

МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО И КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ С БУЛЕВЫМИ ПЕРЕМЕННЫМИ

О.Н. Симашкевич
(представил проф. С.А. Соколов)

Предложен новый метод решения задач линейного и квадратичного программирования с булевыми переменными, на основе которого был разработан ϵ -оптимальный алгоритм. Показано, что при использовании данного метода возможно заранее задавать время и погрешность решения.

Постановка проблемы. Для нахождения приближенного решения задач линейного и квадратичного программирования используются алгоритмы локального поиска, вычислительная сложность которых лежит в пределе 15%, при этом они обладают достаточно высокой временной сложностью. При управлении сложными системами часто возникает задача получения как можно более точных решений за допустимое время. Поэтому является актуальной разработка алгоритмов, обеспечивающих малую погрешность и малую временную сложность.

Анализ последних достижений и публикаций. В настоящее время наиболее эффективными методами решения рассматриваемого класса задач являются методы «рангового подхода» [1, 2] и «ветвей и границ» [3]. Недостатком алгоритмов, основанных на методе «ветвей и границ», является их экспоненциальная сложность, а в алгоритмах, основанных на методе «рангового подхода», неопределено, каким образом реализовать принцип выделения «коридора» в задачах квадратичного программирования.

Постановка задачи. Представляет интерес разработка метода решения задач квадратичного и линейного программирования с булевыми переменными, в котором задается требуемое время решения задачи $T_{\text{треб}}$ и нужно ответить на вопрос, можем ли мы решить задачу с погрешностью, не превышающей некоторой величины $\epsilon_{\text{зад}}$.

Основной материал. Для решения указанной задачи был разработан метод, на основе которого был построен ϵ -оптимальный алгоритм для решения указанной задачи. Метод основан на комбинированном использовании идей «рангового подхода» и метода «ветвей и границ». Суть метода заключается в том, что при формировании множества возможных решений на основе «рангового подхода» отсеиваются неперспективных вариантов внутри мно-

жеств происходит на основе метода «ветвей и границ». При этом для определения верхней и нижней оценки при реализации метода «ветвей и границ» были использованы разработанные для алгоритмов метода «рангового подхода» оптимистический (P_0) [1] и гарантированный прогнозы (P_G) [2] соответственно. В алгоритме для корректировки времени решения используется величина, названная **степенью огрубления** ($\varepsilon_{огр}$). Данная величина измеряется в процентах и показывает степень приближения верхней оценки к нижней оценке, путем принудительного уменьшения верхней. При изменении $\varepsilon_{огр}$ время решения задачи обратно пропорционально погрешности решения.

Выбор прогнозов обоснован тем, что P_0 является верхней оценкой решения, основанного на векторе, для которого он определяется, P_G показывает какое решение, основанное на рассматриваемом векторе, обязательно будет получено. Следовательно, P_G больше или равен нижней оценке. Это позволяет сузить «коридор» [1] и улучшить отсечение перспективных векторов.

При $\varepsilon_{огр} = 0$ алгоритм будет иметь экспоненциальную временную сложность. И возможна такая ситуация, при которой время решения задачи превысит $T_{треб}$. Поэтому для каждой размерности задачи, которую планируется решать, и в зависимости от допустимого времени необходимо выбирать персональное значение $\varepsilon_{огр}$.

Величина коэффициента определяется экспериментальным путем для необходимой размерности задачи и $T_{треб}$. При дальнейшем использовании алгоритма по результатам решенных задач величины возможно корректировать коэффициентом.

Необходимо отметить, что, варьируя величиной коэффициента, можно добиваться не только получения решения за заданное время, но и, как обратная задача, получения решения с заданной точностью.

Пусть **вектор** $\vec{x}_0 = (x_1, x_2, \dots, x_i, \dots, x_n)$, $x_i = \{0, 1\}, \forall i = \overline{1..n}$ – одно из решений задачи, необязательно оптимальное; **нулевой вектор** – $\vec{x}_0 = (0, 0, 0, \dots, 0)$, $x_i = 0, \forall i = \overline{1..n}$; **рассматриваемый вектор** – вектор, рассматриваемый в данный момент; **ранг** ($Rang$) – число, сопровождающее каждый вектор и указывающее какой x_i необходимо изменять при следующем анализе вектора. $Rang = i$; **текущий максимум** (F_T) максимальное значение функционала, которое уже было достигнуто; **текущий оптимальный вектор** – вектор, при котором функционал равен текущему максимуму; **запомнить в стеке** – запомнить рассматриваемый вектор с его $Rang$; **извлечь из стека** – извлечь из стека вектор и присвоить его рассматриваемому, также изменив $Rang$, тогда алгоритм можно описать следующим способом:

Шаг 1. За исходный берем нулевой вектор и присваиваем его рассматриваемому вектору (далее вектору). Для него $Rang := 1$. Определяем P_G вектора и присваиваем полученное значение текущему максимуму, а текущему оптимальному вектору присваиваем вектор.

Шаг 2. Запоминаем вектор в стеке векторов.

Шаг 3. Если стек пуст, то перенести на Шаг 10.

Шаг 4. Извлекаем из стека вектор. Если $Rang = n$, перейти на Шаг 3.

Шаг 5. Присваиваем $x_{Rang} = 1$. Определяем P_G вектора. Если $P_G > F_T$, то $F_T := P_G$, а текущему оптимальному вектору присваивается вектор.

Шаг 6. Определяем P_o вектора. Уменьшаем его на величину $(P_o - P_G) \times \epsilon_{огр}$. Если полученный $P_o > F_T$, то запоминаем в стеке вектор с $Rang = Rang + 1$.

Шаг 7. Присваиваем $x_{Bit} = 0$. Определяем P_G вектора. Если $P_G > F_T$, то $F_T := P_G$, а текущему оптимальному вектору присваивается вектор.

Шаг 8. Определяем оптимистический прогноз (P_o) вектора. Уменьшаем его на величину $(P_o - P_G) * \epsilon_{огр}$. Если полученный $P_o > F_T$, то запоминаем в стеке вектор с $Rang = Rang + 1$.

Шаг 9. Перейти на Шаг 3.

Шаг 10. Полученный текущий оптимальный вектор является решением, а F_T – значение функционала при подстановке оптимального вектора.

Разработанный алгоритм тестировался на задачах линейного и квадратичного программирования. Результаты опытов для квадратичной задачи размерностью 40×15 приведены на рис. 1 и 2.

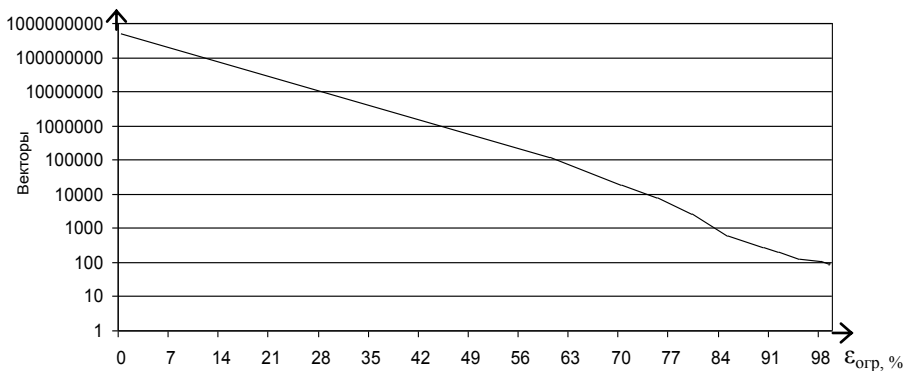


Рис. 1. Зависимость количества обрабатываемых векторов от $\epsilon_{огр}$

На основе полученных характеристик (рис. 1, 2) для заданной временной сложности, выраженной в количестве обрабатываемых векторов,

можно определить значение $\epsilon_{\text{огр}}$, позволяющее определить погрешность решения задачи.

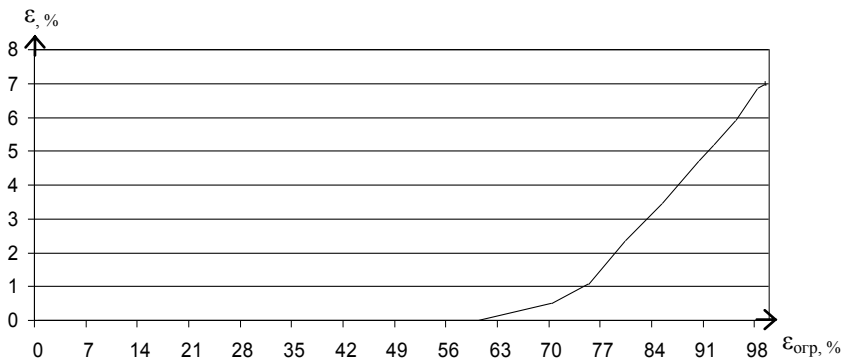


Рис. 2. Зависимость относительной погрешности решения от $\epsilon_{\text{огр}}$

Выводы и перспективы дальнейшего использования. В целом анализ результатов опытов позволяет сделать следующие выводы:

1. Как видно из результатов опытов для квадратичной задачи 40×15 при варьировании временной сложностью алгоритма от 10^9 до 10^2 обрабатываемых векторов погрешность изменяется от 0 до 7%.

2. Алгоритм возможно использовать и для задач большей степени нелинейности, чем квадратичная, при условии разработки способов нахождения верхней и нижней оценок.

ЛИТЕРАТУРА

1. Листровой С.В., Голубничий Д.Ю., Листровая Е.С. Метод решения задач целочисленного линейного программирования с булевыми переменными на основе рангового подхода // *Электронное моделирование*. – 1998. – 20, № 6. – С. 14 – 32.
2. Листровой С.В., Симашкевич О.Н. Об использовании гарантированных прогнозов в методах решения задач булевого программирования на основе рангового подхода // *Электронное моделирование*. – 2003. – 25, № 4. – С. 89 – 103.
3. Кофман А., Анри-Лабродер А. Методы и модели исследования операций. Целочисленное программирование. – М.: Мир, 1977. – 236 с.

Поступила 16.02.2004

СИМАШКЕВИЧ Олег Николаевич, адъюнкт Харьковского военного университета, который окончил в 1999 г. Область научных интересов – задачи дискретной оптимизации и исследования операций.