 519.852: 004.424.27.021

Informatics and Mathematical Methods in Simulation
Vol. 2 (2012), No. 2, pp. 173-179

ОБОБЩЕНИЕ АЛГОРИТМОВ МЕТОДА ВЕТВЕЙ И ГРАНИЦ ДЛЯ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С БУЛЕВЫМИ ПЕРЕМЕННЫМИ

Б.И. Юхименко

Одесский национальный политехнический университет,
просп. Шевченко, 1, Одесса, 65044, Украина; e-mail: pm1987pm@gmail.com

В работе проведен анализ и предложен способ увеличения скорости сходимости алгоритмов метода ветвей и границ путем расширения атрибутики метода за счет введения понятия функции предпочтения. Рассмотрены два типа функций предпочтения: обычные и рандомизированные. Показано, что рандомизированные функции предпочтения приводят к лучшему результату в смысле вычислительной сложности, особенно с учётом длины частичного решения.

Ключевые слова: целочисленное линейное программирование, метод ветвей и границ, функция предпочтения

Введение

Дискретная оптимизация, как достаточно новый математический аппарат, широко используется для принятия решений в различных областях деятельности человека. Сферы применения этой науки не ограничиваются математическим моделированием в организационном управлении. Дискретные задачи возникают в транспорте, при решении самых различных задач в распределении – размещении неделимых объектов, в задачах о покрытии на графах и во многих других проблемах, вплоть до организации компьютерных интернет-сетей.

Что касается методов решения таких задач, то основное внимание уделяется решению отдельной части дискретных задач – задачам целочисленного линейного программирования (ЦЛП). Хорошо известно, что методы решения задач ЦЛП делятся на три основных класса: метод отсекающих плоскостей, метод ветвей и границ и приближения, эвристический метод. Компьютерная их реализация для широкого пользователя сконцентрирована на методах Гомори (отсекающих плоскостей). Комбинаторные методы (методы ветвей и границ) программно менее реализованы, хотя их легко можно модифицировать для конкретного случая решения прикладной задачи, учитывая её специфическую структуру в математической модели, а также специфику фактической исходной информации. Методы ветвей и границ легко модифицируются при превращении их из точных в приближенные методы. Ещё один позитивный момент этого класса методов состоит в том, что логическая их структура несложная, легко реализуемая программно, и вычислительная сложность предопределяется числом итераций. Именно этот момент вызывает необходимость обратить особое внимание на скорость сходимости алгоритмов метода.

Цель статьи и постановка исследования

Целью работы является анализ и увеличение скорости сходимости алгоритмов метода ветвей и границ.

Для достижения цели в работе решаются *задачи*

- расширения атрибутики метода ветвей и границ за счет введения понятия функции предпочтении;
- предлагается ряд функций предпочтения, способствующих увеличению скорости сходимости алгоритмов метода ветвей и границ.

Основная часть

Атрибутика метода ветвей и границ состоит из следующего:

- оценка множества вариантов – число, показывающее, что значение целевой функции для любого варианта из множества не может быть лучше (меньше или больше) чем данное число;
- ветвление – разбиение множества вариантов на подмножества. Это процедура выбора параметра, на основе которого множество вариантов разбивается на ряд подмножеств;
- признак оптимальности – средство, предопределяющее, что лучше найденного варианта не существует во множестве вариантов.

Все эти три момента предопределяют скорость сходимости алгоритмов метода. При алгоритмизации метода от способа оценки множества вариантов требуется, что бы она как можно была бы приближена к значению целевой функции для локального оптимального варианта. Такое требование диктует сам признак оптимальности метода. Однако чем точнее получается оценка, тем вычислительная сложность алгоритма больше. Часто приходится «жертвовать» точностью за счёт упрощения процедуры получения оценок. Вычислительная сложность уменьшается, но увеличивается число итераций при получении оптимального варианта. Частичный перебор вариантов несколько приближается к полному перебору, что не желательно, в особенности для задач большой размерности.

Что касается разбиения множества вариантов на подмножества, то очень важно подобрать параметр, разделяющий варианты на подмножества. В алгоритме Лэнда и Дойга, как бы исключаются из рассмотрения нецелочисленные значения компоненты вектора решений в области её возможного оптимального значения.

При последовательном построении решений [1] параметром деления является конкретизация значений определённой, выбранной каким-то образом компоненты. В задачах линейного программирования с булевыми переменными это осуществить достаточно просто. Если конкретизации подлежит компонента X_{j0}, то множество вариантов делится на два подмножества, одно из которых содержит все варианты, в которых $X_{j0} = 1$.

Сущность признака оптимальности состоит в том, что если получен вариант решения, то следует удостовериться, что, во-первых, ни одно из имеющихся подмножеств не содержат лучше варианта и, во-вторых, что в подмножестве, которому принадлежит проверяемый на оптимальность вариант, не существует варианта лучше, чем проверяемый. Первое требование признака оптимальности очевидное: подмножество, содержащее проверяемый вариант, должно иметь наилучшую по значению оценку, так как при дальнейшем разбиении на более мелкие подмножества значение ни оценки, ни тем самым, значение целевой функции не улучшается. Второй момент – второе требование очень чувствительное по отношению к точности оценки подмножества. Требование совпадения значения целевой функции проверяемого на

оптимальность варианта со значением оценки возможно, в основном, только тогда, когда оценка достаточно точно получена (имеется ввиду, близко к значению целевой функции).

Если полученный вариант не удовлетворяет признаку оптимальности, необходимо продолжить поиск оптимального варианта. Ставится вопрос, какому подмножеству отдать предпочтение. По логике вещей, выбирается подмножество с наилучшей оценкой и продолжается чередование получения оценок с разбиением множества на более мелкие подмножества в поиске лучшего варианта. Опять таки, рассматривая логически, более лучшие (большие в задачах максимизации и меньшие в задачах минимизации) оценки имеют те подмножества, в которых больше количество вариантов. При последовательном построении решения – это подмножество, содержащее меньшее количество конкретизированных переменных – компонент вектора решений.

Предполагается расширить атрибутику метода ветвей и границ ещё одним понятием – функция предпочтения. Функция предпочтения – это способ выбора подмножества, не подвергавшегося разбиению, по определённым правилам, базирующимся на величине оценок.

Рассмотрим более распространенные и проверенные способы оценивания, ветвления, а так же предположим рад функций предпочтения, которые, на наш взгляд, как-то должны способствовать увеличению скорости сходимости алгоритмов метода ветвей и границ.

Итак, пусть имеется задача линейного программирования с булевыми переменными в постановке

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

при ограничениях

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

$$0 \le x_j \le 1 , \quad j = \overline{1,n} ; \tag{3}$$

$$x_j \in Z , \quad j = \overline{1,n} , \tag{4}$$

где Z – множество целых чисел.

Область определённости задачи, задаваемая требованиями (2)-(4) расширяется путём отбрасывания требований (4). Задача в постановке (1)-(3) является соответствующей задачей линейного программирования по отношению к задаче целочисленного линейного программирования в постановке (1)-(4). Очевидно, это экстремальное значение целевой функции $Z(x)$ для x из области определённости (2)-(3) всегда будет не хуже экстремального значения $Z(x)$, когда x из области определённости (2)-(3)-(4). Можно утверждать, что получение оценки множества целочисленных вариантов можно осуществить путём решения соответствующей задачи линейного программирования. Методы решения задач линейного программирования не подлежат обсуждению.

Известно [2], что процедуру получения оценок множества целочисленных вариантов можно привести к решению m нецелочисленных задач о ранце. Решаются задачи в постановках

$$Z_i = \max \sum_{j=1}^{n} c_j x_j \,, \tag{5}$$

при ограничениях

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

$$0 \le x_j \le 1 \,, \quad j = \overline{1,n} \,; \tag{7}$$

когда $i = \overline{1,m}$, т.е. имеется общая целевая функция (5) для всех задач, а основные ограничения типа (2) берутся по одной, составляя задачу о ранце.

Оценка множества целочисленных вариантов $Z(x)$ ограничивается сверху значением $Z_{i_0}(x)$, где

$$Z_{i_0}(x) = \min_{i=\overline{1,m}} Z_i$$

Рассмотрим некоторые способы осуществления процедуры ветвления, используя идею последовательного построения решения. В данном случае конкретизируются значения переменных, переходя от одного яруса в дереве решений к другому. Можно сказать, что вектор решений, как бы собирается в процессе ветвления. И так, пусть величина k предопределяет количество конкретизированных переменных, значения которых фиксируются в частичном решении δ^k, а номера компонентов записывают во множестве I_x^k. И так $I_x^k = \{ j \mid x_j \textit{ конкретизировано} \}$. К примеру, если $I_x^3 = \{3,4,6\}$, а $\delta^3 = \{0,1,0\}$, то это значит, что $x_3 = 0$, $x_4 = 1$, $x_6 = 0$; а остальные компоненты пока не зафиксированы.

Известны [3] и экспериментально оценены [4] два способа выбора конкретизируемых переменных. Первый из них руководствуется лексикографической последовательностью переменных вектора решений. Первым конкретизируется x_1, вторым – x_2 и т.д. Второй способ навеян идеями алгоритма Балаша [6] и учитывает величину «вклада» переменной в значение целевой функции и величину неувязки в системе ограничений. Предположим, что имеет частичное решение длины k. Обозначаем его через δ^k и соответствующее множество индексов конкретизированных переменных I_x^k. Тогда вектор ограничений, состоящий из компонент b_i на $(k+1)$ шаге конкретизации определяется так:

$$b_i^k = b_i - \sum a_{ij} x_j \,,$$

$$i = \overline{1,m} \,, \quad j \in I_x^k \,, \quad x_j \in \delta_k \,.$$

На основании величин b_i^k определяется, так называемое, множество «хороших» компонент, компонент, которые могут ещё принимать значение «1».

$$V_j^s = \left\{ \frac{j}{b_i^k - a_{ij}} \geq 0 \right\}, \quad j \notin I_x;$$

где s предопределяет номер итерации (см. ниже).

Далее оцениваются все компоненты, номера которых находятся во множестве V_j^s, а именно определяется P_j согласно формуле:

$$P_j = \sum_{i=1}^{m} \left(b_i^k - a_{ij} \right) \times \sqrt{\frac{C_j}{\min\limits_{j \in V_j^s} C_j}}, \quad j \in V_j^s.$$

Предположим, что $P_{j_0} = \max\limits_{j \in V_j^s} P_j$. Тогда конкретизации подлежит компонента с номером j_0. Формируется множество $I_x^{k+1} = \left\{ I_x^k V_{j_0} \right\}$ и два частичных решения

$$\delta_{k+1}^0 = \left\{ \delta_k \cup \left(x_{j_0} = 0 \right) \right\} \quad \text{и} \quad \delta_{k+1}^1 = \left\{ \delta_k \cup \left(x_{j_0} = 1 \right) \right\}.$$

Оцениваются подмножества, содержащие частичные решения δ_{k+1}^0 и δ_{k+1}^1 по выше предложенной методике и подмножество с лучшей оценкой выбирается для дальнейшего разбиения. Другими словами определяется частичное решение δ_{k+1}. Если окажется, что $V_j^s = \varnothing$, то все неконкретизированные компоненты частичного решения определяются как нулевые, частичное решение становится вариантом решения и проверяется на оптимальность. Если вариант окажется не оптимальным, то необходимо выбрать вершину дерева решений, которая не подвергалась разбиению, возможно, содержит оптимальное решение и, быстрее всего, приведёт к нему или убедимся, что оптимального решения в данном подмножестве не имеется. Ставится вопрос, какой вершине в дереве отдать предпочтение. Выбор определяется алгоритмически при помощи некоторых функций, основанных на величине оценок, длине частичного решения и называемых функциями предпочтения [5].

Рассматриваются два типа функций предпочтения: обычные функции и рандомизированные. Обычные функции – это правило выбора вершины, осуществляется либо по максимальному значению оценки либо по максимальному количеству конкретизированных переменных вектора решений. Выбор вершины с максимальной оценкой – это классический приём. Что касается выбора согласно величине «заполненности» частичного решения, то, во-первых, вычислительная сложность уменьшается при доведении частичного решения до варианта решения задачи, либо к убеждению, что в данном направлении нет оптимального варианта, просмотр следует остановить. Во-вторых, эффект (в смысле вычислительная сложность) возможен за счёт уменьшения просматриваемых вершин.

При использовании рандомизированных функций предпочтения для выбора вершины в дереве, включается элемент случайности. Выбирается определённое количество вершин с наибольшими значениями оценок подмножеств вариантов, их значения приводятся к шкале, определяемого интервалом $(0,1)$, который делится на участки, соответствующие относительной величине оценок. Генерируется значение случайной величины равномерно – распределённой из интервала $(0,1)$. Соответствие значения полученной случайной величины участку интервала, предопределяет вершину дерева решений, с которой необходимо продолжить поиск. Итак, подлежит рассмотрению:

1) обычная функция предпочтения – выбор вершины осуществляется согласно максимальному значению оценок среди:

- всех вершин, не подвергавшихся разбиению;
- среди вершин, не подвергавшихся разбиению и содержащих частичное решение длины не меньше r;

2) рандомизированные функции предпочтения – выбор вершины осуществляются случайно среди:

- k не подвергавшихся разбиению вершин и имеющих наибольшие оценки;
- k не подвергавшихся разбиению вершин с наибольшими оценками и содержащих частичное решение длины не меньше r.

Величина k и r могут варьироваться и иметь самую различную величину.

Заключение

По результатам проведенного эксперимента можно сделать следующий вывод: рандомизированные функции предпочтения приводят к лучшему результату в смысле вычислительной сложности, особенно с учётом длины частичного решения. В настоящий момент усилия авторов направлены на поиск новых типов функций предпочтения.

Список литературы

1. Михалевич В.С. Последовательные алгоритмы оптимизации и их применение. I. II. / В.С. Михалевич // Кибернетика. – 1965. − №1. – С. 45-55; №2. – С. 85-88.
2. Юхименко Б.И. Разработка и обоснование некоторых схем и методов дискретной оптимизации в автоматизаци Михалевич В.С. Последовательные алгоритмы оптимизации и их применение. I. II. / В.С Михалевич // Кибернетика. – 1965. − №1. – С. 45-55; №2. – С. 85-88. и функций распределения – размещения в АСУ: Автореферат диссертации на соискание учёной степени кандидата экономических наук. – Киев, 1982. – 20 с.
3. Юхименко Б.И. Ускоренный алгоритм метода ветвей и границ для решения задачи целочисленного линейного программирования // Труды Одесского политехнического университета. – 2004. – Вып. 2. – С. 223-226.
4. Юхименко Б.И. Сравнительная характеристика алгоритмов метода ветвей и границ для решения задач целочисленного линейного программирования / Б.И. Юхименко, Ю.Ю. Козина // Труды Одесского политехнического университета. – 2005. – Вып. 2. – С. 199-204.
5. Юхименко Б.И. Об эффективности функций предпочтения в алгоритмах целочисленного линейного программирования // Литовский математический сборник. – Вильнюс, 1983. – Том 23, № 1. – С. 134-140.
6. Balas E. An Additive Algorithm for Solving Linear Programs with Zero-One Variables // Operations Research. – 1965. – Vol.13, Num.4. – PP. 517-546.

УЗАГАЛЬНЕННЯ АЛГОРИТМІВ МЕТОДУ ГІЛОК І МЕЖ ДЛЯ РІШЕННЯ ЗАДАЧ ЛІНІЙНОГО ПРОГРАМУВАННЯ З БУЛЕВИМИ ЗМІННИМИ

Б.І. Юхименко

Одеський національний політехнічний університет,
просп. Шевченка, 1, Одеса, 65044, Україна; e-mail: pm1987pm@gmail.com

В роботі проведено аналіз та запропоновано спосіб збільшення швидкості збіжності алгоритмів методу гілок і меж шляхом розширення атрибутики методу за рахунок введення поняття функції переваги. Розглянуто два типи функцій переваги: звичайні та рандомізовані. Показано, що рандомізовані функції переваги приводять до кращого результату в сенсі обчислювальної складності, особливо з урахуванням довжини часткового рішення.
Ключові слова: цілочисельне лінійне програмування, метод гілок і меж, функція переваг

GENERALIZED BRANCH AND BOUND ALGORITHM FOR SOLVING LINEAR BOOLEAN PROGRAMMING PROBLEMS

Birute I. Yukhimenko

Odessa National Polytechnic University,
1 Shevchenko Ave., Odessa, 65044, Ukraine; e-mail: pm1987pm@gmail.com

Proposed method for increasing the convergence speed of the branch-and-bound algorithm by introducing the concept of preference functions. Described two types of preference functions – common and randomized. According to factual evidence randomized preference functions lead to a better result in sense of computational complexity, especially considering the length of the partial solution.
Keywords: integer linear programming, branch and bound, preference function