

Вячеслав Безбородов

Распараллеливание симплекс-метода на практике

11 октября 2016 г.

Аннотация

В большинстве случаев симплекс-метод является самым эффективным методом решения задач линейного программирования (ЗЛП). В работе рассматриваются существующие попытки распараллеливания симплекс-метода по отношению к эффективной последовательной реализации и характеру практических ЗЛП. Для решения разреженных ЗЛП большой размерности не существует параллельной реализации симплекс-метода, значительно превосходящей по производительности хорошую последовательную реализацию. Хотя существует некоторый прогресс в разработке параллельных реализаций для неразреженных ЗЛП или ЗЛП, не имеющих особых структурных свойств. Как результат такого обзора, эта работа определяет направления будущих исследований в области разработки параллельных реализаций симплекс-метода, имеющих практическое значение.

Ключевые слова: линейное программирование, симплекс-метод, разреженная матрица, параллельные вычисления.

1 Введение

Задачи линейного программирования (ЗЛП) возникают из различных областей науки, в т.ч. и как промежуточные этапы при решении других оптимизационных задач. Симплекс-метод и методы внутренней точки являются двумя главными подходами для решения ЗЛП. В случаях, когда решаются семейства взаимосвязанных ЗЛП (целочисленное программирование, методы разложения, некоторые классы задач линейного программирования), симплекс-метод обычно более эффективен.

Механизмы применения параллельной и векторной обработки к симплекс-методу для решения ЗЛП стали обсуждаться с 1970-х гг., хотя первые попытки разработать практические реализации были предприняты только с начала 1980-х гг. Наибольшая активность в этом направлении наблюдалась с середины

1980-х до середины 1990-х гг. Также предпринимались эксперименты использования векторной обработки данных и ЭВМ с общей разделяемой памятью, подавляющее большинство реализаций использовали мультипроцессоры с распределенной памятью и сетевые кластеры.

2 Симплекс-метод

Симплекс-метод и его требования в вычислительному процессу наиболее удобно обсуждать в контексте ЗЛП в стандартной форме

$$\begin{aligned} c^T x &\rightarrow \min \\ Ax &= b \\ x &\geq 0, \end{aligned} \tag{2.1}$$

где $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$. Матрица A в (2.1) обычно содержит столбцы с единицами, соответствующими фиктивным переменным, возникающим при переводе ограничений-неравенств в равенства. Оставшиеся столбцы A соответствуют обычным переменным.

В симплекс-методе индексы переменных подразделяются на два подмножества: \mathcal{B} , соответствующее m базисным переменным x_B , и \mathcal{N} , соответствующее $n - m$ небазисным переменным x_N . При этом базисная матрица B , составленная из столбцов A , соответствующих \mathcal{B} , является невырожденной. Множество \mathcal{B} условно называют базисом. Столбцы A , соответствующие \mathcal{N} , формируют матрицу N . Компоненты c , соответствующие \mathcal{B} и \mathcal{N} , называют базисными издержками c_B и небазисными издержками c_N .

Когда небазисные переменные нулевые, значения $\hat{b} = B^{-1}b$ базисных переменных соответствуют вершинам допустимого региона при условии, что они неотрицательны. Выражение $x_B + B^{-1}N = \hat{b}$, следующее из (2.1), позволяет убрать базисные переменные из целевой функции, которая становится $(c_N^T - c_B^T B^{-1}N)x_N + c_B^T \hat{b}$. Если все компоненты вектора альтернативных издержек $\hat{c}_N = c_N^T - c_B^T B^{-1}N$ неотрицательны, то текущий базис оптимален.

На каждой итерация симплекс-метода, если текущий базис неоптимален, выбирается небазисная переменная x_q с отрицательной альтернативной издержкой для ввода в базис. Увеличение этой переменной от нуля при выполнении условий (2.1) соответствует перемещению вдоль ребра допустимого региона в направлении уменьшения значения целевой функции. Направление этого ребра определяется столбцом \hat{a}_q при $\hat{N} = B^{-1}N$, соответствующим x_q . При просмотре отношений компонентов вектора \hat{b} к соответствующим положительным компонентам \hat{a}_q находится первая базисная переменная для обнуления при росте x_q и, следовательно, шаг к следующей точке допустимого региона вдоль этого ребра.

Существует много стратегий выбора переменной x_q для ввода в базис. Первоначальное правило выбора переменной с наименьшей альтернативной из-

держкой известно как критерий Данцига. Хотя, если компоненты \hat{a}_j намного превосходят компоненты \hat{c}_j , только небольшое увеличение x_j возможно до того, как одна из базисных переменных обратится в ноль. Другие стратегии выбора взвешивают альтернативную издержку путем деления на длину \hat{a}_j . Точная стратегия выбора ребра [TODO: REF] оперирует весами $s_j = 1 + \|\hat{a}_j\|^2$, соответствующими длине шага при единичном увеличении x_j . Приближенные (применяемые на практике) техники [TODO: REF] и [TODO: REF] оперируют приближенными весами ребер. При использовании этих стратегий количество итераций, необходимых для решения ЗЛП, на практике может быть оценено как $O(m+n)$, и не выявлены проблемы, мешающие достижению теоретической сложности $O(2^n)$.

Одной из широко распространенных способов выбора выводимой из базиса переменной является процедура *EXPAND* [TODO: REF]. С помощью небольшого расширения ограничений эта стратегия позволяет выбрать выводимую переменную из набора возможных на основании численной стабильности.

Существуют две основных варианта симплекс-метода, использующие различные данные для перехода к следующей вершине. В стандартном симплекс-методе альтернативные издержки и направления всех ребер в текущей точке хранятся в прямоугольной таблице. В методе обратной матрицы альтернативные издержки и направления выбранного ребра определяются решением системы с участием базисной матрицы B .

2.1 Табличный симплекс-метод

В табличном симплекс-методе матрица \hat{N} вектор \hat{b} , альтернативные издержки \hat{c}_N и текущее значение целевой функции $\hat{f} = c_B^T \hat{b}$ хранятся в таблице, общий вид которой показан в таблице 1.

	\mathcal{N}	RHS
\mathcal{B}	\hat{N}	\hat{b}
	\hat{c}_N^T	$-\hat{f}$

Таблица 1 – Структура хранения данных табличного симплекс-метода.

Каждая итерация метода требует применения преобразований Жордана-Гаусса к столбцам таблицы таким образом, что новая таблица соответствует новому базису.

Обычно симплекс-метод начинается с базиса, для которого $B = E$, следовательно матрица соответствует N . Как результат, таблица является разреженной. Как правило полагается, что существует достаточная степень заполненности матрицы после выполняемых преобразований. Как результат, табличный симплекс-метод реализуется с использованием плотных структур данных.

Табличный симплекс-метод по существу является численно нестабильным, поскольку необходимо выполнять длительную последовательность исключений с операциями поворота, определенными алгоритмом метода, а не соображениями численной эффективности. Если симплекс-метод имеет дело с плохо обусловленными базисными матрицами, то ожидается, что любая подпоследовательность таблицы 1 будет соответствовать хорошо обусловленному базису для получения ошибок вычисления, отражающих первоначальную плохую обусловленность. Это может привести к выборам вводимых или выводимых переменных таких, что при точных вычислениях, целевая функция уменьшается монотонно, сходимость теряется либо базисная матрица становится вырожденной. Надежность может быть достигнута только при отслеживании ошибок в таблице и, при необходимости, пересчете таблицы численно стабильным способом. Обнаружение ошибок можно осуществить сравнением нового значения альтернативной издержки с значением, вычисленным напрямую с использованием столбца поворота и базисных издержек. С другой стороны, поскольку операции с обратной базисной матрицей могут быть выполнены с использованием подходящих элементов таблицы, вычисление столбца поворота может предоставить более расширенный, но с вычислительной точки зрения более дорогой механизм отслеживания ошибок.

2.2 Метод обратной матрицы

Алгоритм метода обратной матрицы представлен ниже.

Метод эллипсоидов *Шаг 0. Инициализация.*

Положить $x_k = x_0$; $B_k = E$, где E – единичная матрица размерности $n \times n$; $h_k = \frac{R}{n+1}$ – коэффициент, отвечающий за уменьшение объема шара. Перейти к шагу 1.

Шаг 1. Вычислить

$$g(x_k) = \begin{cases} g_0(x_k), & \text{если } \max_{1 \leq i \leq m} f_i(x_k) \leq 0, \\ g_{i^*}(x_k), & \text{если } \max_{1 \leq i \leq m} f_i(x_k) = f_{i^*}(x_k) > 0. \end{cases}$$

Если $g(x_k) = 0$, то завершить алгоритм; x_k – оптимальная точка. Иначе перейти к шагу 2.

Шаг 2. Вычислить $\xi_k = \frac{B_k^T g(x_k)}{\|B_k^T g(x_k)\|}$. Перейти к шагу 3.

Шаг 3. Вычислить $x_{k+1} = x_k - h_k \cdot B_k \cdot \xi_k$. Перейти к шагу 4.

Шаг 4. Вычислить $B_{k+1} = B_k \cdot R_\beta(\xi_k)$, где $R_\beta(\xi_k)$ – оператор растяжения пространства в направлении ξ_k с коэффициентом β (см. определение ??), $\beta = \sqrt{\frac{n-1}{n+1}}$. Перейти к шагу 5.

Шаг 5. Вычислить $h_{k+1} = h_k \cdot r$, где $r = \frac{n}{\sqrt{n^2-1}}$. Перейти к шагу 1.