ПАРАЛЛЕЛЬНЫЙ ПРЯМО-ДВОЙСТВЕННЫЙ АЛГОРИТМ ОТСЕЧЕНИЙ ДЛЯ РЕШЕНИЯ СТРУКТУРИРОВАННЫХ ЛИНЕЙНЫХ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

А.С. Величко

Институт автоматики и процессов управления ДВО РАН, Владивосток

Введение

Актуальность проблемы решения оптимизационных задач связана с усложнением формальных постановок количественных моделей исследования операций, математического моделирования, развитием новых моделей в экономике и финансовой математике. Зачастую структура матрицы ограничений экстремальной задачи обладает структурными особенностями, предоставляющими возможности для «крупнозернистого» представления задачи на относительно слабо связанные блоки меньшей размерности.

Алгоритмы решения больших (порядка 10^4 переменных/ограничений) и сверхбольших (порядка 10^5 – 10^6 переменных/ограничений) линейных экстремальных задач и систем линейных уравнений остаются важной частью арсенала математического программирования, исследования операций, и повышение их вычислительной эффективности имеет важное практическое значение. С распространением подходов, основанных на параллельных вычислениях, возникла проблема параллелизации симплексоподобных алгоритмов, все еще играющих важную роль в приложениях. Среди таковых надо особо отметить потребности в эффективном решении серий задач, являющихся небольшой модификацией друг друга, в чем современные симплексоподобные алгоритмы являются наиболее успешными. Подобные ситуации типичны, например, для приложений, в которых линейное программирование используется для решения внутренних вспомогательных задач типа метода линеаризации, итеративных проективных алгоритмов, методов негладкой оптимизации, целочисленного программирования и др.

Большой потенциал заключается, однако, в использовании параллельных вычислений для решения задач большой размерности со специальной, но часто встречающейся структурой. В первую очередь это относится к блочно-диагональной структуре матрицы ограничений и применению методов типа методов Данцига−Вульфа.

Постановка структурированной оптимизационной задачи

В работе [1] был предложен прямо-двойственный алгоритм, используемый для решения двублочных оптимизационных задач, и доказана его сходимость при достаточно общих предположениях. В качестве основной модели структурированной оптимизационной задачи рассмотрим двублочную проблему следующего вида:

$$\min_{z_A, z_B, x} c_A z_A + c_B z_B, \tag{1}$$

$$A_A z_A + B_A x \le d_A, \tag{2}$$

$$A_B z_B + B_B x \le d_B, \tag{3}$$

$$z_A, z_B \ge 0, \tag{4}$$

где z_A, z_B – векторы переменных задачи; x – вектор связывающих переменных; A_A, B_A, A_B, B_B – матрицы соответствующих размерностей.

При фиксированном x эта задача распадается на два независимых блока, что используется для развития декомпозиционного алгоритма двойственных отсечений.

Введя функции отклика для $M=\{A,B\}$

$$f_M(x) = \min_{z_M \ge 0} c_M z_M, A_M z_M \le d_M - B_M x, \tag{5}$$

получим эквивалентную для (1)−(4) задачу

$$\min_x f_A(x) + f_B(x). \tag{6}$$

Используя сопряженные функции $h_M(p) = \max_x \{px - f_M(x)\}$ задачу (6) легко переписать в терминах сопряженных функций:

$$\min_p h_A(-p) + h_B(p). \tag{7}$$

Такая эквивалентность задач (6) и (7) позволяет организовать эффективный процесс обмена координирующей информацией между прямыми и двойственными задачами линейного программирования в декомпозиционном подходе к решению задачи (1)−(4).

Разработка параллельного прямо-двойственного алгоритма отсечений

Суть алгоритма двойственных отсечений заключается в одновременном решении пары двойственных задач $\min_x f_A(x) + f^k(x)$ и $\min_p h^k(-p) + h_B(p)$. Внешние линейные аппроксимации для функций $f_B(x)$ и $h_A(p)$ имеют вид

$$f^k(x) = \max\{f^{k-1}(x), \bar{p}^k x - h_B(\bar{p}^k), p^k x - h_B(p^k)\},$$

$$h^k(p) = \max\{h^{k-1}(p), p\bar{x}^k - f_A(\bar{x}^k), px^{k+1} - f_A(x^{k+1})\},$$

а векторы $\bar{x}^k, \bar{p}^k$ являются субградиентами данных функций. Указанные задачи минимизации являются аппроксимированными вариантами задач

(6), (7). В результате их решения определяются субоптимальные векторы x^k, p^k. Вычисление значений функций $f_B(x)$, $h_A(p)$ и их субградиентов в точках x^k, $-p^k$ выступают подзадачами алгоритма. Решение подзадач на каждом шаге алгоритма позволяет строить новые линейные отсечения, которые уточняют кусочно-линейные аппроксимации $f^k(x)$ и $h^k(p)$. Блок-схема выполнения k-го шага МАДО представлена на рис. 1.

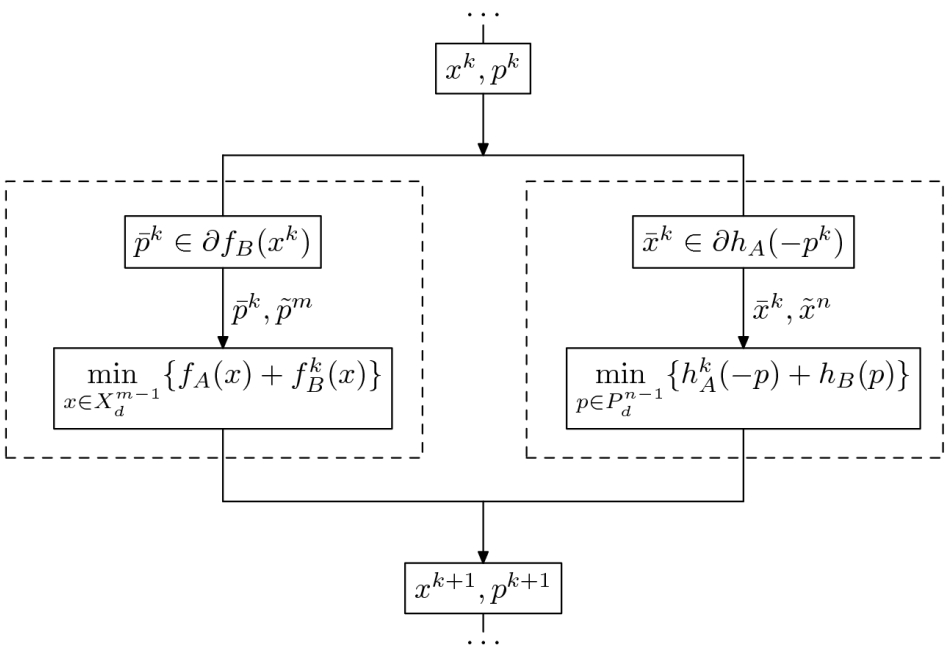


Рис. 1

В процессе выполнения алгоритма могут возникать некорректные задачи, в работе [2] показан способ построения дополнительных отсечений и модификации алгоритма в случаях недопустимости или неограниченности возникающих подзадач. Эта группа отсечений аппроксимирует множество определения функций $f_B(x)$ и $h_A(p)$. Множества

$$X_d^m = X_d^{m-1} \cup \{x : \tilde{p}^m x \le \tilde{g}^m\}, \quad P_d^n = P_d^{n-1} \cup \{p : \tilde{x}^m p \le 0\}$$

обновляются в случае возникновения неограниченных или недопустимых подзадач алгоритма, когда генерируемые алгоритмом векторы x^k, $-p^k$ не принадлежат области определения функций $f_B(x)$ и $h_A(p)$.

Возможны различные стратегии построения отсечений, они показаны на рис. 2, 3. После выполнения вычислений в каждом из блоков осуществляется обмен значениями x^{k+1} и p^{k+1} между процессами для вычислений на следующей итерации алгоритма. Более подробно вычислительные аспекты использования таких прямо-двойственных отсечений описаны, в частности, в работах [1,2].

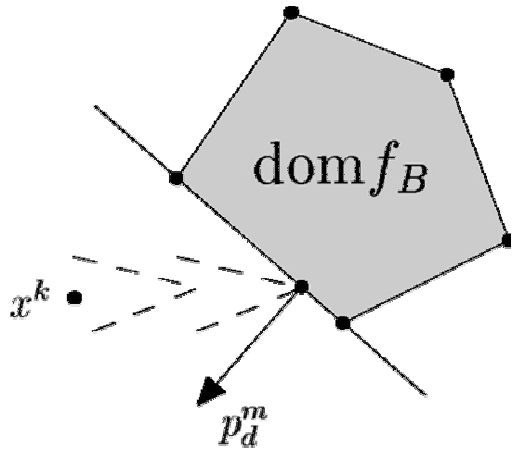


Рис. 2

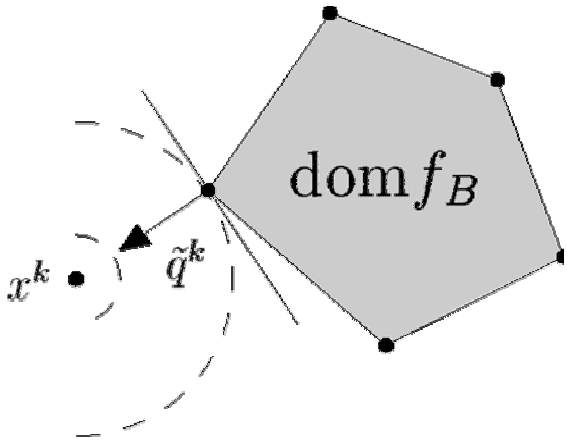


Рис. 3

Важными вычислительными задачами алгоритма являются вычисления субградиентов функций $f_B(x)$ и $h_A(p)$. Пусть $f_B(x)$ определена соотношением (5). Тогда для некоторого фиксированного вектора $x*$ имеем

$$f_B(x) = \min_{z_M \geq 0} c_B z_B, A_B z_B \leq d_B - B_B x, x = x*. \tag{8}$$

Из стандартной теории двойственности линейного программирования и теории выпуклого анализа легко получить следующее утверждение.

Утверждение. *Пусть для заданного $x*$ задача (8) разрешима, $p*$ является вектором оптимальных двойственных переменных к ограничению $x = x*$ в задаче (8). Тогда $p* \in \partial f_B(x*)$, и обратно, любой вектор $p* \in \partial f_B(x*)$ является вектором оптимальных двойственных переменных, соответствующих ограничению $x = x*$ в задаче (8).*

Направлением дополнительного повышения численной эффективности алгоритма является использование е-отсечений, основанных на приближенном решении подзадач алгоритма. Для этого необходимо уметь вычислять е-субградиенты кусочно-линейных функций отклика. Важным для приложений является вопрос использования минимальной доступной прямо-двойственной информации, получаемой в процессе приближенного итеративного решения подзадач.

В приведенных выше формулировках функции $f_B(x)$ и $h_A(p)$ являются выпуклыми кусочно-линейными функциями, в этих условиях удается доказать сходимость алгоритма с использованием приближенных решений подзадач.

Теорема. *Пусть $f_B(x)$ и $h_A(p)$, определенные в (5), являются конечными, выпуклыми и кусочно-линейными, задачи поиска субградиентов функций $f_B(x)$ и $h_A(p)$ решаются приближенно с некоторой точностью ε_k и*

$\lim_{k\to\infty} \varepsilon_k = 0$. *Последовательности* $\{x^k\}$, $\{p^k\}$, *генерируемые алгоритмом, ограничены. Тогда все предельные точки последовательностей* $\{x^k\}$, $\{p^k\}$ *являются решениями задач (6), (7) соответственно.*

Теорема демонстрирует, что вместо использования точных значений субградиентов функций отклика на каждом шаге алгоритма можно вычислять последовательность их е-субградиентов, сходящихся по мере выполнения алгоритма к точным субградиентам.

Задача двушагового стохастического программирования в блочной форме

Рассмотрим процесс декомпозиции оптимизационной задачи на блоки ограничений на примере задачи двушагового стохастического программирования. В линейных оптимизационных задачах стохастического программирования, возникающих в ситуации неопределенности, когда переменные модели являются случайными величинами, возникает следующая математическая постановка вычисления оптимального вектора x:

$$\min_{x\geq 0} c(w)x, S(w)x = h(w),$$

где w – элементарное событие на заданном пространстве событий Ω.

Очевидно, что решение x зависит от реализации определенного события *w*. Однако если решение этой задачи должно быть определено *a priori*, вряд ли найдется вектор x, удовлетворяющий ограничениям задачи для всех событий $w \in \Omega$. Вектор $\delta(x,w) = h(w) - S(w)x$ характеризует степень нарушения ограничений исходной задачи при выборе какого-то x. В двушаговой модели стохастического программирования предполагается, что на первом шаге мы знаем лишь множество событий Ω, на втором шаге мы получаем полную информацию о наступившем событии *w*. В таком случае оптимальный выбор x на первом шаге может проводиться при одновременной минимизации среднего ожидаемого значения величины $c(w)x$ и некоторого (линейного) функционала от вектора $\delta(x,w)$. Выбирая в качестве $\delta(x,w)$ сумму абсолютных значений компонент, запишем исходную задачу в виде

$$\min_{x\geq 0}[\mathrm{M}\{c(w)x\} + \min_{y^+,y^-} \mathrm{M}\{ey^+ + ey^-, y^+ - y^- = h(w)\text{-}S(w)x\}, y^+ \geq 0, y^- \geq 0],$$

где *M* – символ математического ожидания. В случае, когда все множество событий Ω содержит всего два события w_1, w_2, вероятность наступления которых равны соответственно $p_1, p_2 = 1 - p_1$, последняя задача имеет вид

$$\min_{y^+,y^-,x} [(p_1c_1 + p_2c_2)x + p_1(ey_1^+ + ey_1^-) + p_2(ey_2^+ + ey_2^-)],$$

$$y_1^+ - y_1^- + S^1x = h^1,\ y_2^+ - y_2^- + S^2x = h^2,\ x, y_i^+, y_i^- \geq 0,$$

где e – вектор из единиц, $S^i = S(w_i)$, $h^i = h(w_i)$, *i=1,2*.

Ясно, что система ограничений данной задачи является блочной вида (1)–(4). Однако вектор связывающих переменных x в данной задаче входит в минимизируемый функционал.

Обозначим новый вектор связывающих переменных за $\tilde{x}$. Добавим ограничение $x = \tilde{x}$ и агрегируем его, например, с ограничениями $y_1^+ - y_1^- + S^1 x = h^1$. Дальнейшее агрегирование вектора x с векторами y^+, y^- позволяет исключить вхождение вектора x в функционал получаемой задачи, сохраняя двублочность ограничений.

В результате такой агрегации получим структурированную двублочную задачу

$$\min_{\tilde{y}_1, \tilde{y}_2, \tilde{x}} \tilde{c}_1 \tilde{y}_1 + \tilde{c}_2 \tilde{y}_2,$$

$$A_A \tilde{y}_1 + B_A \tilde{x} \le \tilde{h}_1,$$

$$A_B \tilde{y}_2 + S_2 \tilde{x} \le h_2,$$

$$\tilde{x}, \tilde{y}_1, \tilde{y}_2 \ge 0,$$

где $\tilde{x}$ – новый вектор связывающих переменных.

Реализация параллельного прямо-двойственного алгоритма отсечений и численные эксперименты

Для программной реализации параллельного алгоритма использована технология параллельных вычислений стандарта MPI реализации LAM. Вычисления организованы в виде процессов, обменивающихся сообщениями стандарта MPI. Разработанная программа используется на многопроцессорных вычислительных комплексах МВС-1000/16-17 в составе Центра коллективного пользования «Дальневосточный вычислительный ресурс» http://www.dvo.ru/bbc.

Для реализации операций линейной алгебры используется свободно распространяемая библиотека meschach. Для решения подзадач используется симплекс-метод с одновременным определением двойственных переменных и факторизацией базисной матрицы для ее более эффективного обращения. Для дополнительного сокращения вычислений применяется процедура «досчета», когда оптимальное решение задачи, полученное на k-м шаге алгоритма, используется в качестве допустимого решения для решения этой же задачи на (k+1)-м шаге.

В таблице представлена статистика тестовых примеров различной размерности для задачи двушагового стохастического программирования, использующих случайно генерируемые данные. «Количество ограничений» – общее количество ограничений задачи, столбец «Время 1» – время выполнения последовательной версии прямо-двойственного алгоритма в секундах, «Время 2» – время выполнения параллельной реализации алгоритма в секундах. В столбце «Ускорение» показано отношение времени

выполнения последовательной параллельной реализаций алгоритма, в последнем столбце «Выигрыш, %» показано относительное уменьшение времени счета при использовании параллельного алгоритма по сравнению с его последовательным вариантом.

Тест	Количество ограничений	Время 1 (с)	Время 2 (с)	Ускорение	Выигрыш, %
1	209	13,12	9,15	1,43	30,3
2	409	257,54	234,16	1,1	9,1
3	609	1680,79	1226,4	1,37	27,0
4	809	6697,7	5044,71	1,33	24,7
5	1009	17977,52	11975,57	1,50	33,4
6	1209	29252,07	27147,8	1,08	7,2
7	1409	54066,36	41857,47	1,29	22,6
8	1609	111173,05	82079,81	1,35	26,2
9	1809	198491,13	157503,63	1,26	20,6
10	2009	225800,69	200270,27	1,13	11,3

При использовании двух параллельно выполняемых процессов максимальное теоретическое значение коэффициента ускорения равно 2, когда время выполнения процессов на каждом шаге алгоритма одинаково при относительно небольшом объеме обмена данными между процессами. Анализ времени выполнения каждого из блоков алгоритма показывает, что происходит простой второго блока параллельного алгоритма до момента обмена значениями x^{k+1}, p^{k+1} между процессами и перехода на следующий шаг алгоритма, поскольку на решение задач первого блока затрачивается относительно большее время. Такой простой происходит в силу трудоемкости задачи поиска точного субградиента функции $f_B(x)$ в точке из-за невозможности использования описанной выше стратегии “досчета” в этой задаче.

На рис. 4 представлены результаты выполнения тестов для задачи двушагового стохастического программирования на случайно генерируемых данных. Алгоритм демонстрирует линейный характер сходимости, выигрыш от использования параллелизма по времени выполнения алгоритма составляет в большинстве тестов 20–25%.

Заключение

В работе рассмотрен способ организации параллельных вычислений для модифицированного алгоритма двойственных отсечений, разработанного в [1,2], применяемого для эффективной прямо-двойственной декомпозиции структурированных задач линейного программирования. Рассматривается постановка задачи двушагового стохастического программи-

рования, показано, что она может быть представлена в классе структурированных оптимизационных задач. На рис. 4 $2m+Nx$ – число ограничений задачи. Масштаб шкалы логарифмический по обеим осям. Линия обычной толщины соответствует последовательному, а жирная линия – параллельному алгоритму прямо-двойственных отсечений.

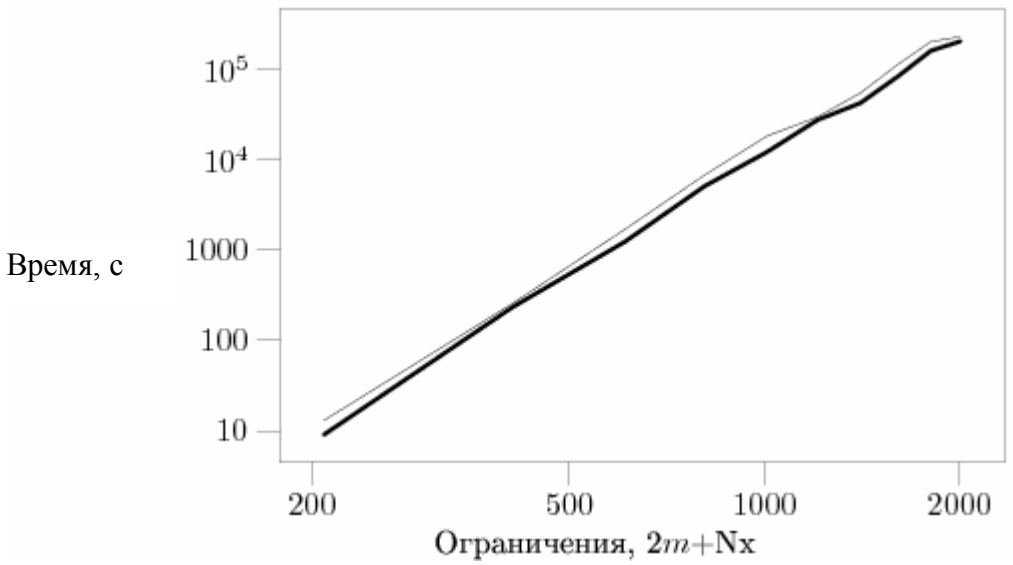


Рис. 4

Для решения подзадач алгоритма был реализован алгоритм симплекс-метода с одновременным вычислением двойственных переменных. Использована стратегия «досчета» допустимого решения подзадач, в качестве которого используется оптимальный базис этой подзадачи на предыдущем шаге алгоритма. Применение данной стратегии возможно для всего класса методов отсечений и обусловлено их спецификой. Использование процедуры LU-разложения для обращения базисной матрицы позволяет дополнительно сократить вычислительные затраты симплекс-метода.

Проведены вычислительные эксперименты с параллельным и последовательным версиями алгоритма на примере задачи двушагового стохастического программирования большой размерности со случайно генерируемыми данными. Алгоритмы демонстрируют линейную скорость сходимости, что наблюдалось авторами в их применении к задачам большой размерности в жестких экстремальных задачах с тривиальным носителем [2] и задаче финансовой математики о репликации портфеля рыночных активов [3]. В этих различных по структуре задачах используемый алгоритм показал свою устойчивость и эффективность, трудоемкость которого определяется в основном количеством связывающих переменных задачи.

Литература

1. Нурминский Е.А. Численные методы выпуклой оптимизации. – М.: Наука, 1991. − 165 с.
2. Величко А.С., Нурминский Е.А. Опыт декомпозиции метода конечных элементов с использованием теории структурированных оптимизационных задач // Электронный журнал «Исследовано в России». – 2002. № 113. – С. 1237–1256. – http://zhurnal.ape.relarn.ru/articles/2002/113.pdf
3. Величко А.С., Нурминский Е.А. Прямо-двойственная декомпозиция задачи о репликации портфеля рыночных активов // Автоматика и телемеханика. – 2004. – № 2. – С. 170–178.