Использование разработанного алгоритма проверки совместности системы линейных неравенств для решения оптимизационных задач в строительстве

УДК 69.003

Мартишин Сергей Анатольевич
К.ф.-м.н., научный сотрудник Федерального государственного бюджетного учреждения науки Институт системного программирования им. В.П. Иванникова Российской академии наук, г. Москва, Россия; e-mail: smartishin@yandex.ru

Статья получена: 23.01.2023. Одобрена:02.02.2023. Опубликована онлайн: 24.03.2023.

Аннотация: Для решения многих задач в строительстве нашли применение методы оптимизации. Такие известные задачи, как транспортная задача, некоторые задачи строительной механики, задача оптимального расположения объектов на строительной площадке, назначения состава строительных бригад при производстве строительно-монтажных работ, задачи технологической комплектации и ряд других могут быть сведены к задачам линейного программирования. Приводится алгоритм проверки совместности системы из η линейных неравенств в пространстве вещественных чисел R^d. Алгоритм основан на последовательном построении гиперплоскостей в пространстве R^{d+2}. Рассматривается применение данного алгоритма для решения задачи линейного программирования.

Ключевые слова: методы оптимизации, проверка совместности системы линейных неравенств, задача линейного программирования

Актуальность работы

Существует большое число задач, которые могут быть сведены к задаче линейного программирования (ЛП), например: транспортная задача [7,9], проблема мгновенной корректировки режима электроэнергетической системы [8] и др. Наряду с этим, существует и ряд задач, в которых ограничения задаются системой линейных неравенств, но не требуется нахождения оптимума линейной целевой функции. Например, проверка, является ли точка крайней для множества точек в многомерном пространстве, сводится к построению системы линейных неравенств и проверки ее на совместность.

Областью допустимых решений называется множество всех точек, для которых выполняются все ограничения. Если допустимая область не пуста, то существует сколь угодно малое $\varepsilon > 0$, такое, что можно найти точку, нарушающую ограничения не более чем на ε.

USE OF THE DEVELOPED ALGORITHM FOR CHECKING THE COMPATIBILITY OF A SYSTEM OF LINEAR INEQUALITIES FOR SOLVING OPTIMIZATION PROBLEMS IN CONSTRUCTION

Martishin Sergei Anatolievich
Candidate of Physical and Mathematical Sciences, Senior Researcher, Federal State Budgetary Institution of Science Institute of System Programming named after V.P. Ivannikov of the Russian Academy of Sciences, Moscow, Russia; e-mail: smartishin@yandex.ru

Abstract: Optimization methods have been used to solve many problems in construction. Such well-known problems as the transport problem, some problems of structural mechanics, the problem of the optimal location of objects on the construction site, the assignment of the composition of construction teams in the production of construction and installation works, the tasks of technological equipment and a number of others can be reduced to linear programming problems. An algorithm for checking the compatibility of a system of n linear inequalities in the space of real numbers R^d is given. The algorithm is based on the sequential construction of hyperplanes in the space R^{d+2}. The application of this algorithm for solving a linear programming problem is considered.

Keywords: optimization methods, checking the compatibility of a system of linear inequalities, linear programming problem

Если при помощи алгоритма точка, нарушающая ограничения не более чем на ε, не найдена, то система признается несовместной.

Предложенный в статье алгоритм позволяет решать задачу ЛП.

Постановка задачи

Пусть в пространстве вещественных чисел R^d задана система из n линейных неравенств $Ax \leq b$. Точку, удовлетворяющую системе линейных неравенств, будем называть допустимой. Множество всех допустимых точек назовем допустимой областью. Если допустимая область не пуста, то система линейных неравенств совместна, в противном случае несовместна.

Требуется определить, является ли система совместной. Если система совместна, то требуется найти точку, которая является допустимой.

Таким образом, на входе имеется матрица ограничений системы размера $n \cdot (d+1)$, где n и d фиксированы. Заметим, что на практике матрица, возникающая из реальных задач, является разреженной. Поэтому в алгоритме предусмотрено компактное хранение матрицы (только ненулевые элементы).

Рассмотренный ниже алгоритм ищет допустимую точку в R^{d+2}. Дополнительные построения осуществляются таким образом, чтобы исходная допустимая область и в R^{d+2} также являлась допустимой областью.

Для этого необходимо преобразовать множество линейных неравенств (обозначим его через H^d) в пространстве R^d в множество линейных неравенств (обозначим его через H^{d+2}) в пространстве R^{d+2}. Множество линейных уравнений (гиперплоскостей), полученных из неравенств заменой знака «меньше или равно» на знак «равно» обозначим через $H_{=}^{d}$ и $H_{=}^{d+2}$ соответственно.

Дополнительные построения

Проведем дополнительные построения.

Алгоритм

Шаг 1. В пространстве R^{d+1} выберем точку O_1 с координатами $(O,\ldots O,C_1)$, - любое, такое, что C_1. Неравенства из множества H^d преобразуем (добавив в каждое неравенство коэффициент a_{id+1} при x_{d+1}) так, чтобы в пространстве R^{d+1} соответствующие им гиперплоскости проходили через точку O_1.

$$a_{id+1} = \frac{b_i}{C_1} \tag{1}$$

Множества построенных неравенств и гиперплоскостей обозначим через H^{d+1} и $H_{=}^{d+1}$. Заметим, что любая точка в пространстве R^d лежит в полупространстве того же знака для ограничения из H^d и соответствующей гиперплоскости из H^{d+1}.

Шаг 2. В пространстве размерности R^{d+2} выберем точку O_2 с координатами $(O,\ldots O,C_1,C_2),C_2,$ - любое, такое, что $C_2 > 0$. Построим сферу с центром в точке O_2 и радиусом r, $r < C_2$.

Неравенства из множества H^{d+1} преобразуем (добавив в каждое неравенство коэффициент a_{id+2} при x_{d+2}) так, чтобы в пространстве R^{d+2} соответствующие им гиперплоскости касались сферы с центром в точке O_2, и точка O_2 удовлетворяла каждому построенному в R^{d+2} неравенству, то есть находилась в его отрицательном полупространстве соответствующей гиперплоскости. Коэффициенты находятся путем решения квадратного уравнения. Выбор константы C_2 и радиуса r обеспечивает существование двух вещественных корней квадратного уравнения. Квадратное уравнение получаем из формулы расстояния от гиперплоскости в R^{d+2} до точки O_2.

$$\frac{a_{id+1} \cdot C_1 + a_{id+2} \cdot C_2 - b_i}{\sqrt{\sum_{j=1}^{d+1} a_{ij}^2 + a_{id+2}^2}} = -r \tag{2}$$

Далее выбирается тот корень, при котором точка O_2 будет находиться в отрицательном полупространстве гиперплоскости соответствующего ограничения. В результате получим множество неравенств и гиперплоскостей H^{d+2} и $H_{=}^{d+2}$. Как и на предыдущем шаге, любая точка в пространстве R^d лежит в полупространстве того же знака для гиперплоскости из $H_{=}^{d}$ и соответствующей гиперплоскости из $H_{=}^{d+2}$.

На практике, компьютерные вычисления используют рациональную арифметику. Известно, что в рациональной арифметике элементарные арифметические операции (сложение, вычитание, умножение и деление) могут быть выполнены полиномиальными алгоритмами. Вычисление a_{id+1} выполняется при помощи элементарной операции деления, следовательно, имеет полиномиальную временную сложность.

При вычислении a_{id+2} помимо элементарных операций используется операция извлечения корня. Однако заметим, что для вычисления квадратного корня существует алгоритм, число шагов которого не превышает полиномов от длин записей используемых чисел [10]. Следовательно, вычисление a_{id+2} имеет полиномиальную временную сложность.

Построение первоначальной $H_{=}^{cur}$

Построим отрезок $[O_1, O_2]$, где координаты точки О0 в R^{d+2} равны (О,...0), а точки O_1 равны $(O, \dots O, C_1, 0)$. Так как вектор нормали имеет вид, $\vec{n}\{b_1, \dots, b_{d+2}\}$ где $b_1, \dots, b_{d+2}$ коэффициенты соответствующей гиперплоскости для этой нормали, то легко найти саму гиперплоскость. Обозначим через $H_{=}^{cur}$ полученную гиперплоскость, проходящую через точку O_1. Заметим, что по построению гиперплоскость $H_{=}^{cur}$проходит и через точку O_2. Коэффициенты $H_{=}^{cur}$ имеют такой знак, что точка O_0 лежит в отрицательном полупространстве.

Гиперплоскость, параллельную $H_{=}^{cur}$ и проходящую через точку O_0 обозначим через $H_{=}^{sol}$.

Для произвольного отрезка, лежащего в R^{d+1}, и рассматриваемого как нормаль к некой $H_{=}^{cur}$, для x_{d+2} гиперплоскости коэффициент при будет нулевым.

Алгоритм проверки совместности системы неравенств

В процессе работы алгоритма происходит перестроение $H_{=}^{cur}$. Для перестроения используются точки касания ограничений к сфере P_i^{cont} в пространстве R^{d+2}.

Так как для хранения всех точек касания P_i^{cont} $(i \in [1,n])$ требуется память размера $n \cdot d$, то предлагается вычислять координаты точек касания динамически. Точка касания ограничения к сфере вычисляется как пересечение гиперплоскости со своей нормалью, проходящей через точку O_2, что требует использование алгоритма пересечения прямой с гиперплоскостью, сложность которого $O(d+2)$.

Алгоритм

Шаг 1. Строится первоначальная $H_{=}^{cur}$.

Шаг 2. Вычисляется центр масс P_{cm} для точек P_i^{cont}, которые оказались в отрицательном полупространстве $H_{=}^{cur}$.

Шаг 3. На прямой (O_1, O_2) выбирается точка O_{new} с $d + 2$ координатой равной $d + 2$ координате точки P_{cm}. Строится прямая lnew, проходящая через точки O_{new} и P_{cm}.

Шаг 4. Из точки P_{cm} опускается перпендикуляр на $H_{=}^{cur}$. Основание перпендикуляра будет точкой $P_{cur}^{\perp}$. На прямой $(O_1, P_{cur}^{\perp})$ выбирается точка P_{cur} с $d + 2$ с координатой равной $d + 2$ координате точки P_{cm}.

Шаг 5. Строится гиперплоскость, перпендикулярная прямой l_{new} и проходящая через точку P_{cur}. Точкой пересечения этой гиперплоскости с прямой l_{new} будет точка P_{new}.

Шаг 6. Строится прямая $(P_{cur} P_{new})$. Параллельная ей прямая, проходящая через точку O_1 будет прямой l. Точка P_{sol} будет пересечением прямой и гиперплоскости $H_{=}^{sol}$. Перпендикулярная прямой l гиперплоскость, проходящая через точку O_1, станет гиперплоскостью $H_{=}^{cur}$.

Шаг 7. Если точка P_{sol} удовлетворяет невязке в исходном пространстве P^d, то решение найдено. Если условия останова не выполнены, то осуществляется переход на Шаг 2.

Теоретические аспекты

Теорема. Если существует такая гиперплоскость $H_{=}^{cur}$, что все точки P_i^{cont} $(i \in [1,n])$ лежат в ее положительном полупространстве, а точка O_0 в ее отрицательном полупространстве, то существует точка в допустимой области неравенств H^d в R^d.

Доказательство. Пусть P_i^{cont} – точка касания, лежащая в положительном полупространстве $H_{=}^{cur}$. По построению существует полуинтервал $[P_i^{cont}, O_2)$ лежащий в положительном полупространстве для $H_{=}^{cur}$.

Восстановим перпендикуляр к гиперплоскости $H_{=}^{cur}$ в точке O_2 и найдем его точку пересечения P_{is} с гиперплоскостью i, которая содержит P_i^{cont}.

Таким образом, существует отрезок, один конец которого лежит в отрицательном полупространстве для гиперплоскости i, а другой принадлежит гиперплоскости i.

Прямая l, параллельная этому отрезку и проходящая через точку O_1 даст точку P_{sol} при пересечении с гиперплоскостью i. Точка будет лежать в отрицательном полупространстве для i гиперплоскости. Это верно и для любого другого i.

В случае, если не существует гиперплоскости $H_{=}^{cur}$ удовлетворяющей условию теоремы, то P_{sol} не будет лежать в отрицательном полупространстве для некой гиперплоскости i. То есть существует P_i^{cont} лежащая в отрицательном полупространстве $H_{=}^{cur}$.

Условия останова

Перечислим условия останова алгоритма:

- построенная гиперплоскость $H_{=}^{cur}$ содержит точку O_0 в положительном полупространстве;
- за установленное число перестроений $H_{=}^{cur}$ не произошло статистического улучшения невязки для точки P_{sol};
- число перестроений $H_{=}^{cur}$ превысило заданное допустимое количество.

Решение задачи линейного программирования (ЛП)

Если рассматривать целевую функцию как ограничение, то алгоритм можно использовать для решения задачи ЛП.

При проверке совместности системы линейных неравенств находится точка P_{sol} . Точка P_{sol} будет наихудшим значением для целевой функции. Можно задать наилучшее желаемое значение целевой функции, то есть построить ограничение из целевой функции с соответствующим свободным членом. Ограничение для целевой функции можно перестраивать, например, при помощи метода дихотомии.

Заключение

Таким образом, динамически генерируя ограничение из целевой функции, можно решить задачу ЛП при помощи проверки совместности системы линейных неравенств.

Алгоритм обеспечивает экономное расходование памяти для разреженной матрицы ограничений. В процессе работы алгоритма точки касания к сфере строятся динамически.

При работе алгоритма не возникает ситуация зацикливания.

Алгоритм хорошо распараллеливается, что позволяет эффективно задействовать все ядра процессора при работе с общей памятью.

Алгоритм не использует сторонние библиотеки, например, для решения системы линейных уравнений.

Входные данные алгоритма не привязаны к устаревшему формату MPS (формат IBM) и могут быть представлены в виде, удобном для хранения разреженной матрицы.

Литература

1. Luenberger D.G. Yinyu Y. Linear and Nonlinear Programming (International Series in Operations Research & Management Science, 228) 4th ed. Springer, 2016.
2. Pellegrini M. Randomizing combinatorial algorithms for linear programming when the dimension is moderately high. Symposium on Discrete Algorithms. Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms. Washington, D.C., United States, P. 101-108, 2001.
3. Roos C, Terlaky T., Vial J.P. Interior point methods for linear optimization (2-nd edition). Boston: Birkhauser, 2006.
4. Topaloglu H. Fundamentals of Linear Optimization: A Hopefully Uplifting Treatment. School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044, 2021.
5. Tsuchiya T. and Muramatsu M. Global convergence of long-step affine scaling algorithm for degenerate linear programming problems. SIAM Journal on Optimization. Vol.5, No.3, pp.525-551, 1995.
6. Александров П.С. Лекции по аналитической геометрии, пополненные необходимыми сведениями из алгебры с приложением собрания задач, снабженных решениями, составленного А.С. Пархоменко: Учебник. 2-е изд., стер.- СПб.: Издательство «Лань», 2008.
7. Васильев Ф.П. Иваницкий А.Ю. Линейное программирование. Изд. 3-е испр. М.: Факториал Пресс, 2008.
8. Дикин И.И. Метод внутренних точек в линейном и нелинейном программировании. М.: КРАСАНД, 2010.
9. Лунгу К.Н. Линейное программирование. Руководство к решению задач. 2-е изд., испр. и доп. М.: ФИЗМАТЛИТ, 2009.
10. Схрейвер А. Теория линейного и целочисленного программирования. В 2-х т. Т.I. М.: Мир, 1991.
11. Препарата Ф., Шеймос М., Вычислительная геометрия: Введение. М.: Мир, 1989.