

Параллельные методы решения задач линейной оптимизации

В.А. Гаранжа, А. И. Голиков, Ю. Г. Евтушенко, И.Е. Капорин
Вычислительный центр им. А.А.Дородницына РАН, evt@ccas.ru

Традиционные методы линейного программирования (ЛП) в ряде случаев теряют свою эффективность при решении задач большой размерности (задачи с десятками миллионов неизвестных и с сотнями тысяч ограничений). Возникает необходимость создания новых численных методов, которые в полной мере используют возможности современных многопроцессорных вычислительных комплексов. Предложены новые алгоритмы для многопроцессорных вычислительных систем с распределенной памятью, их эффективность подтверждена вычислительными экспериментами и решением ряда практических задач [1], [2].

Пусть прямая и двойственная задачи ЛП заданы в стандартной форме

$$f_* = \min_{x \in X} c^\top x, \quad X = \{x \in R^n : Ax = b, x \geq 0_n\}, \quad (P)$$

$$f_* = \max_{u \in U} b^\top u, \quad U = \{u \in R^m : A^\top u \leq c\}. \quad (D)$$

Здесь $A \in R^{m \times n}$, $c \in R^n$ и $b \in R^m$ заданы, x – вектор прямых переменных, а u – двойственных, через 0_i обозначен i -мерный нулевой вектор. Всюду ниже предполагаем, что множество решений X^* прямой задачи (P) непусто, тогда множество решений U^* двойственной задачи (D) также непусто.

Для нахождения проекции заданной точки \hat{x} на множество решений прямой задачи (P) введем вспомогательную задачу безусловной максимизации

$$\max_{p \in R^m} S(p, \beta, \hat{x}), \quad \text{где } S(p, \beta, \hat{x}) = \{b^\top p - \frac{1}{2} \|(\hat{x} + A^\top p - \beta c)_+\|^2\}. \quad (1)$$

Здесь скаляр β фиксирован, a_+ – вектор, у которого i -я компонента совпадает с i -й компонентой вектора a , если она неотрицательна, и равна нулю в противном случае.

Воспользуемся следующими ранее сформулированными в [1] теоремами.

Теорема 1 Существует такое число β_* , что при любом $\beta \geq \beta_*$ пара $[p(\beta), \beta]$, где $p(\beta)$ – решение задачи безусловной максимизации (1), определяет проекцию \hat{x}^* точки \hat{x} на множество решений X^* прямой задачи (P) по формуле

$$\hat{x}^* = (\hat{x} + A^\top p(\beta) - \beta c)_+. \quad (2)$$

Теорема 2 При $\hat{x} = x^* \in X^*$ и при любом $\beta > 0$ точное решение двойственной задачи (D) находится по формуле $u^* = p(\beta)/\beta$, где $p(\beta)$ – решение задачи безусловной максимизации (1).

Итак, в результате решения двух задач безусловной максимизации вогнутой кусочно-квадратичной функции от m переменных получается проекция заданной точки \hat{x} на множество решений X^* прямой задачи (P) и некоторое решение двойственной задачи (D).

Следующий итерационный процесс дает одновременное решение прямой и двойственной задач ЛП

$$p_{s+1} \in \arg \max_{p \in R^m} \{b^\top p - \frac{1}{2} \|(x_s + A^\top p - \beta c)_+\|^2\} \quad (3)$$

$$x_{s+1} = (x_s + A^\top p_{s+1} - \beta c)_+. \quad (4)$$

Здесь произвольный параметр $\beta > 0$ фиксирован.

Теорема 3 При любом $\beta > 0$ и для любой начальной точки x_0 итерационный процесс (3), (4) сходится к $x^* \in X^*$ за конечное число шагов ω . Формула $u^* = p_{\omega+1}/\beta$ определяет точное решение двойственной задачи (D).

Решение задач безусловной максимизации (1) или (3) может выполняться любым методом, например, методом сопряженного градиента. Однако гораздо эффективней использовать обобщенный метод Ньютона. Доказательство конечной глобальной сходимости обобщенного метода Ньютона для безусловной оптимизации кусочно-квадратичной функции с выбором шага по правилу Армихо предложено О. Мангасарьяном. Рассмотрим параллельные реализации метода (3), (4) в вычислительной системе с распределенной памятью. Считаем, что у каждого процесса (исполняемой программы) имеется свое адресное пространство, а обмены данными между процессами осуществляются при помощи библиотеки MPI, каждый процесс выполняется на своем процессоре (ядре).

Расчетные формулы.

1. Задать $\beta > 0$, пороги точности $tol1$ и tol для внешних и внутренних итераций, соответственно, начальные приближения x_0 и p_0 .

2. Вычислить значение функции $S(p_k, \beta, x_s)$ и ее градиент:

$$G_k = \frac{\partial S}{\partial p}(p_k, \beta, x_s) = b - A(x_s + A^T p_k - \beta c)_+.$$

Здесь k - номер внутренней итерации метода Ньютона для решения задачи безусловной максимизации (3), а s - номер внешней итерации.

3. Используя обобщенную матрицу Гессе функции $S(p_k, \beta, x_s)$, сформировать матрицу $H_k \in R^{m \times m}$:

$$H_k = \delta I + A D_k A^T, \quad (5)$$

где δ - некоторое положительное число (обычно 10^{-4}), I - единичная матрица, диагональная матрица $D_k \in R^{n \times n}$ задается равенствами

$$(D_k)_{ii} = \begin{cases} 1 & \text{если } (x_s + A^T p_k - \beta c)^i > 0 \\ 0 & \text{если } (x_s + A^T p_k - \beta c)^i \leq 0 \end{cases}$$

4. Найти направление максимизации Δp из решения линейной системы

$$H_k \Delta p = -G_k \quad (6)$$

с помощью предобусловленного метода сопряженных градиентов. В качестве предобусловливателя используется диагональная часть матрицы H_k .

5. Определить p_{k+1} по формуле $p_{k+1} = p_k - \tau_k \Delta p$, где итерационный параметр τ_k находится из решения одномерной задачи максимизации $\tau_k = \max_{\tau} S(p_k - \tau \Delta p, \beta, x_s)$ методом Армихо.

6. Если выполнен критерий останова для внутренних итераций $\|p_{k+1} - p_k\| \leq tol$, то положить $\tilde{p} = p_{k+1}$ и вычислить x_{s+1} по формуле

$$x_{s+1} = (x_s + A^T \tilde{p} - \beta c)_+.$$

Иначе перейти к шагу 2), положив $k = k + 1$.

7. Если выполнен критерий останова для внешних итераций $\|x_{s+1} - x_s\| \leq tol1$, то вычислить решение двойственной задачи (D) $u^* = \frac{\tilde{p}}{\beta}$ и решение прямой задачи (P) есть $x^* = x_{s+1}$. Иначе положить $p_0 = \tilde{p}$ и перейти к шагу 2), положив $s = s + 1$.

В приведенном алгоритме наиболее трудоемкими операциями являются формирование системы (6) и ее решение. При реализации алгоритма распараллеливаются следующие операции:

- 1) матрично-векторные умножения: Ax и $A^T p$;
- 2) вычисление скалярных произведений вида $x^T y$, $x, y \in R^n$ и $p^T q$, $p, q \in R^m$;
- 3) формирование обобщенной матрицы Гессе и матрицы H_k (5);
- 4) умножение матрицы H_k на вектор;

Все остальные вычисления являются локальными. Было реализовано несколько параллельных схем приведенного алгоритма в зависимости от вида разбиения исходной матрицы A на блоки: клеточная схема, когда матрица A разбивается на одинаковые блоки, количество которых равно числу процессоров; столбцовая схема – матрица A разбивается на блоки по столбцам; строчная схема – матрица A разбивается на блоки по строкам. Каждая из предложенных схем имеет свои достоинства и недостатки, чем и определяется область их применения. Так столбцовая схема весьма эффективна при формировании системы линейных уравнений (6), строчная схема более эффективна для параллельного метода решения системы линейных уравнений. Была еще реализована "безматричной" схема, которая позволила решать задачи ЛП с наибольшим числом ограничений m по сравнению с другими схемами. Эта схема наиболее рационально использует память кластера, поэтому ее целесообразно применять при больших значениях m и n . Ускорение для этой схемы при сравнительно малых m может оказаться меньше единицы.

Для численных экспериментов использовался генератор случайных тестовых задач ЛП [1]. Расчеты проводились на параллельном вычислительном кластере МВС-6000IM, состоящем из двухпроцессорных узлов на основе Intel Itanium 2 с частотой 1.6 GHz, соединенными сетью Meginet 2000 [2]. При решении задач ЛП с одним миллионом неизвестных и при десяти тысячах ограничений с помощью клеточной схемы на 144 процессорах было достигнуто ускорение расчетов примерно в 50 раз, время счета составило 28 сек. Задача ЛП с двумя миллионами переменных при двухстах тысячах ограничений на 80 процессорах была решена с помощью "безматричной" схемы менее, чем за 40 минут. С помощью столбцовой схемы разбиения задача ЛП с максимальным количеством переменных – 60 млн. при четырех тысячах ограничений была решена на 128 процессорах за 140 сек.

Список литературы

1. Голиков А.И., Евтушенко Ю.Г. Метод решения задач линейного программирования большой размерности. // Докл. Академии наук, 2004. Т. 397. № 6. С. 727-732.

2. Гаранжа В.А., Голиков А.И., Евтушенко Ю.Г. Нгуен М.Х. Параллельная реализация метода Ньютона для решения больших задач линейного программирования // Ж. вычислит. матем. и математ. физ. 2009. Т. 49. № 8. С. 1369-1384.