ДЕКОМПОЗИЦИЯ ЗАДАЧ БЛОЧНОГО ПРОГРАММИРОВАНИЯ И АЛГОРИТМЫ ПРОЕКЦИИ [1]

А.С. Величко, Е.А. Нурминский

Институт автоматики и процессов управления ДВО РАН, Владивосток
e-mail: vandre@dvo.ru, nurmi@dvo.ru

Аннотация. В работе на примере блочной задачи линейного программирования изучаются подходы к декомпозиции задач блочного программирования со связывающими переменными с произвольным числом ограничений задачи с помощью последовательных и параллельных модификаций алгоритмов перемежающихся проекций.

Ключевые слова: фейеровский процесс, перемежающийся, проекция, параллельный, декомпозиция, блочное математическое программирование.

Введение

В работах [1, 2] рассматривались различные модификации алгоритма двойственных отсечений, применяемые для декомпозиции задач двублочного линейного программирования. Использование принципа построения отсечений для решения задачи условной выпуклой оптимизации позволило разработать алгоритмы с параллельной организацией вычислений. В случае произвольного количества блоков ограничений задачи может быть использована иерархическая организация вычислений [3], что существенно усложняет практическую реализацию данного алгоритма. В данной работе рассматривается альтернативный подход к декомпозиции, который заключается в построении различных модификаций, в том числе параллельных, алгоритмов перемежающихся проекций, которые используется для решения задач блочного программирования со связывающими переменными с произвольным количеством ограничений.

1. Постановка задачи

Рассматривается экстремальная задача

$$\min_{x \in V} h(x), \tag{1}$$

где $h(\cdot)$ – конечная выпуклая функция на множестве $V = \bigcap\limits_{i \in I_m} C_i$. Множество V представляет собой пересечение конечного числа выпуклых замкнутых множеств $C_i, I_m = \overline{1, M}$.

В данной работе для решения задачи (1) рассматривается класс фейеровских процессов [4] с дополнительным малым исчезающим воздействием, то есть итерационные последовательности $x^{s+1} = F(x^s + z^s), s = 0, 1, \ldots$, где F принадлежит классу фейеровских операторов, а $z^s \to 0$ при $s \to +\infty$.

Для построения различных модификаций алгоритмов в качестве оператора F предлагается использовать операторы проекций на множества C_i или их выпуклую комбинацию. Обстоятельный обзор модификаций проекционных алгоритмов в рамках

[1] Работа выполнена при поддержке РФФИ-ДВО РАН (проект 06-III-A-01-459)

подхода фейеровских процессов для решения задачи определения допустимой точки из пересечения замкнутых выпуклых множеств (convex feasibility problem) можно найти в работе [5]. Ранее была показана сходимость алгоритма перемежающихся проекций в предположениях $\lambda_s \to 0, \sum\limits_{s=0}^{+\infty} \lambda_s = +\infty$ без специальных предположений о начальном приближении x^0.

2. Алгоритмы проекции и декомпозиция задачи блочного линейного программирования

Рассмотрим задачу блочного линейного программирования с непустым допустимым множеством и конечной целевой функцией на множестве допустимых решений:

$$f(y^*) = \min_y \sum_{i=1}^m f_i(y), \text{ где } f_i(y) = \min_{z_i}\{c_i z_i : A_i z_i \le d_i - B_i y\}, i = 1, \ldots, m, \tag{2}$$

где y – вектор связывающих переменных, z_i – векторы блочных переменных в подзадачах определения функций $f_i(y)$, A_i, B_i – матрицы, а d_i – векторы соответствующих размерностей.

Обозначим допустимое множество решений задачи (2) за V_m. Первый подход к применению алгоритма перемежающихся проекций заключается в представлении допустимого множества решений задачи (2) в виде пересечения множеств C_i. Пусть $C_i = \{(z_i, y) \ : A_i z_i \le d_i - B_i y\}$ – замкнутые выпуклые множества, $i \in I_m = \overline{1, m}$. По построению множеств C_i очевидно, что

$$V_m = V = \bigcap_{i \in I_m} C_i,$$

где V – допустимое множество в определении задачи (1). Определим вектор $x = (z_1, z_2, \ldots, z_m, y)'$, где $'$ – операция транспонирования, тогда в обозначениях задачи (1) имеем

$$h(x) = \sum_{i=1}^m f_i(y) = cx, c = (c_1, c_2, \ldots, c_m, 0)'.$$

Альтернативный подход к применению алгоритма перемежающихся проекций заключается в другом описании множества V_m. Пусть $C_i = \{(z_i, u_i) \ : A_i z_i \le d_i - B_i u_i\}$ – замкнутые выпуклые множества, $i \in I_m$. Определим множество $C_{m+1} = \{u_1 = u_2 = \ldots = u_m \equiv y\}$, где u_i – фиктивные связывающие переменные, моделирующие вектор связывающих переменных y задачи (2). По построению множеств C_i очевидно, что

$$V_m = V = \bigcap_{i \in I_{m+1}} C_i, \text{ где } I_{m+1} = \overline{1, m+1}.$$

Обозначим $x = (z_1, z_2, \ldots, z_m, u_1, u_2, \ldots, u_m)'$, где $'$ – операция транспонирования, тогда в обозначениях задачи (1) имеем

$$h(x) = \sum_{i=1}^m f_i(u_i) = cx, c = (c_1, c_2, \ldots, c_m, 0, \ldots, 0)'.$$

Рассмотрим более подробно первый подход к описанию допустимого множества V_m. Алгоритм перемежающихся проекций для решения задачи (1) представляет собой итерационный процесс $x^{s+1} = P_s(x^s - \lambda_s g^s), s = 0, 1, \ldots$, где P_s – оператор проекции вектора

$x^s - \lambda_s g^s$ на множество C_{i_s}, индексы $\{i_s\}$ последовательно принимают значения из множества I_m, то есть $i_0 = 1, i_1 = 2, \ldots, i_{m-1} = m, i_m = 1, \ldots$, вектор градиента g^s не зависит от x^s и равен вектору c, λ_s – шаговые множители. Начальное приближение x^0 и шаговые множители λ_s лучше выбирать с учетом специфики задачи. В силу сходимости алгоритма при $s \to +\infty$ в качестве критерия останова используем условие $\|x^{s+1} - x^s\| \leq \varepsilon$ для некоторого малого, наперед заданного числа $\varepsilon > 0$.

Вектор $x^{s+1} = P_s(x^s - \lambda_s c)$ определяется в результате решения задачи

$$\min_{x \in C_{i_s}} \|x - x^s + \lambda_s c\|^2. \tag{3}$$

Расмотрим задачу для некоторого $i = i_s$. Поскольку множество C_i содержит ограничения только на векторы z_i, y, тогда при решении задачи (3) будут определены оптимальные значения для векторов $\tilde{z}_i, \tilde{y}_i$ и $z^{s+1} = \tilde{z}_i, y^{s+1} = \tilde{y}_i, i \in I_m$, а $z_j^{s+1} = z_j^s - \lambda_s c$ для $j \in I_m \backslash \{i\}$.

Рассмотрим теперь второй подход к описанию допустимого множества V_m. В этом случае алгоритм перемежающихся проекций для решения задачи (1) представляет собой итерационный процесс $x^{s+1} = P_s(x^s - \lambda_s g^s), s = 0, 1, \ldots$, где P_s – оператор проекции вектора $x^s - \lambda_s g^s$ на множество C_{i_s}, индексы $\{i_s\}$ последовательно принимают значения из множества I_{m+1}, то есть $i_0 = 1, i_1 = 2, \ldots, i_m = m+1, i_{m+1} = 1, \ldots$, вектор градиента g^s не зависит от x^s и равен вектору c

Вектор $x^{s+1} = P_s(x^s - \lambda_s c)$ определяется в результате решения задачи (3). Расмотрим задачу для некоторого $i = i_s$. Поскольку множество C_i содержит ограничения только на векторы z_i, u_i, тогда получим $z_j^{s+1} = z_j^s - \lambda_s c$ и $u_j^{s+1} = u_j^s$ для $j \in I_m \backslash \{i\}$. Задача нахождения проекции на множество C_{m+1} приводит к определению оптимальных $\tilde{u}_i = \frac{1}{m} \sum\limits_{i=1}^{m} u_i^s$ и $\tilde{z}_i = z_i^s$ для $i \in I_m$. Далее полагаем $u_i^{s+1} = \tilde{u}_i$ и $z_i^{s+1} = \tilde{z}_i$.

В качестве фейеровского оператора F может быть выбрана выпуклая комбинация операторов проекций на множества C_i, как для первого, так и для второго способа представления множества V_m, то есть $F(x) = \sum\limits_{j=1}^{m+1} w_j P_j(x), \sum\limits_{j=1}^{m+1} w_j = 1, w_j > 0$. В этом случае итерационный процесс алгоритма перемежающихся проекций имеет вид $x^{s+1} = \sum\limits_{j=1}^{m+1} w_j P_j(x^s - \lambda_s c), s = 0, 1, \ldots$, где P_j – оператор проекции вектора $x^s - \lambda_s c$ на множество $C_j, j \in I_{m+1}$. На $(s+1)$-ом шаге определяются оптимальные значения z_i^*, u_i^* в результате решения задачи проекции вектора $x^s - \lambda_s c$ на множество C_i. Тогда для $i \in I_m$ получаем $z_i^{s+1} = w_i z_i^* + (1 - w_i)(z_i^s - \lambda_s c)$, и с учетом задачи проекции на множество C_{m+1} имеем $u_i^{s+1} = w_i u_i^* + \sum\limits_{j \neq i} w_j u_j^s + \frac{w_{m+1}}{m} \sum\limits_{j=1}^{m} u_j^*$.

Каждая из m задач поиска проекции содержит только один блок ограничений, что позволяет надеяться на уменьшение объема вычислений и количества используемой памяти для хранения данных по каждому блоку ограничений по сравнению с решением задачи (2). Платой за уменьшение количества ограничений при решении каждой из задач проекции является необходимость решения квадратичной, а не линейной задачи, что впрочем не является существенной потерей при решении выпуклых оптимизационных задач с нелинейным функционалом и нелинейными ограничениями.

Недостатком первого подхода к представлению допустимого множества решений задачи (2) является последовательный характер возникающей модификации алгоритма перемежающихся проекций по вектору связывающих переменных y, что затрудняет распараллеливание процесса вычислений. Использование выпуклой комбинации операторов проекций решает указанную проблему, однако, ставит задачу обоснованного выбора коэффициентов w_j.

Второй подход с использованием искусственного множества C_{m+1} позволяет предложить параллельную модификацию алгоритма перемежающихся проекций, не прибегая к использованию выпуклой комбинации операторов проекций. В этом случае вычислений проекций на отдельные множества C_i можно осуществлять параллельно на m процессорах. Процесс $m+1$, реализующий решение задачи проекции на множество C_{m+1}, осуществляет пересчет только фиктивных векторов связывающих переменных $u_1 = u_2 = \ldots = u_m \equiv y$, уже полученных в результате решения задач проекции на множества $C_i, i \in I_m$.

3. Выбор шаговых множителей и оценка скорости сходимости алгоритма.

Выбор шаговых множителей может повлиять на скорость сходимости алгоритмов проекций, применяемых для решения задачи (2). Условия, требуемые для обеспечения сходимости алгоритма $\lambda_s \to 0, \sum\limits_{s=0}^{+\infty} \lambda_s = +\infty$ являются достаточно общими, но неконструктивными. Пусть x^* – решение задачи (2). Рассмотрим алгоритм перемежающихся проекций в виде итерационного процесса $x^{s+1} = P_s(x^s - \lambda_s c)$, где x^{s+1} получается в результате решения задачи $\min\limits_{x \in C_{i_s}} \|x - x^s + \lambda_s c\|^2$.

Получим для вектора $\bar{x}^s = x^s - \lambda_s c$, что

$$\|\bar{x}^s - x^*\|^2 = \|x^s - \lambda_s c - x^*\|^2 = \lambda_s^2 \|c\|^2 - 2\lambda_s c(x^s - x^*) + \|x^s - x^*\|^2.$$

Из неравенства Коши-Буняковского следует, что $c(x^s - x^*) = p_s \cdot \|c\| \cdot \|x^s - x^*\|$, где $|p_s| \leq 1$. Тогда

$$\|\bar{x}^s - x^*\|^2 = \lambda_s^2 \|c\|^2 - 2\lambda_s p_s \|c\| \|x^s - x^*\| + p_s^2 \|x^s - x^*\|^2 + (1 - p_s^2)\|x^s - x^*\|^2 =$$

$$= (\lambda_s \|c\| - p_s \|x^s - x^*\|)^2 + (1 - p_s^2)\|x^s - x^*\|^2.$$

Выбирая шаговые множители по правилу

$$\lambda_s = \frac{p_s \|x^s - x^*\|}{\|c\|} = \frac{c(x^s - x^*)}{\|c\|^2} = \frac{h(x^s) - h(x^*)}{\|c\|^2},$$

получим, что

$$\|\bar{x}^s - x^*\| = \theta_s \|x^s - x^*\|, 0 \leq \theta_s = \sqrt{1 - p_s^2} \leq 1. \tag{4}$$

С учетом свойства локальной сильной фейеровости для оператора проекции P_s или невырожденной выпуклой комбинации операторов проекции, то есть $\|P_s(x) - v\| \leq q_s \|x - v\|$ для любого $v \in V$ и $x = x^s \notin C_{i_s}, 0 \leq q_s < 1$ с учетом (4) получим для $x^{s+1} = P_s(\bar{x}^s)$, что

$$\|x^{s+1} - x^s\| = \|P_s(\bar{x}^s) - x^*\| \leq q_s \|\bar{x}^s - x^*\| = \gamma_s \|x^s - x^*\|, 0 \leq \gamma_s = \theta_s q_s < 1.$$

Оценивая величину $\|x^{s+1} - x^s\|$ получим

$$\|\bar{x}^s - x^s\| = \|x^s - \lambda_s c - x^s\| = \lambda_s \|c\|.$$

Тогда более конструктивное условие выбора шаговых множителей $\lambda_s = p_s \|x^s - x^{s-1}\| / \|c\|$, где p_s необходимо выбирать на интервале $(0, 1)$. При таком выборе шаговых множителей

$\|x^{s+1} - x^s\| = p_s\|x^s - x^{s-1}\|$. Выбор константы p_s в этом случае не предопределен неравенством Коши-Буняковского, например, p_s можно выбрать в виде константы, не зависящей от s. С одной стороны, выбор p_s близкими к нулю позволяет надеяться на ускорение сходимости алгоритма, с другой стороны, в этом случае шаговые множители по абсолютной величине уменьшаются при $s \to +\infty$.

Список литературы

[1] А.С. Величко *Об алгоритме двойственных отсечений для задачи двуэтапного стохастического программирования.* - Известия ВУЗов. Математика, 2006, N4, с. 78-81.

[2] А.С. Величко *О минимальном наборе отсечений для алгоритма прямо-двойственной декомпозиции.* - Информационный бюллетень ассоциации математического программирования. Екатеринбург, 2007, N11, с. 23-24.

[3] А.С. Величко *Иерархическая параллельная декомпозиция многоблочных оптимизационных задач.* - Российская конференция "Дискретная оптимизация и исследование операций": Материалы конференции. Новосибирск: Изд-во Института математики, 2007, с. 142.

[4] В.В. Васин, И.И Еремин *Операторы и итерационные процессы фейеровского типа. Теория и приложения.* Екатеринбург, 2005, 200 с.

[5] H. Bauschke, J. Borwein *On projection algorithms for solving convex feasibility problems.* - CECM Research Report 95:034, SIAM Review, 1996, Vol. 38, N3, p. 367-426.

DECOMPOSITION OF BLOCK-ANGULAR MATHEMATICAL PROGRAMMING PROBLEMS USING PROJECTIONS METHODS

A.S. Velichko, E.A Nurminskiy

Institute of Automation and Control Processes FEB RAS, Vladivostok
e-mail: vandre@dvo.ru, nurmi@dvo.ru

Abstract. In this paper approaches to decomposition of structured (block-angular) mathematical programming problems with any number of constraints on linear programming example using iterative and parallel modifications of intermittent projections methods are discussed.

Key words: fejer process, intermittent, projection, parallel, decomposition, structured, block-angular, mathematical programming