# Параллельная проектирование на выпуклую оболочку семейства множеств *

Нурминский Е.А.

**Аннотация**

Предложен параллельный метод проектирования на выпуклую оболочку конечного семейства подмножеств конечномерного векторного пространства. Приведена принципиальная схема алгоритма и доказана его сходимость.

В теории и практике математического программирования часто встречается задача

$$\min_{x \in X} \|x\| = \|z^*\| \tag{1}$$

нахождения в заданном множестве $X$ элемента с минимальной нормой $\|\cdot\|$. На (1) можно смотреть и как на нахождение проекции начала координат на множество $X$, поэтому мы будем называть эту проблему задачей проекции.

Такая задача встречается в теории двойственности ( построение отделяющей гиперплоскости ), обобщенном решении несовместных задач, распознавании образов, оценке параметров, методах условной оптимизации и пр. Чаще всего используется евклидовая норма, однако, особенно в экономических приложениях, встречаются и другие.

Следует отметить, что на практике приходится чаще всего находить проекцию на множество, конструируемое с помощью некоторых базовых операций из более "простых" элементов. Наиболее известный пример — это система ограничений

$$X = \{x : g_i(x) \leq 0, i = 1, 2, \dots, M\},$$

которая представляет собой пересечение семейства множеств $X_i, i = 1, 2, \dots, M$, т.е. $X = \cap_{i=1}^{M} X_i$, а каждое $X_i$ задается единичным неравенством $X_i = \{x : g_i(x) \leq 0\}$.

Даже при аффинных $g_i(\cdot)$ проекция на множество $X$ представляет собой достаточно нетривиальную задачу, сложность которой существенно выше, чем простейшая операция проекции на отдельное полупространство $X_i$. Это уже давно [1, 2, 3] и, правда, для более простой задачи допустимости ( т.е. нахождении какого-либо $x \in X$ ), породило идею последовательной циклической проекции (CSP), т.е. к последовательности проекций на отдельные множества $X_i$:

$$\min_{x \in X_i} \|x\| = \|x^i\| \tag{2}$$

когда решение предыдущей задачи $x^i$ проектируется на последующее множество $X_{i+1}$. При исчерпании всего списка множеств $X_i$, осуществляется снова перепроектирование на первое множество и т.д. В дальнейшем, с распространением параллельных вычислений, эта идея была модифицирована в том плане, что проекции выполнялись одновременно, а затем результаты усреднялись с какими-то весами (см, например, [4] и ссылки там). Эти алгоритмы получили название PBS-методов.

В настоящей работе рассмотрен другой способ конструирования множества $X$, а именно, с помощью операции взятия выпуклой оболочки.

*Представлено и опубликовано в *Известия вузов. Сер. Математика.-2003.-№12 (499).- С. 78-82.*

**Определение 1** *Выпуклой оболочкой семейства множеств* $X_i, i = 1, 2, \dots, M$ *называется множество*

$$X = \{x = \sum_{i=1}^{M} \lambda_i x^i, \text{ где } x^i \in X_i, \lambda_i \geq 0, i = 1, 2, \dots, M; \sum_{i=1}^{M} \lambda_i = 1\}. \tag{3}$$

Выпуклую оболочку $X_i, i = 1, 2, \dots, M$ будем обозначать

$$X = \mathrm{co}\,\{X_i, i = 1, 2, \dots, M\}. \tag{4}$$

Множество $X$, построенное таким образом, выпукло, более подробное рассмотрение свойств выпуклых оболочек можно найти, например, в [5].

Множества $X_i$ могут быть различной природы, которую мы на данном этапе не конкретизируем. Поскольку

$$\mathrm{co}\,\{X_i, i = 1, 2, \dots, M\} = \mathrm{co}\,\{\mathrm{co}\,\{X_i\}, i = 1, 2, \dots, M\}$$

то не ограничивая общности эти множества можно считать выпуклыми. В последующем мы также будем считать норму $\|\cdot\|$ евклидовой, т.е. индуцированой скалярным произведением $\|x\| = \sqrt{xx}$.

Классическим примером представления множества в виде выпуклой оболочки является, конечно, допустимое множество задачи линейного программирования, которое, при условии его ограниченности, представимо в виде (4) с одноточечными множествами $X_i$, представляющими собой крайние точки полиэдра допустимости. Заметим, что мы можем агрегировать крайние точки ( или множества $X_i$ ) и получать различные представления $X$ в виде (4).

Наша основная цель заключается в том, чтобы предложить алгоритм, использующий параллельное решение задач типа (2) и их последующее агрегирование для того чтобы получить решение исходной задачи (1) с множеством (3). Неявное предположение при этом, которое тем не менее следует иметь в виду, состоит в том, что задачи проекции типа (2) непосредственно с $X_i$ или их простыми модификациями достаточно легко разрешимы по сравнению с первоначальной задачей (1).

# 1 Алгоритм

Предлагаемый метод решения задачи (1)-(3) имеет итеративный характер и состоит из начальной инициализации:

**Начало.** Положить $k = 0$ ( счетчик итераций ) и выбрать начальное приближение $z^0 \in X$ для решения задач (1)–(3).

и последующих итераций, каждая из которых состоит из следующих шагов:

**Шаг 1.** Присоединить к каждому множеству $X_i$ точку $z^k$, т.е. сформировать множества $\bar{X}_i = \mathrm{co}\,\{X_i, z^k\}$ и для каждого $i = 1, 2, \dots, M$ решить задачу

$$\min_{x \in \bar{X}_i} \|x\| = \|x^{i,k}\| \tag{5}$$

**Шаг 2.** Из точек $x^{i,k}, i = 1, 2, \dots, M$ составить множество $Z_k = \mathrm{co}\,\{x^{i,k}, i = 1, 2, \dots, M\}$ и решить координирующую задачу

$$\min_{x \in Z_k} \|z\| = \|z^{k+1}\| \tag{6}$$

**Шаг 3.** Увеличить счетчик итераций $k \to k + 1$, повторить итерацию.

С точки зрения параллельных вычислений координирующая задача (6) существенно уменьшает степень параллелизма, однако можно предположить, что в реальных параллельных вычислительных системах количество $M$ одновременно выполняющихся блоков (5) не слишком велико.

Фактически оно должнно соответствовать архитектуре вычислительной системы , т.е. количеству процессорных элементов. В параллельных вычислителях массового применения это количество, как правило, не превосходит нескольких десятков ( см. статистику [6] ). Соответственно задача (6) имеет относительно небольшой обьем и ее вклад в общий обьем вычислений не слишком велик. Таким образом при крупнозернистой параллелизации, где основные вычислительные затраты связаны с решением задач (5), замедление, вызванное синхронизирующим барьером (6), не будет оказывать черезмерно большого негативного влияния.

Заметим. что мы не предписывает каких-либо определенных способов решения задач (5)–(6). Эти задачи могут решаться итеративно или, при их разумных обьемах, конечными методами квадратичного программирования. При итеративном решении возникают, однако, интересные вопросы контроля точности вычислений и использования приближенных решений (5), что будет исследовано в дальнейшем. В настоящий момент мы рассмотрим лишь принципиальную схему алгоритма и убедимся в ее разумности.

## 2 Анализ сходимости

Предварительно введем некоторые вспомогательные обозначения и приведем, для самодостаточности изложения, некоторые полезные свойства выпукцлых множеств.

Обозначим через $(X)_z$ опорную функцию множества $X$:

$$(X)_z = \sup_{x \in X} zx.$$

В этих обозначениях проекция начала координат на множество $X$ определяется следующей леммой.

**Лемма 2** *Точка $z^* \in X$ является решением задачи (1) тогда и толко тогда, когда*

$$\|z^*\|^2 + (X)_{-z^*} \leq 0.$$

Д о к а з а т е л ь с т в о. Фактически, это просто другой способ записать вариационную форму условий оптимальности $z^*$, которая выглядит как

$$\|z^*\|^2 \leq z^* x \text{ для всех } x \in X.$$

□

Если множество $X$ задается соотношением (3), то его опорная функция легко вычисляется через опорные функции составляющих $X_i, i = 1, 2, \ldots, M$. Обозначим через $\Delta_M$ множество неотрицательных весов $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_M)$, нормированных на единицу:

$$\Delta_M = \{\lambda : \sum_{i=1}^{M} \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, \ldots, M\}.$$

Тогда справедлива

**Лемма 3** *Пусть $X = \mathrm{co}\{X_i, i = 1, 2, \ldots, M\}$, тогда*

$$(X)_z = \sup_{\lambda \in \Delta_M} \sum_{i=1}^{M} \lambda_i (X_i)_z = \max_{i=1,2,\ldots,M} (X_i)_z .$$

Д о к а з а т е л ь с т в о. Следует из определения опорной функции. $\square$

В качестве предварительного результата касающегося сходимости алгоритма раздела 1 покажем монотонное убывание и, следовательно, равномерную ограниченность длины вектора $z^k$.

**Лемма 4** *Последовательность $\{\|z^k\|\}$ монотонно убывает и, следовательно, ограничена в совокупности.*

Д о к а з а т е л ь с т в о. Следует из оценок:

$$\|z^{k+1}\|^2 \le \min_{i=1,2,\ldots,M} \|x^{i,k}\|^2 = \min_{i=1,2,\ldots,M} \min_{x \in \mathrm{co}\left(X_i, z^k\right)} \|x\|^2 \le \min_{i=1,2,\ldots,M} \|z^k\|^2 = \|z^k\|^2.$$

$\square$

Из леммы 4 немедленно следует также и существование предела $\lim_{k\to\infty} \|z^k\| = \rho$.

Основной результат о сходимости алгоритма раздела 1 представляет собой следующее утверждение.

**Теорема 5** *Последовательность $\{z^k\}$ сходится к единственной предельной точке $z^*$, являющейся решением задачи (1).*

Д о к а з а т е л ь с т в о. Заметим, что задача (1) имеет единственное решение в силу строгой квазивыпуклости евклидовой нормы, поэтому для доказательства утверждения теоремы достаточно показать, что любая предельная точка последовательности $\{z^k\}$ является ее решением.

Предположим, поэтому, что $\bar{z}$ — некоторая предельная точка $\{z^k\}$, которая существует в силу леммы 4. и пусть $K$ — множество индексов, таких, что

$$\lim_{k\in K, k\to\infty} z^{k+1} = \bar{z}.$$

Из оценок

$$\rho \le \|z^{k+1}\| \le \|x^{i,k}\| \le \min_{x\in \mathrm{co}\left(X_i, z^k\right)} \|x\| \le \|z^k\|$$

следует, что и $\|x^{i,k}\|$ монотонно убывают и имеют тот же предел $\rho$, что и $\|z^k\|$.

Пусть $\bar{x}^i$ — некоторые предельные точки последовательностей $\{x^{i,k}, k \in K\}$. Поскольку

$$z^{k+1} \in \mathrm{co}\,\{x^{i,k}, i = 1, 2, \ldots, M\}$$

то и

$$\bar{z} \in \mathrm{co}\,\{\bar{x}^i, i = 1, 2, \ldots, M\}$$

с $\rho = \|\bar{z}\| = \|\bar{x}^i\|, i = 1, 2, \ldots, M$. Последнее возможно только в том случае, когда $\bar{z} = \bar{x}^i, i = 1, 2, \ldots, M$.

Далее, по построению

$$\|x^{i,k}\|^2 + (X_i)_{-x^{i,k}} \le \|x^{i,k}\|^2 + \left(\mathrm{co}\,\{X_i, z^k\}\right)_{-x^{i,k}} \le 0, i = 1, 2, \ldots, M$$

и переходя к пределу по $k \to \infty, k \in K$ получаем

$$\|\bar{z}\|^2 + (X_i)_{-\bar{z}} \le 0, i = 1, 2, \ldots, M. \tag{7}$$

Отсюда и

$$\|\bar{z}\|^2 + \max_{i=1,2,\ldots,M} (X_i)_{-\bar{z}} = \|\bar{z}\|^2 + (X)_{-\bar{z}} \le 0$$

и, следовательно, $\bar{z}$ — решение задачи (1), т.е. $\bar{z} = z^*$. Единственность этого решения влечет за собой и сходимость последовательности $z^k$:

$$\lim_{k\to\infty} z^k = z^*$$

причем одновременно и $x^{i,k} \to z^*$ для всех $i = 1, 2, \ldots, M$. $\square$

## Список литературы


[1] Kaczmarz S. Angenherte auflösung von systemen linearer gleichungen, Bull. Acad. Polon. Sci. Lett., A 35 ( 1937) 355-357.

[2] Брегман Л.М. Метод последовательных проекций для поиска общей точки выпуклых множеств ДАН СССР, 1965, стр. 688-692.

[3] Губин Л.Г., Поляк Б.Т., Райк Е.В. Метод проекции для нахождения общей точки выпуклых множеств, Журн. вычисл. матем. и мат. физики, т. 7, 1, 1967, стр. 1-24.

[4] Censor Y., Cohen N., Kutscher ( Kotzer ) T., Shamir J., Summed squared distance error reduction by simultaneous multiprojections and applications, Applies Mathematics and Computations, 126 ( 2002 ), pp. 157-179.

[5] Лейхтвейс К., Выпуклые множества:Пер. с нем.-М.: Наука. Главная редакция физико-математической литературы, 1985.-336 с.

[6] Учебно-информационный центр по параллельным вычислениям: `http://parallel.ru`.