Параллельные методы решения экстремальных задач [1]

Голиков А. И.[1], **Евтушенко Ю. Г.**[1], **Посыпкин М. А.**[2]

[1] Вычислительный центр им. А.А.Дородницына РАН, [2] Институт системного анализа РАН

Рассмотрены метод неравномерных покрытий и обобщенный метод Ньютона для решения различных задач оптимизации. Показана возможность эффективно решать задачи большой размерности на многопроцессорных вычислительных комплексах.

Parallel Methods for Solving Optimization Problems

Golikov A. I.[1], Evtushenko Y. G.[1], Posypkin M. A.[2]

[1] Dorodnicyn Computing Centre of RAS, [2] Insistute for Systems Analysis RAS, Moscow, Russia

Non-uniform coverage method and generalized Newton method for solving various optimization problems are considered. Parallel implementation of these methods for solving large scale-scale problems is discussed.

1 Введение

Понятие большой размерности задачи оптимизации зависит от вида решаемой задачи и характеристик применяемой вычислительной техники. Характерное число переменных в задачах глобальной оптимизации измеряется десятками-сотнями. В задачах линейного программирования число переменных может достигать десятков миллионов. Рассматриваются методы, хорошо поддающиеся распараллеливанию: метод неравномерных покрытий и обобщенный метод Ньютона.

2 Теоретические основы метода неравномерных покрытий

Для непрерывной функции $f : \mathbb{R}^n \to \mathbb{R}$ ставится задача отыскания глобального минимума на *допустимом компактном множестве* $X \subseteq \mathbb{R}^n$. Формально задача записывается следующим образом:

[1]Работа выполнена при финансовой поддержке РФФИ, проект № 09-01-12098-офи_м, поддержке ведущих научных школ НШ-4096.2010.1 и программе Президиума РАН П-14.

$$f_* = glob \min_{x \in X} f(x) = f(x_*), \tag{1}$$

где x_* — точка, в которой достигается глобальный минимум, равный f_*.

Определим множество *решений* X_* этой задачи и множество ϵ-*оптимальных решений* X_*^ϵ:

$$\begin{array}{l} X_* = \{x \in X : f(x) = f_*\}, \\ X_*^\epsilon = \{x \in X : f(x) \leq f_* + \epsilon\}, \epsilon \in \mathbb{R}_+. \end{array} \tag{2}$$

Пусть задан набор из i допустимых точек $N_i = \{x_1, \ldots, x_i\}$. Определим текущий *рекорд* u_i и соответствующую *рекордную точку* x_r по формулам:

$$u_i = \min_{1 \leq j \leq i} f(x_j) = f(x_r), \tag{3}$$

Индекс r принимает значения в интервале $1 \leq r \leq i$ и, в общем случае определяется неоднозначно. Из (3) следует, что последовательность текущих рекордов такова, что

$$u_1 \geq \cdots \geq u_i \geq f_*. \tag{4}$$

Для функции $f(x) : \mathbb{R}^n \to \mathbb{R}$, множества $Z \subseteq \mathbb{R}^n$ и числа $\lambda \in \mathbb{R}$ определим понятие Лебеговского множества $S(Z, f(x), \lambda) = \{x \in Z : f(x) \geq \lambda\}$.

С каждой точкой x_i свяжем некоторую ее окрестность X_i, таким образом $x_i \in X_i$. На каждом множестве X_i определим миноранту $\mu_i(x)$, такую что $f(x) \geq \mu_i(x)$ для всех $x \in X_i$. Пусть множество S_i удовлетворяет условию

$$S_i \subseteq S(X_i, \mu_i(x), u_i - \epsilon). \tag{5}$$

Будем говорить, что совокупность множеств $\{S_i\}$, $1 \leq i \leq k$ *образует покрытие множества* X, если

$$X \subseteq \cup_{i=1}^k S_i. \tag{6}$$

Теорема 1. Пусть набор допустимых точек N_k удовлетворяет условию (6), тогда для величины глобального минимума функции $f(x)$ на множестве X имеет место оценка

$$u_k \geq f_* \geq u_k - \epsilon \tag{7}$$

и любая точка $x_r \in N_k$, такая что $f(x_r) = u_k$ является ϵ-оптимальным решением задачи (1).

Теорема 1 является теоретической базой для реализации различных вычислительных схем метода неравномерных покрытий [1, 2], в основе которых лежит построение последовательностей точек $\{x_i\}$ и соответствующих лебеговских множеств таким образом, чтобы обеспечить выполнение условия (6), называемого *условием покрытия множества* X. При этом множество X может иметь произвольную мощность и структуру.

Задача многокритериальной минимизации записывается в виде:

$$\min_{x \in X} F(x). \tag{8}$$

Здесь вектор-функция $F(x) : \mathbb{R}^n \to \mathbb{R}^m$ определяет векторный критерий, его компоненты $f^1(x), \ldots, f^m(x)$ составляют набор из m скалярных непрерывных критериев. Образ $\Omega = F(X)$ допустимого множества X при отображении F назовем *множеством достижимых критериев*.

Для произвольной точки $\theta \in \mathbb{R}^m$ определим *юго-западное* $\mathrm{SW}(\theta)$ и *северо-восточное* $\mathrm{NE}(\theta)$ множества следующим образом:

$$\mathrm{SW}(\theta) = \{y \in \mathbb{R}^m : y \leq \theta\}, \mathrm{NE}(\theta) = \{y \in \mathbb{R}^m : y \geq \theta\}.$$

Расширим эти определения на случай произвольного множества $\Theta \subseteq \mathbb{R}^m$:

$$\mathrm{SW}(\Theta) = \cup_{\theta \in \Theta} \mathrm{SW}(\theta), \;\; \mathrm{NE}(\Theta) = \cup_{\theta \in \Theta} \mathrm{NE}(\theta).$$

Для множества $\Theta \subseteq \mathbb{R}^m$ определим его *Парето-оптимальное подмножество* $\mathcal{P}(\Theta)$ следующим образом:

$$\mathcal{P}(\Theta) = \{\theta \in \Theta : \Theta \cap \mathrm{SW}(\theta) = \theta\}.$$

Результат $\Omega_* = \mathcal{P}(\Omega)$ применения введенного оператора $\mathcal{P}$ к множеству достижимых критериев Ω назовем *множеством Парето* для задачи 8. Пусть $A_* \subseteq X$ — прообраз множества Ω_* при отображении F, т.е. $\Omega_* = F(A_*)$. Решение задачи (8) состоит в нахождении множеств A_* и Ω_*.

Пусть задан набор допустимых точек $N_i = \{x_1, \ldots, x_i\} \subseteq X$. Ему соответствует множество образов $F(N_i) = \{F(x_1), \ldots, F(x_i)\}$. Из этого подмножества выделим парето-оптимальное подмножество $\Omega_i = \mathcal{P}(F(N_i))$. Оператор $\mathcal{P}$ легко алгоритмически реализуется.

Будем говорить, что множество Ω_*^ϵ является — ϵ-Парето множеством, если для каждого $\omega \in \Omega_*$ найдется точка $\omega' \in \Omega_*^\epsilon$, такая что $\omega \in \mathrm{NE}(\omega' - \epsilon)$. Также потребуем, чтобы в множестве Ω_*^ϵ не было двух таких различных точек ω_1, ω_2, что $\omega_1 \leq \omega_2$. Пусть A_*^ϵ — прообраз множества Ω_*^ϵ при отображении $F(x)$, т.е. $\Omega_*^\epsilon = F(A_*^\epsilon)$. Приближенное решение задачи (8) состоит в нахождении множеств A_*^ϵ и Ω_*^ϵ.

С каждой точкой x_i свяжем некоторую ее окрестность X_i, таким образом $x_i \in X_i$. На каждом множестве X_i определим миноранту $\mu_i(x) : \mathbb{R}^n \to \mathbb{R}^m$, такую что $F(x) \geq \mu_i(x)$ для всех $x \in X_i$. Элемент покрытия — множество S_i, определяемый в однокритериальном случае соотношением (5), в случае многокритериальной оптимизации должен удовлетворять условию:

$$S_i \subseteq \{x \in X : \mu_i(x) \in \mathrm{NE}(U_i - \epsilon)\}. \tag{9}$$

Справедливо следующее утверждение, обобщающее теорему 1 на случай многокритериальной оптимизации.

Теорема 2. Пусть набор допустимых точек $N_k = \{x_1, \ldots, x_k\}$ таков, что выполнено условие покрытия $X \subseteq \cup_{i=1}^k S_i$, тогда множество Ω_k является ϵ-Парето множеством задачи (8).

3 Программная реализация метода неравномерных покрытий

Рассмотрим обобщение одной из возможных схем реализации метода неравномерных покрытий для задач с одним критерием — алгоритма бисекций [2]. Предполагается, что существует n-мерный параллелепипед P, с гранями, параллельными координатным плоскостям, содержащий допустимое множество X.

Обобщенный алгоритм бисекций:

$\mathcal{L}$ — список покрываемых параллелепипедов;

1. На первом шаге поместить в список $\mathcal{L}$ параллелепипед P: $\mathcal{L} = \{P\}$.
2. Выбрать параллелепипед $P_i \in \mathcal{L}$.
3. Выбрать точку $x_i \in P_i \cap X$ и при выполнении условия $f(x_i) < u_{i-1}$ изменить текущий рекорд u_i и рекордную точку x_i по формуле (3).

4. Используя миноранты, определить подмножества, заведомо не содержащие допустимых точек из $P_i \cap X$, либо точек, в которых может быть улучшен текущий рекорд более чем на ϵ.

5. Если совокупность полученных подмножества покрывает параллелепипед P_i, то выполнить переход к шагу 2. В противном случае, произвести сокращение и разбиение (с использованием найденных множеств) параллелепипеда P_i на несколько новых параллелепипедов, которые добавить к списку $\mathcal{L}$.

6. Если список пуст, то перейти к шагу 7, в противном случае перейти к шагу 2.

7. Если на шаге 3 не было найдено ни одной допустимой точки, то ограничения задачи несовместны. В противном случае рекордная точка будет ϵ-оптимальными решением задачи.

Обобщенный алгоритм бисекций может быть легко обобщен на случай многопритериальных задач.

Метод бисекций эффективно распараллеливается. На многопроцессорных системах с общей памятью несколько потоков независимо выполняют итерации 1-6 алгоритма, обращаясь к общему списку $\mathcal{L}$. Для предотвращения потерь, связанных с частой синхронизацией при доступе к общему списку, каждый поток также поддерживает локальный список подмножеств. Часть подмножеств из этого списка периодически копируется в общий. В момент, когда локальный список исчерпывается поток берет новое подмножество для обработки из общего списка. Реализация для систем с распределенной памятью следует той же самой схеме, но вместо копирования применяется передача сообщений по сети. Ключевым для управления процессом вычислений является выбор параметров, определяющих частоту обменов и число передаваемых данных.

Для уменьшения числа итераций применяются эвристические алгоритмы или методы локального поиска, позволяющие быстро находить рекордное значение, близкое к оптимальному. Тем самым усиливается отсев подмножеств, заведомо не содержащих оптимальных решений. Вычислительные ресурсы распределяются между процессами, выполняющими приведенную выше о бщую схему алгоритма, и процессами, выполняющими эвристики. Важнейшим фактором, влияющим на производительность, является соотношение количества процессоров, реализующих

эти два подхода. Более подробно параллельная реализация рассмотрена в работе [3].

4 Метод решения задач линейного программирования большой размерности

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. Ниже приводятся алгоритмы, позволяющие не только распараллелить процесс расчетов и тем самым ускорить вычисления, но и решать задачи значительно большей размерности. Предложенные алгоритмы реализованы на многопроцессорных вычислительных системах с распределенной памятью, их эффективность подтверждена вычислительными экспериментам и решением ряда практических задач.

Пусть прямая и двойственная задачи ЛП заданы в стандартной форме

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

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

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

$$\max_{p \in R^m}\ S(p, \beta, \hat{x}), \qquad (10)$$
$$\text{где } S(p, \beta, \hat{x}) = \{b^\top p - \tfrac{1}{2}\|(\hat{x} + A^\top p - \beta c)_+\|^2\}.$$

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 [4] теоремами.

Теорема 3. Существует такое число β_*, что при любом $\beta \geq \beta_*$ пара $[p(\beta), \beta]$, где $p(\beta)$ – решение задачи безусловной максимизации (10), определяет проекцию $\hat{x}^*$ точки $\hat{x}$ на множество решений X^* прямой задачи (P) по формуле

$$\hat{x}^* = (\hat{x} + A^T p(\beta) - \beta c)_+. \tag{11}$$

Теорема 4. 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 (10).

Итак, в результате решения двух задач безусловной максимизации вогнутой кусочно-квадратичной функции от m переменных получается проекция точки на множество прямой задачи (P) и некоторое решение двойственной задачи (D).

Следующий итерационный процесс дает одновременное решение прямой и двойственной задач ЛП

$$p_{s+1} \in \arg\max_{p \in R^m} \{b^T p - \tfrac{1}{2}\|(x_s + A^T p - \beta c)_+\|^2\} \tag{12}$$

$$x_{s+1} = (x_s + A^T p_{s+1} - \beta c)_+. \tag{13}$$

Здесь произвольный параметр $\beta > 0$ фиксирован.

Теорема 5. При любом $\beta > 0$ и для любой начальной точки x_0 итерационный процесс (12), (13) 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 (10) или (12) 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 (12), (13) 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 (12), а s - номер внешней итерации.

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

$$H_k = \delta I + A D_k A^T, \tag{14}$$

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

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

4. Найти направление максимизации Δp из решения линейной системы

$$H_k \Delta p = -G_k \tag{15}$$

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 $\tau_k = \max\limits_{\tau} S(p_k - \tau \Delta p, \beta, x_s)$ 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 (15) и ее решение. При реализации алгоритма распараллеливаются следующие операций:

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 (14);

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; столцовая 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 (15), 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 [4]. Расчеты проводились на параллельном вычислительном кластере МВС-6000IM, состоящем из двухпроцессорных узлов на основе Intel Itanium 2 с частотой 1.6 GHz, соединенными сетью Myrinet 2000 [5].

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] Евтушенко Ю.Г.Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // ЖВМ и МФ, 1971. Т. 11., №6, — С. 1390-1403.

[2] Евтушенко Ю.Г., Ратькин В.А. Метод половинных делений для глобальной оптимизации функции многих переменных. //Техническая кибернетика. 1987, №1, — С. 119-127.

[3] Evtushenko Y. Posypkin M. Sigal I. A framework for parallel large-scale global optimization.// Computer Science - Research and Development 23(3), 2009, — pp. 211-215.

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

[5] 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. Т. 49. №8. — С. 1369-1384.