ДАЛЬНЕВОСТОЧНЫЙ МАТЕМАТИЧЕСКИЙ ЖУРНАЛ. 2012. Том 12. № 2. С. 160–170

УДК 519.852.67, 519.853.6
MSC2010 90C05, 90C06, 90C52

© А. С. Величко[1]

О выборе шага в проективных алгоритмах для задач линейного программирования большой размерности

Для решения задач линейного программирования большой размерности рассматриваются алгоритмы, использующие операцию проекции точки на множество. Предложен специальный способ выбора начального приближения и шаговых множителей для сокращения объема вычислений. Для специальной тестовой задачи со случайно генерируемыми данными выполнен сравнительный анализ времени работы и скорости сходимости алгоритма при различном выборе шаговых множителей.

Ключевые слова: *условная оптимизация, линейное программирование, задача большой размерности, численный метод, проективный алгоритм.*

Введение

Пристальный интерес к возможностям использования проекции на множество для решения задач математического программирования появился с 60-х годов XX века в работах советских математиков И.И. Еремина [1, 2, 3], Л.М. Брэгмана [4], Б.Т. Поляка [5], Е.Г. Гольштейна [6]. В зарубежной литературе развиваются вычислительные алгоритмы для нахождения точки, принадлежащей пересечению выпуклых множеств (convex feasibility problem) [7, 8].

В работах [9, 10] авторы развивают аппарат итерационных процессов фейеровского типа с оператором проекции. Этот подход используется для разработки методов решения систем линейных уравнений и неравенств, соответствующих необходимым и достаточным условиям экстремума для задач линейного программирования. В статьях Е.А. Нурминского [11, 12] развит подход фейеровских алгоритмов с убывающим возмущением, применяемых непосредственно к задачам условной оптимизации. В этом случае преобразование исходной оптимизационной задачи к системе линейных неравенств не используется. В работах Мишело [13] и Нурминского [14] для решения задач оптимизации предлагалось использовать специальные методы проекций на внешне заданные полиэдры.

Совершенствование численных методов оптимизации является актуальным в случае большого числа ограничений и/или переменных. С одной стороны, для задач линейного программирования классические алгоритмы типа симплекс-метода имеют неполиномиальную оценку сложности. С другой стороны, алгоритмы с полиномиальной вычислительной сложностью и линейной оценкой скорости сходимости для задач большой размерности могут демонстрировать очень медленную сходимость по числу итераций и по времени работы

[1]Институт автоматики и процессов управления ДВО РАН, 690041, Владивосток, ул. Радио, 5.
Электронная почта: vandre@dvo.ru

алгоритма. Класс задач линейного программирования с ограничениями в виде линейных неравенств является важным объектом для разработки эффективных численных методов, поскольку зачастую в алгоритмах для нелинейных или недифференцируемых оптимизационных задач требуется решать вспомогательные задачи линейного программирования. Проективные алгоритмы привлекают внимание специалистов по численным методам оптимизации еще и в силу того, что в этом классе алгоритмов возможно организовать параллельные вычисления.

В данной работе рассматриваются проективные алгоритмы с произвольным возмущением, в которых не налагается условий на убывание шаговых множителей. Для задач линейного программирования с ограничениями типа неравенств показан способ выбора начального приближения и шаговых множителей для сокращения объема вычислений, под которым понимается время работы алгоритма.

1. Фейеровские и проективные алгоритмы для экстремальных задач

Рассмотрим экстремальную задачу

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

где $h(\cdot)$ – выпуклая функция на множестве $V = \bigcap\limits_{i\in I} C_i, I = \overline{1,m}$, которое представляет собой пересечение конечного числа выпуклых замкнутых множеств C_i. Будем в дальнейшем считать, что $h(\cdot) \not\equiv -\infty, V \neq \emptyset$, и рассматриваемая задача (1) имеет решение.

Для решения экстремальной задачи (1) в работах [11, 12] был обоснован класс фейеровских алгоритмов с убывающим возмущением и сильным аттрактантом. В данных работах рассматривалась итерационная последовательность $x^{s+1} = \mathcal{F}_V(x^s + \lambda_s\Phi(x^s)), s = 0, 1, \ldots,$ где $\mathcal{F}_V$ – локально сильно фейеровский оператор относительно множества V, Φ – ограниченный сильный аттрактант ([11], теорема 2). Для обеспечения сходимости алгоритма на последовательность шаговых множителей $\lambda_s > 0$ налагаются условия убывания $\lambda_s \to 0, \sum\limits_{s=0}^{+\infty} \lambda_s = +\infty$, которые обычно используются в классе субградиентных алгоритмов. Такие условия порождают уменьшение возмущений $\lambda_s\Phi(x^s)$ в ходе работы алгоритма. Сходимость в данном классе алгоритмов доказана в предположении о произвольном начальном приближении x^0 [11, 12].

В том случае, когда в качестве фейеровского оператора $\mathcal{F}_V$ используется оператор P_V проекции на множество V, возникает семейство проективных алгоритмов. В предположении конечности и дифференцируемости функции $h(\cdot)$ на множестве V в работах [11, 12] предлагается выбирать в качестве значений ограниченного сильного аттрактанта $\Phi(\cdot)$ вектор $-g^s$ антиградиента функции $h(\cdot)$. При таком выборе $\mathcal{F}_V(\cdot)$ и $\Phi(\cdot)$ вектор $x^{s+1} = P_V(x^s - \lambda_s g^s)$ определяется в результате решения задачи квадратичного программирования с ограничениями $\min\limits_{x\in V} \|x - x^s + \lambda_s g^s\|^2$. Таким образом, рассматриваемый проективный алгоритм является частным случаем в классе фейеровских алгоритмов с убывающим возмущением и ограниченным сильным аттрактантом.

Метод $x^{s+1} = P_V(x^s - \lambda_s g^s)$ известен в литературе как метод проекции градиента, применяемый для решения задачи (1), однако ранее он был обоснован для шаговых множителей, зависящих от константы Липшица для градиента функции $h(\cdot)$ [15].

Задачи линейного программирования могут рассматриваться как частный случай задачи (1), когда $h(x) = cx$, градиент g^s функции $h(\cdot)$ на допустимом множестве V равен

вектору c, множества $C_i = \{a^i x \le b_i\}$ представляют собой линейные полупространства, то есть задача (1) принимает вид

$$\min_x \{cx : a^i x \le b_i, i = \overline{1,m}\}. \tag{2}$$

Зачастую в численных алгоритмах решения нелинейных или недифференцируемых задач возникают вспомогательные задачи линейного программирования.

Например, нелинейную задачу (1) можно представить в виде задачи с линейной целевой функцией

$$\min_{x \in V, u} u, h(x) \le u,$$

и для выпуклого ограничения $h(x) \le u$ построить набор внешних линейных аппроксимаций в пространстве переменных (x, u). Таким образом, решение нелинейной задачи сводится к последовательности задач линейного программирования и задач проекции на множество, что используется в методах типа метода Келли и методе уровней [16].

Задачи недифференцируемой оптимизации с минимизацией гельдеровой L_1 нормы в евклидовом пространстве – задачи регуляризации по Тихонову с недифференцируемым стабилизирующим функционалом, поиск универсальных решений задач интервальной математики, робастное оценивание в задачах аппроксимации – сводятся к задачам линейного программирования.

Например, применение метода регуляризации Тихонова к одномерному интегральному уравнению Фредгольма $\int\limits_a^b K(t,s)u(s)\,ds = y(t)$ приводит к задаче безусловной недифференцируемой оптимизации

$$\min_{\{u_j\}} \Big\{ \sum_{i=1}^{m} h \Big| \sum_{j=1}^{n} hK(t_i, s_j)u_j - y_i \Big| + \alpha \sum_{j=1}^{n-1} |u_{j+1} - u_j| \Big\},$$

где h – шаг сетки, α – параметр регуляризации. Эту задачу можно эквивалентно представить в виде задачи линейного программирования с ограничениями в виде неравенств. Для этого необходимо ввести новые переменные $v_j = |u_{j+1} - u_j|$, $w_i = |\sum\limits_{j=1}^{n} hK(t_i, s_j)u_j - y_i|$, добавить ограничения

$$v_j \ge u_{j+1} - u_j, v_j \ge -(u_{j+1} - u_j), w_i \ge \Big(\sum_{j=1}^{n} hK(t_i, s_j)u_j - y_i\Big), w_i \ge -\Big(\sum_{j=1}^{n} hK(t_i, s_j)u_j - y_i\Big)$$

и заменить целевую функцию задачи на $\Big(h \sum\limits_{i=1}^{m} w_i + \alpha \sum\limits_{j=1}^{n-1} v_j\Big)$.

2. Стандартные способы выбора шаговых множителей

Выбор определенной последовательности шаговых множителей λ_s в конкретном классе задач может повлиять на скорость сходимости проективных алгоритмов, применяемых для решения задачи (1). В качестве указанных в п.1 условий, требуемых для обеспечения сходимости в классе фейеровских алгоритмов, в литературе зачастую используется последовательность шагов вида $\lambda_s = u/(v+s), u, v > 0$ [15, 16]. Такой выбор шаговых множителей в дальнейшем будем называть последовательностью "малых шагов".

Рассмотрим оптимальный выбор шага для метода проекций градиента, применяемого для решения задачи (2).

Теорема 1. *При выборе стратегии оптимальных шагов* $\lambda_s = c(x^s - x^*)/\|c\|^2$ *последовательность* $\{x^s\}$, *генерируемая проективным алгоритмом* $x^{s+1} = P_V(x^s - \lambda_s c)$, *сходится к оптимальному решению задачи (2) с линейной скоростью.*

Доказательство. Пусть x^* – решение задачи (1). Рассмотрим итерационный процесс $x^{s+1} = P_V(x^s - \lambda_s c)$, где x^{s+1} – решение задачи $\min\limits_{x \in V} \|x - x^s + \lambda_s c\|^2$.

Обозначим $\bar{x}^s = x^s - \lambda_s c$, получим

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

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

$$\lambda_s = p_s\|x^s - x^*\|/\|c\| = c(x^s - x^*)/\|c\|^2,$$

получим, что

$$\|\bar{x}^s - x^*\| = \theta_s\|x^s - x^*\|, \theta_s = \sqrt{1 - p_s^2}, 0 \le \theta_s \le 1. \tag{3}$$

С учетом свойства локальной сильной фейеровости для оператора проекции P_V, то есть $\|P_V(x) - v\| \le q_s\|x - v\|, 0 \le q_s < 1$ для $v \in V$ и $x \notin V$, и с учетом (3) получим для $v = x^* \in V$ и $\bar{x}^s = x^s - \lambda_s c \notin V$ соотношение

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

что означает линейную скорость сходимости алгоритма. Теорема доказана.

Последовательность шаговых множителей в теореме 1 известна в литературе как шаг Моцкина – Агмона [17], который для задачи (1) имеет вид $\lambda_s = (h(x^s) - h(x^*))/\|g^s\|^2$. Такой выбор λ_s обеспечивает минимальное значение величины $\theta_s^2 = \|\bar{x}^s - x^*\|^2/\|x^s - x^*\|^2$.

Обобщение данного шага в виде $\lambda_s = \gamma(h(x^s) - h(x^*))/\|g^s\|^2, 0 < \gamma < 2$ рассматривалось в работе Б.Т. Поляка [18]. Из доказательства теоремы 1 следует, что такой выбор шагов возможен и для рассматриваемого проективного алгоритма, если величины λ_s выбирать из условия $\theta_s^2 = \|\bar{x}^s - x^*\|^2/\|x^s - x^*\|^2 \le 1$.

Недостатком стандартных способов выбора шагового множителя является необходимость дополнительного определения величин u, v в случае применения стратегии "малых шагов" и необходимость знания величины cx^* при выборе шага Моцкина – Агмона. Произвольный выбор начального приближения может представляться удобным, но в результате оказаться малоэффективным по времени работы алгоритма. В работе Н.З. Шора [19] было предложено, при отсутствии информации об оптимальном значении функции $h(x^*)$ в задаче (1), использовать оценки для $h(x^*)$, получаемые в ходе работы алгоритма.

3. Специальный выбор шаговых множителей

Отсутствие предварительного анализа при определении значений $u, v > 0$ в последовательности "малых шагов" и неудачное расположение начальной точки в рассматриваемом проективном алгоритме может привести к высокому объему вычислений. Поэтому возникает важная практическая задача специального выбора шаговых множителей для сокращения времени работы алгоритма. Анализ следующего утверждения оказывается плодотворным для построения такой последовательности шаговых множителей.

Теорема 2. *При выборе последовательности шаговых множителей $\lambda_s \to +\infty$ проективный алгоритм решает задачу линейного программирования (2).*

Доказательство. Проективный алгоритм принимает вид $x^{s+1} = P_V(x^s - \lambda_s c)$, и вектор x^{s+1} определяется в результате решения задачи квадратичного программирования

$$\min_{x \in V} \|x - x^s + \lambda_s c\|^2 = \min_{x \in V} \Big\{ \|x - x^s\|^2 + \lambda_s^2 \|c\|^2 + 2\lambda_s c(x - x^s) \Big\}.$$

Величина $\lambda_s^2\|c\|^2 - 2\lambda_s c x^s$ не влияет на оптимальное решение x^*, поэтому предварительно умножив целевую функцию на $\frac{1}{2\lambda_s} > 0$, что также не влияет на положение оптимума, получим эквивалентную задачу $\min\limits_{x \in V} \Big\{ \frac{1}{2\lambda_s} \|x - x^s\|^2 + cx \Big\}$. Отсюда сразу видно, что решаемая квадратичная задача при $\lambda_s \to +\infty$ эквивалентна задаче линейного программирования $\min\limits_{x \in V} cx$, совпадающей с (2). Теорема доказана.

Результат теоремы 2, несмотря на очевидность доказательства, является весьма неожиданным, поскольку условие $\lambda_s \to +\infty$ противоречит стратегии "малых" и оптимальных шагов.

В рассматриваемом проективном алгоритме возможна стратегия выбора большого по абсолютной величине постоянного шага $\lambda_s = \lambda \gg 1$. Например, можно использовать шаговый множитель, исходя из оценки расстояния от начального приближения до границы допустимого множества. Поскольку в качестве допустимого множества задачи (2) рассматривается система неравенств, которую в векторно-матричном виде можно представить как $Kx \le b$, тогда можно оценить число λ для начального приближения x^0, исходя из условий $K(x^0 + \lambda e^j) \ge b$, где e^j – орты евклидового пространства, на котором задан вектор x. Для $x^0 = 0$ получим систему неравенств $\lambda k^j \ge b$, где k^j – векторы-столбцы матрицы ограничений K, тогда можно построить оценку $\lambda = \max\limits_j \Big\{ \sum\limits_i b_i / \sum\limits_i k_i^j \Big\}$. Далее будем называть такую последовательность шаговых множителей постоянным шагом.

Несмотря на свою простоту, указанный выбор постоянного шага грубо аппроксимирует расстояние от начального приближения до границы допустимого множества, и такой шаг может оказаться настолько большим, что это приведет к высоким погрешностям вычислений в процессе работы рассматриваемого проективного алгоритма. Можно предложить и другие оценки расстояния до границы допустимого множества.

Из теоремы 2 и свойств оператора проекции следует, что для операции проекции на множество предпочтительно, чтобы проецируемая точка находилась вне допустимого множества и располагалась как можно "дальше" от этого множества. Как следует из теоремы 2, возможно такое расположение начальной точки алгоритма, что оптимальное решение задачи (2) будет получено проекцией на допустимое множество всего за одну итерацию.

Покажем теперь специальный способ выбора шагов, используя дополнительную информацию о задаче (2). Пусть x^s – приближенное решение задачи на s-ом шаге проективного алгоритма. Определим шаговый множитель λ_s так, чтобы точка $x^s - \lambda_s c$, полученная сдвигом в направлении антиградиента $-c$ целевой функции задачи (2), находилась вне допустимого множества, то есть так, чтобы было выполнено неравенство $a^i(x^s - \lambda_s c) > b_i$. Таким образом, условие описываемого выбора вектора $x^s - \lambda_s c$ представляет собой систему неравенств $\lambda_s a^i c < a^i x^s - b_i$.

Будем рассматривать сдвиги точки x^s в направлении антиградиента $-c$ целевой функции задачи (2) относительно тех ограничений с номерами i, для которых точка x^s является недопустимой, то есть выполнено $a^i x^s - b_i < 0$. Поскольку задача (2) является задачей минимизации, то имеет смысл рассматривать только такие векторы a^i, для которых $a^i c < 0$, то есть вектор антиградиента $-c$ имеет острый угол с нормалями гиперплоскостей $a^i x = b_i$, которые представляют собой грани допустимого множества.

Таким образом, получим набор условий $\lambda_s > (a^i x^s - b_i)/(a^i c) > 0$ для таких i, что $a^i x^s - b_i < 0, a^i c < 0$. Следовательно, обеспечить совместный выбор начальной точки и шаговых множителей можно, полагая произвольной начальную точку x^0 и используя последовательность $\tilde{\lambda}_s > \max\limits_i\{(a^i x^s - b_i)/(a^i c)\}$ по всем таким i, что $a^i x^s - b_i < 0, a^i c < 0$. Выбор $\tilde{\lambda}_s$ в качестве шаговых множителей будем называть стратегией "больших" шагов, он показан на рисунке 1.

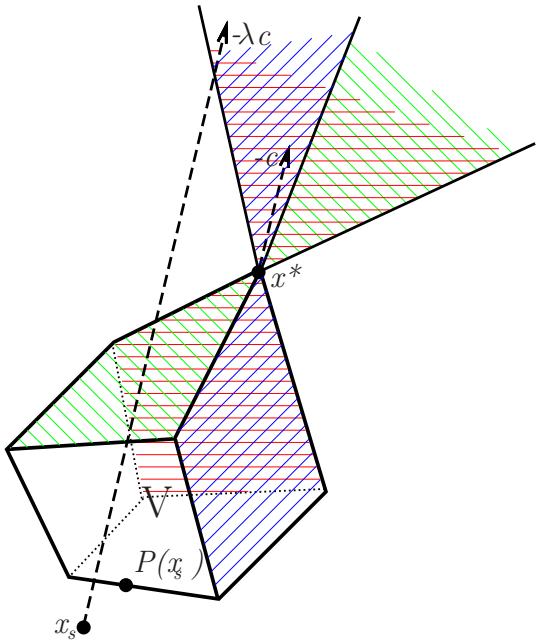

Рис. 1. Стратегия выбора "больших" шагов

Выбор шаговых множителей согласно стратегии "больших" шагов не только решает проблему необходимости привлечения дополнительной информации при использовании стратегии "малых" шагов, шага Моцкина – Агмона, но и позволяет избежать неэффективного выбора начальной точки проективного алгоритма. Рассматриваемая стратегия выбора последовательности шаговых множителей позволяет добиться расположения приближенных решений x^s алгоритма с "подветренной" к экстремуму стороны допустимого множества и избежать его медленного "обхода" при выборе "малых" или оптимальных шагов.

При практическом использовании можно выбирать $\lambda_s = \varepsilon_s + \tilde{\lambda}_s$, где ε_s выбирается из условия "малых" шагов. В случае, если на какой-то итерации алгоритма значение максимума в правиле выбора $\tilde{\lambda}_s$ не существует, можно для этой итерации использовать "малый" шаг. Однако, такой выбор неэффективен, если этот случай повторяется слишком часто.

При приближении x^s к оптимальному решению и в непосредственной близости к допустимому множеству указанная стратегия "больших" шагов становится эффективной, поскольку она предотвращает эффект замедления сходимости при малых значениях шаговых множителей. Однако, если приближенное решение x^s находится относительно далеко от допустимого множества, возможен "малый" шаг, что обеспечит выполнение классических условий сходимости. Поэтому перспективным является "гибридный" подход, при котором "большие" шаги совершаются, когда $\max\limits_i\{a^i x^s - b_i\} \le \delta_s$ для некоторого малого $\delta_s > 0$. Это означает, что хотя бы одно ограничение задачи выполняется "почти" как точное равенство, то есть $a^i x^s = b_i + \delta_s$. По сути это условие определяет, насколько "далеко" приближенное решение x^s расположено от границы допустимого множества.

4. Вычислительные эксперименты

В вычислительных экспериментах использовались случайно генерируемые данные разных размерностей для тестовой задачи, описанной в работе [14]. В этих тестах система неравенств строилась как набор m случайных опорных плоскостей к n-мерной сфере с центром в случайной точке x^0 с нормально распределенными координатами и радиусом $r = \theta||x^0||, \theta \in (0,1)$. Для того чтобы избежать тривиального решения, к набору опорных плоскостей добавлялась одна специальная гиперплоскость, построенная таким образом, чтобы гарантированно строго отделить начало координат от многогранника. Для этого использовалась одна из касательных плоскостей к сфере, проходящая также и через начало координат.

Условие $|h(x^{s+1}) - h(x^s)|/|h(x^s)| \leq 10^{-5}$ используется в качестве критерия завершения алгоритма. При "гибридном" выборе шаговых множителей "большие" шаги совершаются, когда $\delta_s = 10^{-7}$.

Алгоритм реализован [20] с использованием свободно распространяемого программного обеспечения `Octave` (`http://www.octave.org`). Для решения задач проекции на допустимое множество использовался численный алгоритм минимизации квадратичной целевой функции на линейных ограничениях (функция `qp` в `Octave`). Вычислительные эксперименты проводились на ЭВМ в Центре коллективного пользования ДВО РАН "Дальневосточный вычислительный ресурс" (`http://www.cc.dvo.ru`).

На рисунках 2, 3 в логарифмическом масштабе показана практическая вычислительная сложность проективного алгоритма при выборе различных последовательностей шаговых множителей ('small' – 'малый" шаг, 'Motz' – шаг Моцкина – Агмона, 'diam' – постоянный шаг, 'huge' – "гибридная" стратегия). Использовалось два набора случайно генерируемых данных для рассматриваемой тестовой задачи. В первом из них количество генерируемых ограничений существенно превышало число переменных, а именно $m = 2^6 n$ (рисунок 2). Во втором наборе тестов выполнялось соотношение $m = 2n$ (рисунок 3). Результаты вычислительных экспериментов демонстрируют преимущество "гибридной" стратегии и постоянного шага по сравнению с шагами Моцкина – Агмона и "малыми" шагами.

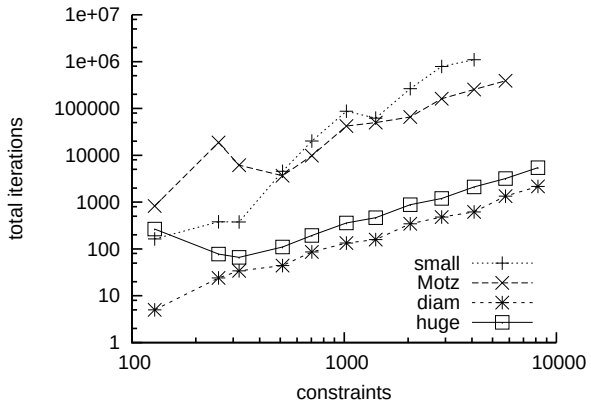

Рис. 2.

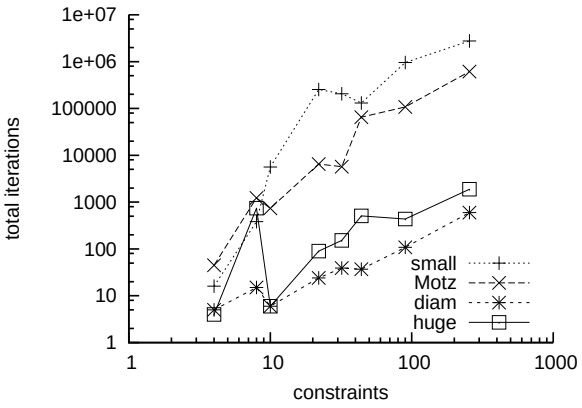

Рис. 3.

В работе [21] был предложен способ визуализации результатов вычислительных экспериментов в виде "профилей производительности", что в настоящее время получило широкое распространение в зарубежной литературе, посвященной численным методам оптимизации. Опишем основные положения данной методики.

Задается множество тестовых наборов данных $P = 1, \ldots, I$ для определенной задачи или множество различных оптимизационных задач с собственным набором тестовых дан-

ных. Для поиска решения задач используется несколько алгоритмов, образующих множество $S = 1, \ldots, J$. В качестве показателя, характеризующего выполнение алгоритмом $s \in S$ тестовой задачи $p \in P$, можно использовать различные величины, например, время работы t_{ps} алгоритма s для тестовой задачи p и достигнутую точность при решении тестовой задачи p алгоритмом s. Определим достигаемую алгоритмом s точность по значению оптимизируемой функции для тестовой задачи p как $\nu_{ps} = |h(\hat{x}_{ps}) - h(x_p^*)|/|h(x_p^*)|$, где $\hat{x}_{ps}$ – полученное алгоритмом s решение тестовой задачи p, и x_p^* – оптимальное решение тестовой задачи p. Введем относительное отклонение решения $\hat{x}_{ps}$, полученного алгоритмом s, от оптимального решения x_p^* как $\mu_{ps} = \|\hat{x}_{ps} - x_p^*\|/\|x_p^*\|$. В простом случае, когда каждая тестовая задача выполняется одним и тем же множеством алгоритмов, можно считать, что величины $T_{ps}, \nu_{ps}, \mu_{ps}$ заданы числовыми матрицами размерности $I \times J$.

Далее для каждой тестовой задачи $p \in P$ рассчитываются показатели эффективности алгоритмов $s \in S$ по формулам $T_{ps} = t_{ps}/\min\limits_{s \in S} t_{ps}$, $V_{ps} = \nu_{ps} - \min\limits_{s \in S} \nu_{ps}$ и $W_{ps} = \mu_{ps} - \min\limits_{s \in S} \mu_{ps}$. Отметим, что для каждого $p \in P$ выполняется $\min\limits_{s} T_{ps} = 1, \min\limits_{s} V_{ps} = 0, \min\limits_{s} W_{ps} = 0$. Затем для показателя эффективности T и каждого алгоритма s определяются функции производительности $\rho_s(t) = N_s(t)/n_s$, где $N_s(t)$ – количество тестовых задач для алгоритма s таких, что $T_{ps} \le t$, n_s – общее число тестовых задач для алгоритма s. Область значений функций $\rho_s(t)$ – отрезок $[0, 1]$. Область определения функций $\rho_s(t)$ – интервал $t \ge 1$. Обозначим через $\rho_s^f(t), \rho_s^x(t)$ функции производительности для показателей эффективности V и W соответственно, область определения этих функций – интервал $t \ge 0$.

В работе [21] графики функций $\rho_s(t)$ на своей области определения названы “профилями производительности” алгоритмов $s \in S$. В рамках рассматриваемой методики чем выше по оси ординат расположен график функции производительности алгоритма, тем более хорошим показателем производительности этот алгоритм обладает по сравнению с остальными.

Далее в рамках описанной методики “профилей производительности” [21] под алгоритмами будем понимать различные способы выбора шаговых множителей в рассматриваемом проективном алгоритме.

На рисунке 4 показаны “профили производительности” $\rho_s(t)$ для времени работы проективного алгоритма (’small’ – ‘малый” шаг, ’Motz’ – шаг Моцкина – Агмона, ’diam’ – постоянный шаг, ’huge’ – “гибридная” стратегия).

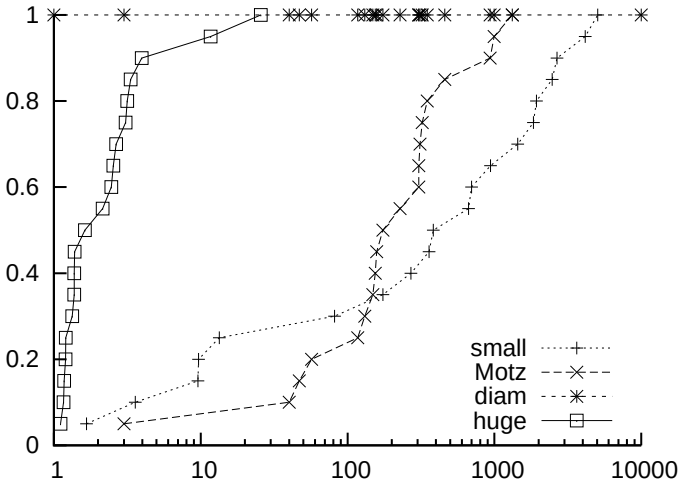

Рис. 4. “Профили производительности” $\rho_s(t)$

На рисунках 5, 6 показаны “профили производительности” $\rho_s^f(t), \rho_s^x(t)$ для достигнутой

точности проективного алгоритма ('small' – 'малый" шаг, 'Motz' – шаг Моцкина – Агмона, 'diam' – постоянный шаг, 'huge' – "гибридная" стратегия). Судя по результатам, представленным на рисунках 4-6, оказывается, что только использование "гибридной" стратегии выбора шага позволяет получить решение, которое, с одной стороны, обеспечивает выигрыш по времени работы алгоритма, и, с другой стороны, является наиболее близким к оптимальному решению в смысле величин ν_{ps} и μ_{ps}. Действительно, проективным алгоритмом с использованием "гибридного" шага достигается высокая точность как по значению оптимизируемой функции, о чем свидетельствуют "профили производительности" $\rho_s^f(t)$, так и по решению оптимизационной задачи, что подтверждается "профилями производительности" $\rho_s^x(t)$.

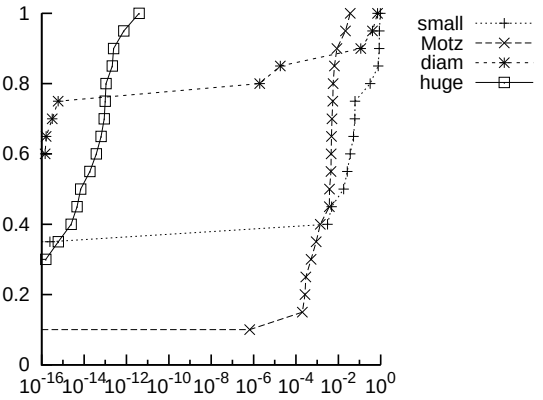

Рис. 5.

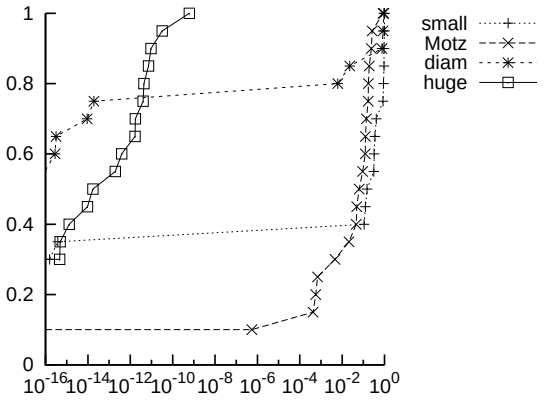

Рис. 6.

На рисунке 7 представлен график сходимости проективного алгоритма при выборе последовательности "малых" шагов (сплошная линия) и шагов Моцкина – Агмона (пунктирная линия). На рисунке 8 показан аналогичный график для "гибридной" стратегии выбора шагов.

Во всех рассматриваемых вычислительных тестах, проективный алгоритм с "гибридной" стратегией выбора шаговых множителей демонстрирует эффект ускорения сходимости на последних шагах алгоритма.

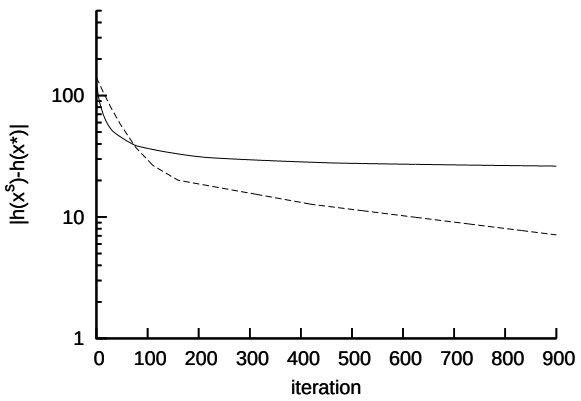

Рис. 7.

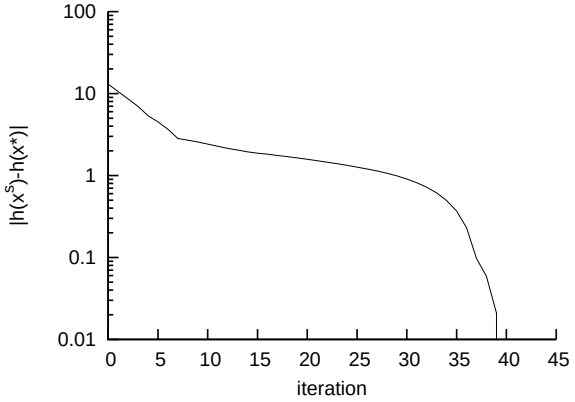

Рис. 8.

Заключение

Эффект ускорения сходимости на последних шагах проективного алгоритма при использовании "гибридной" стратегии для шаговых множителей не противоречит часто наблюдаемому явлению замедления сходимости в алгоритмах градиентного типа. Дело в том, что классические аналитические оценки вычислительной сложности и скорости сходимости алгоритмов получены в рамках концепции "черного ящика" [16], когда применяется "оракул" – вспомогательный алгоритм, возвращающий информацию о градиентах, целевой функции и ограничениях, но не использующий никакой дополнительной информации. В предлагаемом подходе построения шаговых множителей используется информация о структуре ограничений задачи, представленных в виде совокупности линейных неравенств, что позволяет сократить объем вычислений. В работе [14] был получен похожий результат по скорости сходимости проективных алгоритмов: использовался подход описания ограничений исходной задачи в виде выпуклой оболочки точек и специальный метод аффинных подпространств для решения задач проекции.

Полученный результат открывает возможности для разработки эффективных численных алгоритмов решения нелинейных и недифференцируемых оптимизационных задач, использующих вспомогательные задачи линейного программирования.

Недостатком рассматриваемого подхода является то, что стратегия "больших" шагов может давать нулевую оценку шага в ходе работы алгоритма. В этом случае возможно, что на некоторых шагах алгоритма будет использован "малый" шаг, что скажется на росте объема вычислений. Тогда можно перейти к другим стратегиям на основе теоремы 2 или выбрать большой по абсолютной величине постоянный шаг.

Актуальной остается задача разработки эффективных алгоритмов проекции, поскольку в дополнительных расчетах автора при работе с тестовыми задачами библиотеки "NETLIB LP" функция `qp` в `Octave` для решения задач квадратичного программирования не всегда давала оптимальное решение при больших значениях шаговых множителей.

Список литературы

[1] И. И. Еремин, "О некоторых итерационных методах в выпуклом программировании", *Экономика и математические методы*, **2**:6 (1966), 870–886.

[2] И. И. Еремин, В. Л. Мазуров, *Нестационарные процессы математического программирования*, Наука, М., 1979.

[3] И. И. Еремин, "Методы фейеровских приближений в выпуклом программировании", *Математические заметки*, **3**:2 (1968), 217–234.

[4] Л. М. Брэгман, "Релаксационный метод нахождения общей точки выпуклых множеств и его применение для задач выпуклого программирования", *Журн. вычисл. матем. и матем. физики*, **7**:3 (1967), 620–631.

[5] Л. Г. Гурин, Б. Т. Поляк, Э. В. Райк, "Методы проекций для отыскания общей точки выпуклых множеств", *Журн. вычисл. матем. и матем. физики*, **7**:6 (1967), 1211–1228.

[6] Е. Г. Гольштейн, Е. Г. Третьяков, *Модифицированные функции Лагранжа. Теория и методы оптимизации*, Наука, М., 1989.

[7] H. H. Bauschke, J. M. Borwein, "On projection algorithms for solving convex feasibility problems", *SIAM Review*, **38**:3 (1996), 367–426.

[8] Y. Censor, W. Chen, P. L. Combettes, R. Davidi, G. T. Herman, "On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints", *Computational Optimization and Applications*, 2012, № 3, 1065–1088.

[9] Е. А. Бердникова, И. И. Еремин, Л. Д. Попов, "Распределенные фейеровские процессы для систем линейных неравенств и задач линейного программирования", *Автоматика и телемеханика*, 2004, № 2, 16–32.

[10] И. И. Еремин, Л. Д. Попов, "Фейеровские процессы в теории и практике: обзор последних результатов", *Известия ВУЗов. Математика*, 2009, № 1, 44–65.

[11] Е. А. Нурминский, “Использование дополнительных малых воздействий в фейеровских моделях итеративных алгоритмов”, *Журн. вычисл. матем. и матем. физики*, **48**:12 (2008), 2121–2128.

[12] Е. А. Нурминский, “Фейеровские процессы с малыми возмущениями”, *Доклады АН*, **422**:5 (2008), 601–605.

[13] C. Michelot, “A finite algorithm for finding the projection of a point onto the canonical simplex of R^n”, *Journal of Optimization Theory and Applications*, **50**:1 (1986), 195–200.

[14] Е. А. Нурминский, “Проекция на внешне заданные полиэдры”, *Журн. вычисл. матем. и матем. физики*, **48**:3 (2008), 387–396.

[15] Б. Т. Поляк, *Введение в оптимизацию*, Наука, М., 1983.

[16] Ю. Е. Нестеров, *Введение в выпуклую оптимизацию*, МЦНМО, М., 2010.

[17] И. И. Еремин, “Обобщение релаксационного метода Моцкина – Агмона”, *Успехи матем. наук*, **20**:2 (1965), 183–187.

[18] Б. Т. Поляк, “Минимизация негладких функционалов”, *Журн. вычисл. матем. и матем. физики*, **9**:3 (1969), 509–521.

[19] Н. З. Шор, “О скорости сходимости метода обобщенного градиентного спуска с растяжением пространства”, *Кибернетика*, 1970, № 2, 80–85.

[20] А. С. Величко, “Решение задач оптимизации методом последовательных проекций с различными способами выбора начального приближения и шаговых множителей : св. о гос. рег. прог. для ЭВМ Росс. Фед. 2011614018 от 24.05.11 ; заявл. 06.04.11 ; опубл. 20.09.2011. В бюлл.: RU ОБПБТ, 3. С. 319”.

[21] E. D. Dolan, J. J. More, “Benchmarking optimization software with perfomance profiles”, *Mathematical programming*, **91**:2 (2002), 201–213.

Представлено в Дальневосточный математический журнал 23 сентября 2011 г.

Работа выполнена при частичной поддержке грантов РФФИ 09-01-00042-а, ДВО РАН 09-III-А-01-004

Velichko A. S. On the Step Choice in Projection Algorithms for Large-Scale Linear Programming Problems. Far Eastern Mathematical Journal. 2012. V. 12. № 2. P. 160–170.

ABSTRACT

In order to solve large-scale linear programming problems the group of algorithms with projection of a point onto the set is considered. Special method for selecting the initial approximation and the step-size parameters is given to reduce computational time. Comparative analysis of the rates of convergence and running time of the algorithm with various step parameters is done for the special large-scale test problem with randomly generated data.

Key words: *constrained optimization, linear programming, large-scale problem, numerical method, projection algorithm.*