

Опыт декомпозиции метода конечных элементов с использованием теории структурированных оптимизационных задач

А. С. Величко (vandre@dvo.ru), Е. А. Нурминский (nurmi@dvo.ru)

Институт Автоматики и Процессов Управления ДВО РАН

Аннотация

Большие и особенно сверхбольшие системы уравнений, порождаемые сложными задачами моделирования, как правило, имеют структурные особенности, предоставляющие определенные возможности для "крупнозернистой" декомпозиции задачи на относительно слабо связанные блоки. В данной работе подобные задачи рассматриваются как специальный класс структурированных задач линейного программирования, для решения которых применяется общий метод декомпозиции, модифицированный на случай неразрешимости подзадач. В качестве примера рассмотрено решение структурированной системы линейных уравнений метода конечных элементов, описывающей стационарное распределение температур в области сложной формы.

Введение

Большие и особенно сверхбольшие системы уравнений, порождаемые сложными задачами моделирования, как правило, имеют структурные особенности, предоставляющие определенные возможности для "крупнозернистой" декомпозиции задачи на относительно слабо связанные блоки. Такие структурные особенности могут быть связаны как с пространственными спецификациями задачи (области сложной формы с относительно узкими сечениями в определенных местах), так и с динамикой (зависимость будущего поведения системы только от текущего состояния или небольшой предыстории, разделением по физическим процессам и т.д.).

Рассматривая эти задачи как специальный класс экстремальных задач с возможно тривиальным оптимизируемым функционалом, мы приобретаем возможность использовать методы декомпозиции экстремальных задач при решении задач математического моделирования [8]. При этом возникает, однако, ряд теоретических и практических вычислительных проблем, связанных с неизбежной вырожденностью подзадач, формулируемых в ходе процесса декомпозиции. Исследование этих проблем, модификация алгоритма декомпозиции с учетом этого обстоятельства, программная реализация и вычислительные эксперименты представляют собой предмет настоящей работы.

Оценивая состояние теории и практики методов декомпозиции можно отметить, что особенно полно развита теория декомпозиции задач линейного программирования, где ее применение может существенно уменьшить вычислительные затраты [6], [7]. В алгоритмах типа симплекс-метода для решения задач линейного программирования с n переменными и m ограничениями пессимистические оценки объема вычислений имеют экспоненциальный порядок вида $O(\alpha^{n+m})$ стандартных операций, где $\alpha > 1$. В последние годы большое внимание уделялось развитию несимплексных алгоритмов с полиномиальной сложностью, для которых такие оценки имеют вид $O(n^k)$, где показатель степени обычно порядка 5. Однако, эта оценка предсказывает для практических задач, где $n \approx 10^4$, также весьма значительный объем вычислений. Кроме этого в этих алгоритмах существенным являются и требования к памяти ЭВМ (порядка m^2).

Все вышеперечисленное оправдывает дальнейшую разработку алгоритмов решения задачи задач линейного программирования большого объема, в частности, для задач с ограничениями специального вида, для которых удастся предложить более экономные алгоритмы решения.

В реальных прикладных задачах матрица может иметь нескольких тысяч строк-столбцов. Как правило, они имеют специальную структуру со сравнительно небольшим количеством связывающих переменных или ограничений. В этом случае перспективным направлением развития алгоритмов решения таких задач являются схемы декомпозиции, одному из вариантов которых и посвящена настоящая работа.

1 Постановка задачи

В качестве основной модели структурированной оптимизационной задачи рассмотрим двублочную проблему следующего вида:

$$\min_{z_A, z_B, x} c_A z_A + c_B z_B \quad (1)$$

$$A_A z_A + B_A x \leq d_A \quad (2)$$

$$A_B z_B + B_B x \leq d_B \quad (3)$$

$$z_A \geq 0, z_B \geq 0, x \geq 0, \quad (4)$$

где z_A, z_B – векторы переменных задачи размерностей N_A и N_B соответственно, x – вектор связывающих переменных задачи размерности N_x , c_A, c_B – векторы длины N_A и N_B соответственно, A_A, A_B, B_A, B_B – матрицы соответствующих размерностей, d_A – вектор длины K_A , d_B – вектор длины K_B .

При фиксированном x эта задача распадается на два независимых блока, что и будет в основном использовано для развития специальных алгоритмов.

Без выделения блоков задача (1)-(4) запишется как $\min_z cz$ при $Az \leq d$, $z \geq 0$, где $z = (z_A, z_B, x)$, $c = (c_A, c_B, 0)$, $d = (d_A, d_B)$,

$$A = \begin{pmatrix} A_A & 0 & B_A \\ 0 & A_B & B_B \end{pmatrix},$$

что демонстрирует специфику задачи. Неформальным требованием является небольшое число связывающих переменных, такие задачи хорошо приспособлены для применения принципов декомпозиции.

Одним из источников задач большой размерности являются проблемы вычислительной математической физики. Использование экстремальных принципов в таких задачах имеет давнюю историю и составляет основу как многих теоретических результатов, так и вычислительных подходов. В классическом варианте применение вариационных принципов ведет к формулировке необходимых условий экстремума, что переходит в вычислительную форму задачи, каковой является система уравнений большой размерности. Такая система зачастую имеет ряд особенностей типа ленточной структуры и решается специальными методами [1].

Вместе с тем, такую систему уравнений можно рассматривать как ограничения экстремальной задачи с тривиальной целевой функцией и применить для ее решения развиваемые в математическом программировании подходы. Мы продемонстрируем этот подход на примере метода конечных элементов [3].

Система уравнений метода конечных элементов с точки зрения теории оптимизации является вырожденной экстремальной задачей с тривиальным функционалом, где основная и единственная проблема заключается в удовлетворении ограничений. Тем не менее мы пока рассмотрим общий случай структурированной двублочной задачи с возможно ненулевым функционалом.

Введя функции

$$\begin{aligned} f_A(x) = \min_{z_A} c_A z_A, & \quad f_B(x) = \min_{z_B} c_B z_B, \\ A_A z_A \leq d_A - B_A x & \quad A_A z_B \leq d_B - B_B x \\ z_A \geq 0, x \geq 0 & \quad z_B \geq 0, x \geq 0 \end{aligned} \quad (5)$$

получим эквивалентную (1)-(4) задачу:

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

Как известно, $f_A(x)$ и $f_B(x)$ – выпуклые и кусочно-линейные функции [4]. Введем для них сопряженные функции $h_A(p)$ и $h_B(p)$:

$$h_A(p) = f_A^*(x) = \max_x \{px - f_A(x)\}, \quad h_B(p) = f_B^*(x) = \max_x \{px - f_B(x)\}.$$

Отсюда в частности следует, что

$$\begin{aligned} -h_A(-p) &= \min_{z_A, x} \{c_A z_A + px\}, & -h_B(p) &= \min_{z_B, x} \{c_B z_B - px\}, \\ A_A z_A + B_A x &\leq d_A & A_B z_B + B_B x &\leq d_B \\ z_A &\geq 0, x \geq 0 & z_B &\geq 0, x \geq 0 \end{aligned}$$

то есть вычисление $h_A(-p)$, $h_B(p)$ также сводится к решению задач линейного программирования, но уже по всем переменным с дополнительными стоимостями для связывающих переменных в качестве параметров.

Задачу (6) можно легко переписать в терминах сопряженных функций:

$$\begin{aligned} \min_x \{f_A(x) + f_B(x)\} &= \min_{x^1 - x^2 = 0} \{f_A(x^1) + f_B(x^2)\} = \\ &= \min_{x^1 - x^2 = 0} \max_p \{f_A(x^1) + px^1 - px^2 + f_B(x^2)\} = \\ &= \max_p \{\min_x (f_A(x) + px) + \min_x (f_B(x) - px)\} = \\ &= \max_p \{-\max_x (-px - f_A(x)) - \max_x (px - f_B(x))\} = \\ &= \min_p \{\max_x (-px - f_A(x)) + \max_x (px - f_B(x))\} = \\ &= -\min_p \{h_A(-p) + h_B(p)\}. \end{aligned}$$

Последнюю задачу в этой цепочке равенств

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

будем называть двойственной к (6).

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).

2 Алгоритм двойственных отсечений (АДО)

Следуя [4] сформулируем алгоритм двойственных отсечений (АДО) для задачи (6). Суть этого подхода заключается в замене некоторых функций в (6)-(7) на их аппроксимации, получаемые в процессе вычислений и обмена. В (6) аппроксимируется функция $f_B(x)$, в (7) – сопряженная функция другого блока $h_A(p)$. Эти аппроксимации будем обозначать как $f_B^k(x)$ и $h_A^k(p)$ соответственно. Собственно алгоритм формулируется следующим образом:

Шаг 1 Выбрать начальные приближения x^0 , p^0 .

Шаг 2 Для начального приближения p^0 вычислить $h_B(p^0)$. Определить начальные аппроксимации $f_B^{-1}(x) = -\infty$, $h_A^{-1}(p) = -\infty$. Положить $k = 0$.

Шаг 3 ЕСЛИ текущая точка x^k не оптимальна, то выполнять:

BEGIN

- Решить задачу

$$\begin{aligned} f_B(x^k) = \min_{z_B, x} c_B z_B & \quad (8) \\ A_B z_B + B_B x \leq d_B \\ x = x^k \\ z_B \geq 0, x \geq 0 \end{aligned}$$

и определить $\bar{p}^k \in \partial f_B(x^k)$, $h_B(\bar{p}^k)$,

- Модифицировать прямую аппроксимацию

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

- Решить задачу

$$\min_x \{f_A(x) + f_B^k(x)\} \quad (9)$$

и определить x^{k+1} , $f_A(x^{k+1})$,

- Решить задачу

$$\begin{aligned} -h_A(-p^k) = \min_{z_A, x} (c_A z_A + p^k x) & \quad (10) \\ A_A z_A + B_A x \leq d_A \\ z_A \geq 0, x \geq 0 \end{aligned}$$

и определить $\bar{x}^k \in \partial h_A(-p^k)$, $f_A(\bar{x}^k)$,

- Модифицировать двойственную аппроксимацию

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

- Решить задачу

$$\min_p \{h_A^k(-p) + h_B(p)\} \quad (11)$$

и определить p^{k+1} , $h_B(p^{k+1})$,

- $k = k + 1$
- Перейти к шагу 3.

END

ИНАЧЕ x^k – решение задачи (6).

3 Вычислительные аспекты АДО

Важными вычислительными задачами в АДО являются вычисления субградиентов функций $f_B(\cdot)$, $h_A(\cdot)$ в (8) и (10). Из стандартной теории двойственности линейного программирования [5] легко получить следующее утверждение, которое мы все же докажем в целях самодостаточности работы.

Пусть $f_B(x)$ определена соотношением (5). Тогда для некоторого фиксированного вектора \bar{x} ее можно представить в эквивалентном виде:

$$\begin{aligned} f_B(\bar{x}) = \min_{z_B, x} c_B z_B & \quad (12) \\ A_B z_B + B_B x \leq d_B & \\ x = \bar{x} & \\ z_B \geq 0, x \geq 0. & \end{aligned}$$

Теорема 3.1 Пусть задача (12) разрешима для заданного \bar{x} , и p^* является вектором оптимальных двойственных переменных, соответствующих ограничению $x = \bar{x}$ в задаче (12). Тогда $p^* \in \partial f_B(\bar{x})$, и обратно, любой вектор $p^* \in \partial f_B(\bar{x})$ является вектором оптимальных двойственных переменных, соответствующих ограничению $x = \bar{x}$ в задаче (12).

Доказательство:

Покажем сначала, что если p_B^* является вектором оптимальных двойственных переменных, соответствующих ограничению $x = \bar{x}$ в задаче (12), то $p_B^* \in \partial f_B(\bar{x})$.

В силу теорем двойственности для $\bar{x} \in \text{dom } f_B$

$$\begin{aligned} f_B(\bar{x}) = \max_{u, p_B} du + \bar{x} p_B & \quad (13) \\ u' A_B \leq c & \\ u' B_B + p_B \leq 0 & \\ u \leq 0. & \end{aligned}$$

Тогда

$$f(x) - f(\bar{x}) \geq du^* + x p_B^* - (du^* + \bar{x} p_B^*) = (x - \bar{x}) p_B^*,$$

и, следовательно, $p_B^* \in \partial f_B(\bar{x})$.

Докажем теперь обратное утверждение. Пусть $p_B^* \in \partial f_B(\bar{x})$. Известно, что $p_B^* \in \partial f_B(\bar{x})$ тогда и только тогда, когда

$$f_B(\bar{x}) - \bar{x} p_B^* = -h_B(p_B^*),$$

где h_B – функция, сопряженная для f_B .

По определению

$$\begin{aligned} -h_B(p_B^*) = \min_{z_B, x} c_B z_B - p_B^* x & \\ A_B z_B + B_B x \leq d_B & \\ z_B \geq 0, x \geq 0, & \end{aligned}$$

для которой двойственная задача имеет вид

$$\begin{aligned} -h_B(p_B^*) = \max_u du & \quad (14) \\ u' A_B \leq c & \\ u' B_B \leq -p_B^* & \\ u \leq 0. & \end{aligned}$$

Сравнивая задачи (13) и (14), легко заметить, что при фиксированном p_B^* выполняется $-h_B(p_B^*) = f_B(\bar{x}) - \bar{x}p_B^*$, ограничения этих задач совпадают, следовательно, p_B^* является вектором оптимальных переменных в задаче (13) и, следовательно, вектором двойственных переменных, соответствующих ограничению $x = \bar{x}$ в задаче (12). ■

Вычисление $h_B(\bar{p}^k)$ в алгоритме двойственных отсечений приводит к задаче

$$h_B(\bar{p}^k) = - \min_{z_B, x} (c_B z_B - \bar{p}^k x) = \bar{p}^k \bar{x}^k - c_B \bar{z}_B^k$$

$$A_B z_B + B_B x \leq d_B$$

$$z_B \geq 0, x \geq 0,$$

где (\bar{z}_B^k, \bar{x}^k) – решение задачи (12).

Задача (9) может быть представлена в следующем виде:

$$\min_{z_A, x, v} (c_A z_A + v)$$

$$A_A z_A + B_A x \leq d_A$$

$$v \geq p^i x - h_B(p^i), 0 \leq i \leq k$$

$$v \geq \bar{p}^i x - h_B(\bar{p}^i), 0 \leq i \leq k$$

$$z_A \geq 0, x \geq 0.$$

Из (9) ясно, что если (z_A^k, x^k, v^*) – оптимальное решение задачи, то $f_A(x^k) = c_A z_A^k$.

В силу теоремы 3.1 нахождение $\bar{x}^k \in \partial h_A(-p^k)$ приводит к вычислению оптимального вектора \bar{x}^k в задаче

$$-h_A(-p^k) = \min_{z_A, x} (c_A z_A + p^k x) \tag{15}$$

$$A_A z_A + B_A x \leq d_A$$

$$z_A \geq 0, x \geq 0.$$

По определению функции $f_A(x)$

$$f_A(\bar{x}^k) = \min_{z_A} c_A z_A = \min_{z_A} c_A z_A$$

$$A_A z_A + B_A \bar{x}^k \leq d_A \quad A_A z_A \leq d_A - B_A \bar{x}^k$$

$$z_A \geq 0 \quad z_A \geq 0.$$

Ясно, что если (\bar{z}_A^k, \bar{x}^k) – оптимальное решение задачи (15), то

$$f_A(\bar{x}^k) = -h_A(-\bar{p}^k) - \bar{p}^k \bar{x}^k = c_A \bar{z}_A^k.$$

Решение задачи (11) не позволяет сразу получать переменные z_B и x , поэтому запишем эту задачу в терминах сопряженной переменной x .

$$- \min_p \{h_A^k(-p) + h_B(p)\} = - \min_{p^1+p^2=0} \{h_A^k(p^1) + h_B(p^2)\} =$$

$$\begin{aligned}
 &= - \min_{p^1+p^2=0} \max_x \{h_A^k(p^1) - p^1x - p^2x + h_B(p^2)\} = \\
 &= - \max_x \{ \min_p (h_A^k(p) - px) + \min_p (h_B(p) - px) \} = \\
 &= - \max_x \{ - \max_p (px - h_A^k(p)) - \max_p (px - h_B(p)) \} = \\
 &= \min_x \{ f_B(x) + \max_p (px - h_A^k(p)) \}.
 \end{aligned}$$

Последнюю задачу на максимум представим как

$$\begin{aligned}
 &\max_{p,v} (px - v) \\
 &ev \geq p' \bar{X}^k - \bar{F}_A^k \\
 &ev \geq p' X^k - F_A^k,
 \end{aligned}$$

где $\bar{X}^k = (\bar{x}^0, \dots, \bar{x}^k)$, $X^k = (x^0, \dots, x^k)$, $\bar{F}^k = (f(\bar{x}^0), \dots, f(\bar{x}^k))$, $F^k = (f(x^0), \dots, f(x^k))$.

Перейдя от переменных (p, v) к вектору двойственных переменных λ , получим двойственную задачу вида

$$\begin{aligned}
 &\min_{\lambda} \bar{F}_A^k \lambda \\
 &x - \bar{X}^k \lambda = 0 \\
 &e\lambda = 1, \lambda \geq 0,
 \end{aligned}$$

где $\tilde{X}^k = (\bar{X}^k, X^k)$, $\tilde{F}^k = (\bar{F}^k, F^k)$.

Окончательно получаем эквивалентную для (11) задачу:

$$\min_{z_B, x, \lambda} (c_B z_B + \tilde{F}_A^k \lambda) \tag{16}$$

$$A_B z_B + B_B x \leq d_B \tag{17}$$

$$x - \tilde{X}^k \lambda = 0 \tag{18}$$

$$e\lambda = 1 \tag{19}$$

$$z_B \geq 0, x \geq 0, \lambda \geq 0. \tag{20}$$

Вектор p^k определяется как оптимальные двойственные переменные ко второй группе ограничений задачи. Ясно, что если (z_B^k, x^k, λ^*) – оптимальное решение задачи, то $h_B(p^k) = p^k x^k - c_B z_B^k$, поскольку функционалы сопряженных задач равны с точностью до знака.

Заметим, что вычисление $\bar{p}^k \in \partial f_B(x^k)$ представляет собой частный случай задачи (16)-(20), когда переменная λ_i , соответствующая x^k , равна 1.

В приведенных выше формулировках f_A, f_B являются выпуклыми кусочно-линейными функциями, и при этом можно доказать конечную сходимость алгоритма [4].

Теорема 3.2 Пусть f_A, f_B являются конечными, выпуклыми и кусочно-линейными, все подзадачи АДО являются разрешимыми для любой итерации k . Тогда задача (1)-(4) имеет конечное решение, и для некоторого k

$$\min_x \{f_A(x) + f_B(x)\} = f_A(x^k) + f_B(x^k).$$

Доказательство:

По построению последовательность $\{f_B^k(x)\}$ монотонна и ограничена сверху:

$$f_B^k(x) \leq f_B^{k+1}(x) \leq f_B(x) < +\infty.$$

С другой стороны

$$f_B(x^k) \leq \max\{f_B^k(x^k), f_B(x^k)\} = f_B^{k+1}(x^k).$$

Отсюда $f_B(x^k) = f_B^{k+1}(x^k)$. Тогда

$$f_A(x^k) + f_B(x^k) = f_A(x^k) + f_B^{k+1}(x^k) = \min_x \{f_A(x) + f_B^{k+1}(x)\} \leq \min_x \{f_A(x) + f_B(x)\},$$

откуда следует, что x^k решает задачу $\min_x \{f_A(x) + f_B(x)\}$. ■

В алгоритме мы не конкретизировали условие прекращения его работы. В качестве признака оптимальности полученного решения можно проверять отделяемость приближенного решения (z_A^k, z_B^k, x^k) на k -ом шаге алгоритма вновь добавляемым отсечением.

4 Модификация АДО для задач с нетривиальным носителем

Во многих постановках область изменения связывающих переменных неявно ограничена требованиями допустимости задач вычисления функций f_A, h_B . Особенно явно это проявляется в декомпозиции задач решения систем уравнений, где подзадачи неминуемо оказываются либо пере- либо недоопределенными. Это приводит к неявно заданным нетривиальным ограничениям на связывающие переменные, которые в предыдущей версии алгоритма [4] не учитывались. Этот эффект весьма существен, на практике во время работы алгоритма, особенно на начальных итерациях, вероятно, а фактически неизбежно, возникновение недопустимых подзадач (8), (10). Характер неразрешимости этих задач может быть двойкой природы: допустимое множество задачи может оказаться пустым, либо ее целевая функция не ограничена на допустимом множестве задачи, то есть допустимое множество задачи содержит направление бесконечного убывания целевой функции (для задачи минимизации).

В АДО вычисление $\bar{p}^k \in \partial f_B(x^k)$ может привести к задаче с пустым допустимым множеством, а вычисление $\bar{x}^k \in \partial h_A(-p^k)$ – к задаче, где целевая функция не является ограниченной. Ясно, что в этих случаях становятся неопределенными величины $h_B(\bar{p}^k)$, $f_A(\bar{x}^k)$. С точки зрения теории двойственности случай неограниченной задачи может быть сведен к вопросу о коррекции алгоритма в случае возникновения подзадачи с пустым допустимым множеством на некотором шаге, поэтому рассмотрим сначала первый случай.

Решить проблему неразрешимости подзадачи алгоритма можно добавляя дополнительные отсеечения, которые отделяют область недопустимых аргументов x^k в задаче $\bar{p}^k \in \partial f_B(x^k)$ от допустимых векторов, множество которых описывается условием $f_B(x) < +\infty$, $f_B(x) \neq -\infty$. Такие отсеечения отделяют область недопустимых аргументов x^k от эффективного множества функции $f_B(x)$, которое обозначим как $\text{dom } f_B$.

Определение 4.1 Гиперплоскость $px = \alpha$ назовем отделяющей точку \bar{x} от множества Φ , если $px \leq \alpha \leq p\bar{x}$ для $x \in \Phi$. Такую гиперплоскость будем также называть отсечением.

Множество $\text{dom } f$ описывается с помощью задачи поиска допустимого базиса в задаче линейного программирования.

Стандартный процесс решения задачи линейного программирования включает фазу 1 – поиск допустимого базиса. В канонической постановке при этом решается вспомогательная задача с дополнительными искусственными переменными-невязками, добавляемыми в ограничения задачи, и искусственным функционалом. Модифицируемый функционал обычно представляет собой сумму абсолютных значений невязок, чтобы сохранить линейность задачи. Если оптимум в этой задаче равен нулю, то допустимый базис в этой задаче существует и определяется.

Положим $\|s\|_1 = \sum_i |s_i|$ и запишем задачу поиска допустимого базиса при нахождении $\bar{p}^k \in \partial f_B(x^k)$ в виде

$$\phi(x^k) = \min_{z_B, x, s} \|s\|_1 \geq 0 \quad (21)$$

$$A_B z_B + B_B x \leq d_B \quad (22)$$

$$x + s = x^k \quad (23)$$

$$z_B \geq 0, x \geq 0. \quad (24)$$

Функция $\phi(x)$ конечна и выпукла и $\phi(x) = 0$ для $x \in \text{dom } f_B$. При $q^k \in \partial \phi(x^k)$

$$q^k(x - x^k) \leq \phi(x) - \phi(x^k).$$

Если $x \in \text{dom } f_B$, то

$$q^k x \leq q^k x^k - \phi(x^k). \quad (25)$$

Поскольку $\phi(x^k) \geq 0$, то

$$q^k x \leq q^k x^k - \phi(x^k) \leq q^k x^k$$

для $x \in \text{dom } f_B$. Отсюда следует, что (25) – отсечение, отделяющее x^k от $\text{dom } f_B$. Задача отделения точки x^k от $\text{dom } f_B$, как правило, имеет множество решений. По теореме 3.1 q^k находится как двойственный вектор к ограничению (23) в задаче поиска допустимого базиса. Однако, q^k , построенный как двойственный вектор к ограничению (23) в задаче (21)-(24), обладает специальным свойством оптимальности.

Определение 4.2 Назовем гиперплоскость в определении 4.1 плотным отсечением точки \bar{x} от множества $\text{dom } f$, если дополнительно выполнено

$$\alpha = \min_{x \in \text{dom } f} \alpha_1 = \max_{x \in \text{dom } f} px$$

$$px \leq \alpha_1$$

$$x \in \text{dom } f$$

Рассмотрим отсечение (25), $q^k \in \partial\phi(x^k)$, и

$$\alpha^* = \max_{x \in \text{dom } f_B} q^k x$$

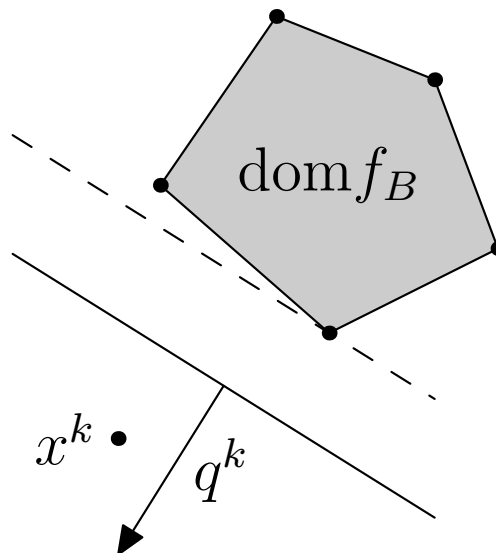


Рис. 1: Отсечения, отделяющие вектор x^k от $\text{dom } f_B$. Пунктирная линия – плотное отсечение.

Теорема 4.1 Отсечение (25) является плотным, т.е. $\alpha^* = q^k x^k - \phi(x^k)$.

Доказательство :

Двойственной задачей для (21)-(24) будет задача линейного программирования вида

$$\begin{aligned} \phi(x^k) = \max_{u,p} & (du + px^k) \\ & u'A_B \leq 0 \\ & u'B_B + p \leq 0 \\ & -e \leq p \leq e \\ & u \leq 0, \end{aligned}$$

где e – вектор из единиц соответствующей размерности.

Обозначая ее решение через (q^k, \tilde{u}) , имеем $\phi(x^k) = q^k x^k + d\tilde{u}$. Так как $q^k \in \partial\phi(x^k)$ и $\phi(x) - \phi(x^k) \geq q^k(x - x^k)$, то для $x \in \text{dom } f_B$ имеем $q^k x \leq q^k x^k - \phi(x^k) = -d\tilde{u}$ или $q^k x + d\tilde{u} \leq 0$.

Для того, чтобы показать плотность данного отсечения, достаточно показать, что

$$\max_{x \in \text{dom } f_B} (q^k x + d\tilde{u}) = 0.$$

Но если $x \in \text{dom } f_B$, то $\phi(x) = 0$, тогда $\max_{x \in \text{dom } f_B} (q^k x + d\tilde{u}) = 0$. ■

Отсечение (25), аппроксимирующее $\text{dom } f_B$, добавляется к ограничениям задачи (9), которые препятствуют возникновению недопустимости на следующих итерациях АДО. После накопления отсечений допустимости областью изменения связанных переменных будет множество, показанное на рис. 2.

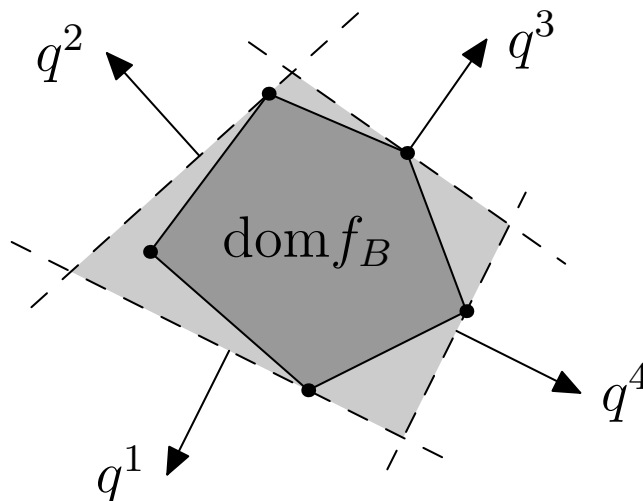


Рис. 2: Плотные отсечения, аппроксимирующие $\text{dom } f_B$.

Вычисление $\bar{x}^k \in \partial h_A(-p^k)$ связано с нахождением оптимального вектора \bar{x}^k в задаче

$$\begin{aligned} -h_A(-p^k) = \min_{z_A, x} & (c_A z_A + p^k x) \\ & A_A z_A + B_A x \leq d_A \\ & z_A \geq 0, x \geq 0. \end{aligned}$$

Эта задача может оказаться неограниченной. Тогда ее двойственная является несовместной, то есть $-p^k \notin \text{dom } h_A$. Этот случай аналогичен недопустимости в прямой задаче (8), и тогда необходимо аппроксимировать область $\text{dom } h_A$ с тем, чтобы дальнейшие приближения не выходили из этого множества.

Для двойственных переменных гиперплоскость, отделяющая вектор $-p^k$ от множества $\text{dom } h_A$, будет задаваться опорным вектором прямых переменных \tilde{x}^k . Этот вектор является решением задачи

$$\begin{aligned} \min_{z_A, x} (c_A z_A + p^k x) \\ A_A z_A + B_A x \leq 0 \\ 0 \leq z_A \leq e, \quad 0 \leq x \leq e, \end{aligned}$$

где e – вектор из единиц соответствующей размерности.

Однако, в данной ситуации возможен более простой метод построения требуемого отсечения. Неограниченность задачи (15) означает существование вектора a , что для любого допустимого вектора x_0 и скаляра $\theta \rightarrow +\infty$, $x_0 + \theta a \rightarrow -\infty$. Тогда отделение недопустимых векторов p достигается с помощью отсечения $ap \leq 0$, которое должно быть добавлено к ограничениям задачи (11) аналогично дополнительным отсечениям в прямой задаче (9).

5 Модифицированный алгоритм двойственных отсечений (МАДО)

В результате введения дополнительных отсечений допустимости МАДО для задачи (6) приобретает следующий вид:

Шаг 1 Выбрать начальные приближения x^0 , p^0 .

Шаг 2 Для начального приближения p^0 вычислить $h_B(p^0)$. Определить начальные аппроксимации $f_B^{-1}(x) = -\infty$, $h_A^{-1}(p) = -\infty$. Положить $k = 0$, $m = 0$, $n = 0$ и ввести множества $X_d^m = \emptyset$, $P_d^n = \emptyset$.

Шаг 3 ЕСЛИ текущая точка x^k не оптимальна, то выполнять:

- Решить задачу

$$\begin{aligned} \phi(x^k) = \min_{z_B, x, s} \|s\|_1 \\ A_B z_B + B_B x \leq d_B \\ x + s = x^k \\ z_B \geq 0, \quad x \geq 0 \end{aligned} \quad (BX0)$$

ЕСЛИ $\|s\|_1 = 0$, то решить задачу

$$f_B(x^k) = \min_{z_B, x} c_B z_B \tag{BX}$$

$$A_B z_B + B_B x \leq d_B$$

$$x = x^k$$

$$z_B \geq 0, x \geq 0$$

и определить $\bar{p}^k \in \partial f_B(x^k)$, $h_B(\bar{p}^k)$,

ИНАЧЕ

определить p_d^m , $h_d^m = p_d^m x^k - \phi(x^k)$ и положить множество допустимости

$$X_d^m = X_d^{m-1} \cup \{x : p_d^m x \leq h_d^m\}, m = m + 1$$

- Модифицировать прямую аппроксимацию

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

- Решить задачу

$$\min_{x \in X_d^{m-1}} \{f_A(x) + f_B^k(x)\} \tag{AX}$$

и определить x^{k+1} , $f_A(x^{k+1})$,

- ЕСЛИ задача

$$-h_A(-p^k) = \min_{z_A, x} (c_A z_A + p^k x) \tag{AP}$$

$$A_A z_A + B_A x \leq d_A$$

$$z_A \geq 0, x \geq 0$$

неограничена, то определить отсечение допустимости вида $x_d^n p \leq 0$ и множество допустимости $P_d^n = P_d^{n-1} \cup \{p : x_d^n p \leq 0\}$, $n = n + 1$,

ИНАЧЕ определить $\bar{x}^k \in \partial h_A(-p^k)$, $f_A(\bar{x}^k)$,

- Модифицировать двойственную аппроксимацию

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

- Решить задачу

$$\min_{p \in P_d^{n-1}} \{h_A^k(-p) + h_B(p)\} \tag{BP}$$

и определить p^{k+1} , $h_B(p^{k+1})$,

- $k = k + 1$, перейти к шагу 3.

ИНАЧЕ x^k – решение задачи (6).

Сходимость этого модифицированного алгоритма может быть доказана аналогично теореме 3.2. По сути дела этот алгоритм обладает тем же свойством монотонности аппроксимирующей функции f_B^k по итерациям алгоритма, что представляет собой основу доказательства сходимости АДО. Если понимать f_B^k как частично определенную функцию вида

$$f_B^k(x) = \begin{cases} < +\infty, & x \in X_k, \\ = +\infty, & x \notin X_k, \end{cases}$$

то отсечения дополнительности гарантируют и выполнение условия $f_B^k(x^k) = f_B^{k+1}(x^k)$ в том смысле, что если $x^k \notin \text{dom } f_B$, то $x^k \notin X_{k+1}$.

6 Тестовый пример

Хорошим источником большеразмерных систем линейных уравнений являются задачи математической физики. Эти большие разреженные системы линейных уравнений являются и подходящим тестовым примером для проверки эффективности применения МАДО. Помимо их вычислительной важности такие задачи представляют собой и практический интерес.

При декомпозиции исходной задачи на подзадачи решение системы будет единственной точкой пересечения нетривиальных допустимых областей подзадач. В этом смысле такие задачи являются хорошими тестами для МАДО. Оптимизируемый функционал в таких задачах несущественен и может быть положен равным нулю.

В качестве примера рассмотрена задача нахождения стационарного распределения температур в некоторой компактной двумерной области D с известными условиями на границе Γ . Функция распределения температур $T(x, y)$ является решением уравнения Лапласа

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, \quad (x, y) \in D \quad (26)$$

с граничными условиями Дирихле на границе,

$$T(x, y) = t(x, y), \quad (x, y) \in \Gamma. \quad (27)$$

Известно, что решение $T(x, y)$, удовлетворяющее уравнениям (26)-(27), совпадает с функцией, минимизирующей функционал

$$\chi = \frac{1}{2} \iint_D \left[\left(\frac{\partial^2 \hat{T}}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \hat{T}}{\partial y^2} \right)^2 \right] dx dy, \quad (28)$$

где $\hat{T}(x, y)$ – функция из допустимого множества пробных функций, заданных в D . Для этой задачи пробные функции допустимы, если они непрерывны и имеют кусочно-непрерывные первые производные. Кроме того пробные функции должны удовлетворять граничным условиям (27).

Классический подход применения метода конечных элементов к решению задачи (28) [3] приводит к решению линейной системы

$$K\bar{T} = R, \quad (29)$$

где K – матрица жесткости системы, \bar{T} – искомый узловый вектор, R – вектор правой части системы. Детали вычисления соответствующих матриц можно найти, например, в [3].

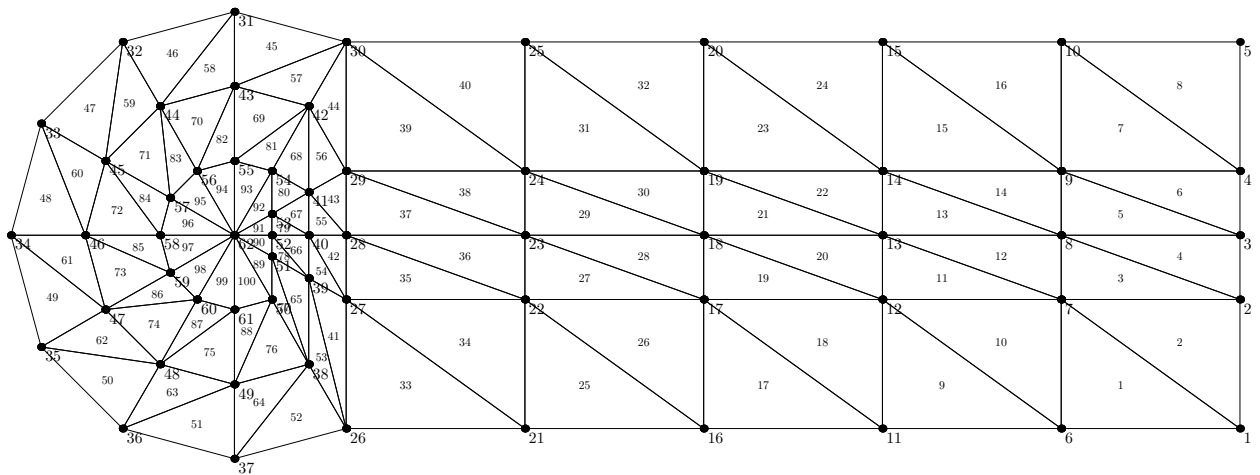


Рис. 3: Разбиение области на треугольные элементы

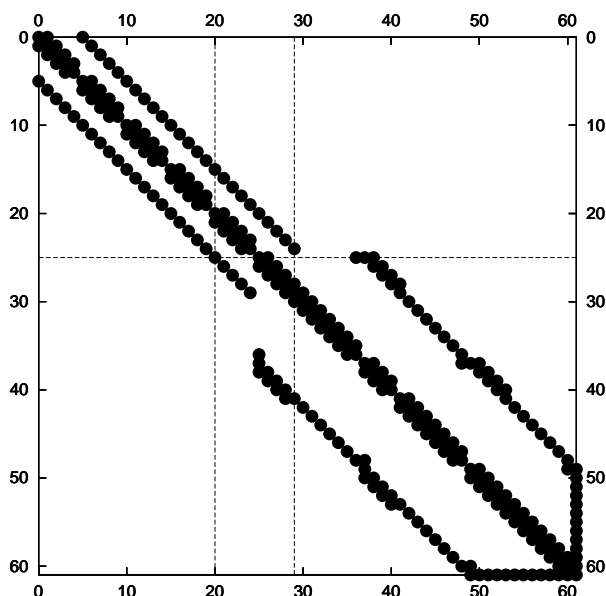
Для достижения высокой точности решения необходимо использовать большое количество элементов, что соответственно повышает размерность матрицы жесткости системы K . Последнее время широко исследуются подходы, связанные с эффективным предобуславливанием системы линейных уравнений с учетом ее блочной структуры [2], [9], [11], [12].

В качестве тестовой решалась задача нахождения стационарного распределения температур в двумерной области, показанной на рис. 3. Использовались кусочно-линейные аппроксимации поля температуры в треугольных трехузловых лагранжевых элементах. На правой части границы данной области, в узлах 1-5, задано условие Дирихле $T_0 = 50$, на остальной части границы выполнено граничное условие Неймана $\frac{\partial T}{\partial n} = 0$, где n – нормаль к границе. Известно, что граничные условия Неймана выполнены автоматически для функции, минимизирующей функционал (28), как естественное следствие вариационной формулировки. Точное решение задачи очевидно – температура во всех узлах области $T = T_0 = 50$.

Количество узлов системы равно 62, область разбита на 100 элементов. “Портрет” матрицы жесткости системы с распределением ненулевых элементов приведен на рис. 4. Блочное представление матрицы жесткости системы очевидно. Пунктирной линией показаны границы подматриц A_A , B_A , B_A , B_B . На рис. 5 показана матрица жесткости системы с разбиением на блоки. Типичные характеристики подзадач приведены в таблице 1.

7 Результаты вычислительного эксперимента

Модифицированный алгоритм двойственных отсечений реализован на языке C с использованием библиотеки функций линейной алгебры `meschach` [10]. Расчет производился

Рис. 4: "Портрет" матрицы жесткости K

	Блок А	Блок В
Количество внутренних переменных z	20	32
Количество связывающих переменных x	10	10
Количество ограничений	25	37
Количество граничных условий	5	0

Таблица 1: Характеристика подзадач тестового примера

на ПК типа PIII/850 под ОС Linux с использованием компилятора gcc. Текст программы (не считая meschach) составляет около 2000 операторов. Эта реализация может рассматриваться в качестве предварительной, в дальнейшем будет реализован параллелизованный вариант МАДО.

Статистика решения тестового примера приведена в табл. 2. Результаты решения задачи показывают достаточно быструю сходимость по "большим" итерациям, оптимальное решение устанавливается уже после $k = 5$, это говорит о небольшом обмене данными между задачами (АХ) и (ВР), что способствует разработке параллелизованного алгоритма. На итерациях 1-5 задача ВР возвращает опорные векторы p_d^m отсечений допустимости, на 6-ой итерации алгоритма вектор p^k становится нулевым. Для решения подзадач МАДО использовался симплекс-метод с определением двойственных переменных [6]. Суммарное количество итераций симплекс-метода при решении подзадач (АХ) составило 265, в под-

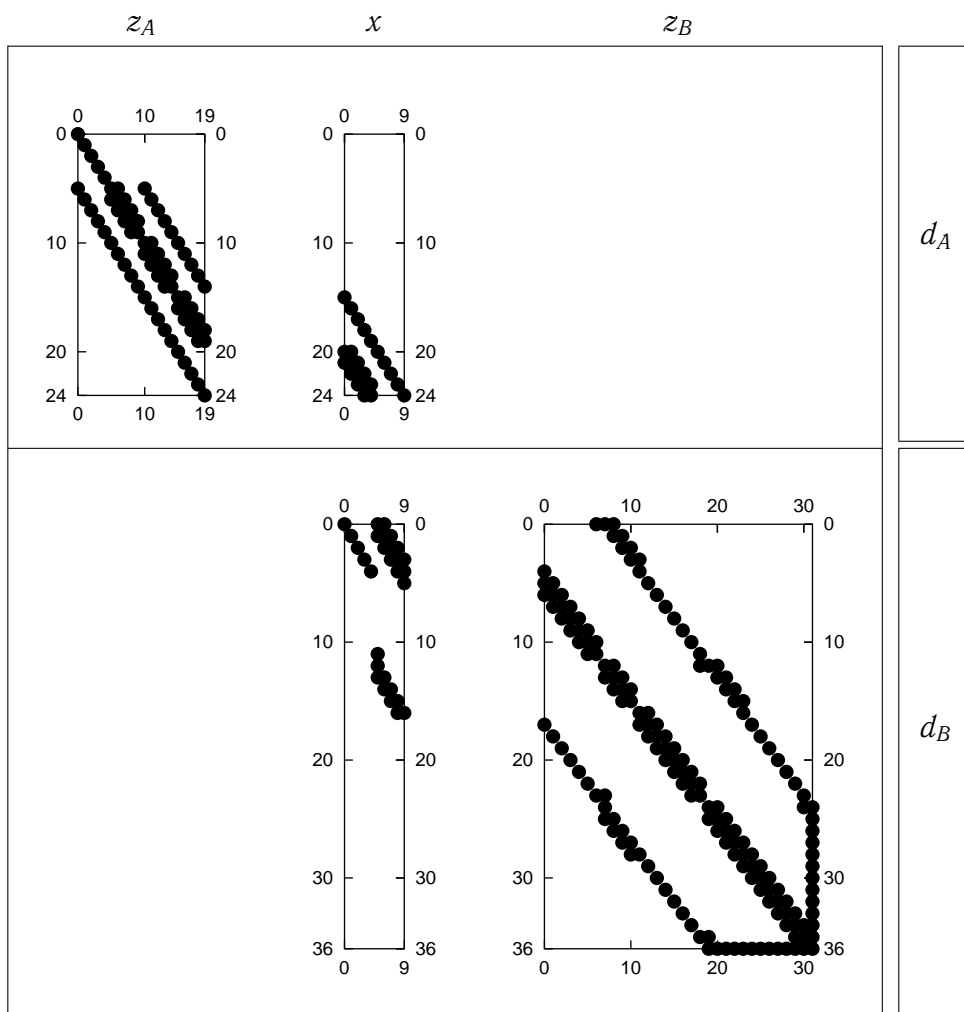


Рис. 5: “Портрет” матрицы жесткости с разбиением на блоки A_A, B_A, B_B, A_B

задачах (BP) – 162.

Данная задача является примером проблемы поиска допустимой точки, поэтому для нее можно вести дополнительные улучшения в виде эффективных рестартов. Простейшая процедура рестарта основана на специфике подзадач.

Задача (AX) на $(k + 1)$ -ом шаге отличается от задачи (AX) на k -ом шаге наличием дополнительных отсечений. Соответственно двойственная постановка задачи (AX) на $(k + 1)$ -ом шаге по сравнению с k -ым шагом содержит дополнительные переменные. Отсюда следует, что оптимальное решение двойственной для (AX) задачи на k -ом шаге является допустимым для этой же задачи на $(k + 1)$ -ом шаге. Использование формулировки задачи (BP) в двойственной постановке позволяет использовать ее оптимальное решение на k -ом шаге в качестве допустимого для этой же задачи на $(k + 1)$ -ом шаге, полагая добавленные дополнительные переменные нулевыми.

Применение такой стратегии рестарта при решении подзадач позволяет уменьшить суммарное количество “малых” итераций симплекс-метода в подзадаче (AX) до 45, в подзадаче

(BP) – до 26.

Насколько позволяет судить наш пока весьма ограниченный вычислительный опыт, алгоритм достаточно хорошо масштабируется при увеличении размеров задачи. Решались задачи и большей размерности с количеством связывающих переменных равным 50 и 102. Количество "больших" итераций сохраняется порядка числа связывающих переменных. Вместе с тем существует проблема повышения эффективности при решении подзадач, доминируют затраты на решение задач (BX). Возможно, что трудоемкость данной задачи может быть уменьшена при использовании проекционных алгоритмов для решения подзадач [4],[13].

k	тип	p^k, p_d^m					тип	x^k				
0	BP	0.36	0.54	0.36	0.54	0.36	AX	52.92	66.85	50.59	40.94	34.80
		-0.36	-0.54	-0.36	-0.54	-0.36		0.00	200.00	0.00	0.00	0.00
1	BP	0.36	0.54	0.36	0.54	0.36	AX	78.97	58.87	45.47	37.45	32.29
		-0.36	-0.54	-1.00	-1.00	-1.00		178.11	82.26	0.00	0.00	0.00
2	BP	0.36	0.54	0.36	0.54	0.36	AX	48.33	48.05	49.26	56.76	45.85
		-0.36	-0.54	-1.00	-0.54	-1.00		48.18	39.37	0.00	129.07	0.00
3	BP	0.36	0.54	0.36	0.54	0.36	AX	50.32	53.42	53.50	49.99	41.13
		-0.37	-0.53	-0.37	-0.54	-1.00		38.16	63.65	84.07	54.92	0.00
4	BP	0.36	0.54	0.36	0.54	0.36	AX	49.42	49.98	49.86	50.27	50.63
		-0.37	-0.52	-0.37	-0.54	-0.36		46.80	52.01	45.62	51.66	52.44
5	BP	0.04	-0.12	-0.01	-0.08	0.17	AX	50.00	50.00	50.00	50.00	50.00
		-0.45	1.00	-0.55	1.00	-1.00		50.00	50.00	50.00	50.00	50.00
6	BP	0.00	0.00	0.00	0.00	0.00	AX	50.00	50.00	50.00	50.00	50.00
		0.00	0.00	0.00	0.00	0.00		50.00	50.00	50.00	50.00	50.00

Таблица 2: Статистика решения тестового примера. k – номер итерации алгоритма двойственных отсечений, x^k, p^k – значения векторов связывающих и сопряженных переменных на k -ой итерации, p_d^m – опорный вектор отсечения допустимости. Столбцы "тип" описывают типы решаемых подзадач

Список литературы

- [1] Гловински Р., Лионс Ж.-Л., Трёмольер Р. Численное исследование вариационных неравенств.–М.: Мир, 1979.
- [2] Корнеев В.Г. Эффективное преобуславливание методом декомпозиции областей.– Известия вузов, Математика, 1999, No. 5, 11.

- [3] Норри Д., де Фриз Ж. Введение в метод конечных элементов.–М.: Мир, 1981.
- [4] Нурминский Е.А. Численные методы выпуклой оптимизации.–М.: Наука, 1991.
- [5] Пшеничный Б.Н. Выпуклый анализ и экстремальные задачи.–М.: Наука, 1980.
- [6] Танаев В.С. Декомпозиция и агрегирование в задачах математического программирования.–Мн.: Наука и техника, 1987.
- [7] Цурков В.И. Декомпозиция в задачах большой размерности.–М.: Наука, 1981.
- [8] S. Elhedhli, J.-L. Goffin, J.-P. Vial Cutting plane methods for nondifferentiable optimization. in "Encyclopedia on Optimization" C. Floudas, P. Pardalos, editors, Kluwer Academic Publishers, 1999. ftp://ftp.crt.umontreal.ca/pub/users/jlg/ElGoVi2_3.ps.gz
- [9] J. Mandel Balancing domain decomposition, 1992. <http://www.mgnet.org/mgnet/papers/Mandel/bdd.ps.gz>
- [10] Meschach library version 1.2b <ftp://thrain.anu.edu.au/pub/meschach/>
- [11] S.V. Nepomnyaschikh, Method of splitting into subspaces for solving elliptic boundary value problems in complex-form domains. Russian (Sov) J. Numerical Anal. and Math. Model., 1991, Vol. 6, No. 2, 151-168.
- [12] J.-P. Shao Domain decomposition algorithms, University of California, Los Angeles, PhD thesis, 1993. <http://www.mgnet.org/mgnet/papers/Shao/jpshao1.ps.gz>
- [13] J. Gondzio, R. Sarkissian and J.-P. Vial Using an interior point method for a master problem in a decomposition approach iterative algorithms. Logilab, HEC, Section of Management Studies, University of Geneva, Technical Report 1995.30, October 24, 1995, revised May 15, 1996. <http://ecolu-info.unige.ch/logilab/sarkissian/pointers/master.ps>