

# Math-Net.Ru

Общероссийский математический портал

И. Г. Черноруцкий, Релаксационные методы жесткой оптимизации,  
*Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление*, 2014, выпуск 4, 69–78

<https://www.mathnet.ru/ntitu50>

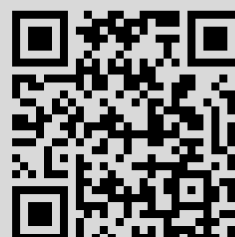
Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<https://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.9.169

18 апреля 2025 г., 06:08:00



## РЕЛАКСАЦИОННЫЕ МЕТОДЫ ЖЕСТКОЙ ОПТИМИЗАЦИИ

I.G. Chernorutskiy

### SEQUENCE RELAXATION METHODS FOR STIFF OPTIMIZATION PROBLEMS

Предложены методы покоординатного спуска (Гаусса–Зейделя) со специальным выбором системы координат, согласованной с собственными векторами локальных матриц Гессе минимизируемого функционала. В отличие от классических покоординатных процедур, построенные методы сохраняют сходимость и эффективность в условиях высокой степени жесткости целевых функционалов.

МЕТОДЫ ПОСЛЕДОВАТЕЛЬНОЙ РЕЛАКСАЦИИ; НЕВЫПУКЛЫЕ ЗАДАЧИ; ЖЕСТКИЕ ФУНКЦИОНАЛЫ.

New sequence relaxation methods are discussed. In contrast to the classical relaxation procedures, the methods retain the convergence and efficiency for non-convex nonlinear programming problems under conditions of high stiffness of target functionals.

SEQUENCE RELAXATION METHODS; NON-CONVEX OPTIMIZATION PROBLEMS; STIFF FUNCTIONALS.

Решается задача безусловной минимизации вида

$$J(x) \rightarrow \min_x, \quad x = (x_1, x_2, \dots, x_n) \in R^n, \\ J \in C^2(R^n)$$

с жестким в смысле определения [1] целевым функционалом. Основные проблемы, связанные с минимизацией жестких функционалов, изложены в [1–8].

Для построения минимизирующей последовательности  $\{x^i\}$ ,  $\lim_{i \rightarrow \infty} J(x^i) \in \text{Arg min}_x J(x)$  рассматривается класс методов последовательной релаксации с использованием различных систем координат и реализацией на каждом шаге спуска вдоль очередного координатного орта.

Например, в соответствии с классическим методом циклического покоординатного спуска Гаусса–Зейделя (ЦПС), использующим естественную систему координат, переход от вектора  $x^i$  к вектору  $x^{i+1}$  происходит следующим образом: для  $l \in [1: n]$  компонент  $x_l^{i+1}$  вектора  $x^{i+1}$  определяется как

$$x_l^{i+1} \in \text{Arg min}_{x \in R^1} J(x_1^{i+1}, x_2^{i+1}, \dots,$$

$$x_{l-1}^{i+1}, x, x_{l+1}^i, \dots, x_n^i).$$

В общем случае при использовании системы координат  $u^1, u^2, \dots, u^n$  вектор  $x^{i+1}$  получается из вектора  $x^i$  следующим образом:

$$x^{i,l} \in \text{Arg min}_{h \in R^1} J(x^{i,l-1} + hu^l), \quad l = 1, \dots, n; \\ x^{i,0} = x^i, \quad x^{i,n} = x^{i+1}.$$

Принципиальная схема рассматриваемого далее класса методов последовательной релаксации – *обобщенного покоординатного спуска* (ОПС) основана на процедуре диагонализации локальной матрицы Гессе  $J''(x)$  на каждом шаге итерационного процесса с последующим циклическим покоординатным спуском вдоль собственных векторов. Целесообразность такого подхода вытекает из геометрически очевидного факта, заключающегося в том, что оси наиболее рациональной системы координат при минимизации квадратичных функционалов (независимо от их выпуклости) методом покоординатного спуска, совпадают с собственными векторами матрицы вторых производных, являющихся осями симметрии

соответствующих поверхностей уровня. Эта идея неоднократно высказывалась в литературе и даже строились соответствующие алгоритмы. Однако численные эксперименты показывали низкую эффективность такого подхода. Существует и объяснение этих неудовлетворительных результатов. Оно основано на том, что при определении собственных векторов, соответствующих близким или кратным собственным значениям, возникают принципиальные вычислительные трудности. Аналогичные трудности, связанные с ограниченной точностью задания исходной информации, а также последующих вычислений, наблюдаются и при диагонализации плохо обусловленных матриц, имеющих относительно малые по модулю спектральные составляющие. Указанные обстоятельства, вероятно, явились основным сдерживающим фактором, не позволившим внедрить обсуждаемые ниже методы в вычислительную практику. Однако неудачи при численном экспериментировании были вызваны, по-видимому, особенностями реализации метода, рассчитанной на случай выпуклых функционалов.

Как показано ниже, для целей оптимизации как в выпуклой ситуации, так и в невыпуклой, достаточно вычислить произвольный ортонормированный базис в инвариантном подпространстве, отвечающем каждой изолированной группе собственных значений. При этом собственные векторы могут быть вычислены со значительными погрешностями, что не снижает эффективность процесса минимизации. А вот отвечающие этим базисам линейные оболочки определяются с высокой точностью и совпадают с истинными подпространствами, определяемыми невозмущенной диагонализируемой матрицей. Как раз это обстоятельство является принципиальным и, как показано ниже, приводит к локальной декомпозиции плохо обусловленной задачи на несколько относительно хорошо обусловленных задач меньшей размерности.

Этот вывод в известной степени подтверждается следующей теоремой.

**Теорема 1.** Пусть  $A$  – симметричная матрица  $n \times n$ ;  $\{u^i\}$  – ортонормированные собственные векторы;  $\lambda_i$  – собственные значения.

Тогда при  $\lambda_i \neq \lambda_j$  с точностью до величин второго порядка малости имеем

$$\langle u^i + du^i, u^j \rangle = (\lambda^i - \lambda^j)^{-1} \langle (dA)u^i, u^j \rangle, \quad (1)$$

где  $dA$  – возмущение матрицы  $A$ ,  $du^i$  – соответствующее возмущение вектора  $u^i$ .

**Доказательство.** Отбрасывая величины второго порядка малости, из равенства  $Au^i = \lambda^i u^i$  получим:

$$(dA)u^i + Adu^i = \lambda_i du^i + d\lambda_i u^i.$$

Отсюда, умножая обе части равенства скалярно на  $u^j$ , имеем

$$\begin{aligned} \langle (dA)u^i, u^j \rangle + \langle Adu^i, u^j \rangle &= \\ &= \langle \lambda_i du^i, u^j \rangle + \langle d\lambda_i u^i, u^j \rangle. \end{aligned}$$

В силу равенств  $\langle u^i, u^j \rangle = 0$ ,  $\langle Adu^i, u^j \rangle = \lambda_j \langle du^i, u^j \rangle$  имеем

$$(\lambda_i - \lambda_j) \langle du^i, u^j \rangle = \langle (dA)u^i, u^j \rangle,$$

откуда следует (1). Теорема доказана.

Пусть теперь

$$M_1 = \sum_{i=1}^{n-r} \alpha_i u^i, \quad M_2 = \sum_{j=n-r+1}^n \alpha_j u^j$$

есть два линейных многообразия, порожденных непересекающимися системами собственных векторов  $\{u^i, i \in [1: n-r]\}$ ,  $\{u^j, j \in [n-r+1: n]\}$  матрицы  $A$ . Если соответствующие множества собственных значений  $\{\lambda^i, i \in [1: n-r]\}$ ,  $\{\lambda^j, j \in [n-r+1: n]\}$  строго разделены в смысле  $|\lambda_i| \gg |\lambda_j|$ , то из (1) следует  $\langle u^i + du^i, u^j \rangle \cong 0$  при достаточно малой величине  $\|dA\|/|\lambda_i|$ . Это означает, что все собственные векторы под действием возмущения  $dA$  изменяются только в пределах своих линейных многообразий, сохраняя с высокой точностью свойство ортогональности к векторам из дополнительных многообразий. При этом сами вариации векторов при близких  $\lambda_i \cong \lambda_j$  собственных значениях в пределах фиксированного линейного многообразия, как это следует из (1), могут быть весьма значительными.

Изложенное выше позволяет в качестве модели программ, реализующих различные методы диагонализации матрицы  $A$ , использовать оператор  $\Lambda(A)$ , ставящий в соответствие произвольной симметричной матрице  $A$  ортогональную матрицу  $V$ , отличную, вообще говоря, от истинной матрицы  $U$ , состоящей из собственных векто-

ров матрицы  $A$ . Оператор  $\Lambda$  характеризуется тем, что если спектр матрицы  $A$  разделяется на  $p$  групп

$$\lambda_i^t(A), i \in [1 : k_t], \sum_{t=1}^p k_t = n$$

«близких» между собой собственных чисел, то каждой группе соответствует набор столбцов  $\{v_{it}\}$  матрицы  $V$ , задающий точное линейное многообразие, порожденное соответствующими столбцами  $\{u^i\}$  точной матрицы  $U$ .

Рассмотрим квадратичную аппроксимацию

$$f(x) = 1/2 \langle Ax, x \rangle - \langle b, x \rangle + c \quad (2)$$

исходного функционала  $J(x)$  в окрестности точки  $x$ . Допустим, что известна матрица  $A$  и ортогональная матрица  $U$ , приводящая ее к диагональному виду  $U^T A U = \text{diag} \lambda_i$ . Тогда замена переменных  $x = Uy$  приводит квадратичный функционал к сепарабельному виду:

$$f(x) = f(Uy) = \sum_{i=1}^n f_i(y_i), \quad (3)$$

где  $f_i$  – квадратичные функции одной переменной (параболы).

Таким образом, достигается *полная локальная декомпозиция* исходной задачи, и последняя сводится к  $n$  независимым экстремальным задачам. В результате поиск оптимального вектора  $u^*$  может осуществляться покомпонентно, ибо связь между аргументами  $y_i$  фактически исчезает. В указанной идеализированной ситуации явление заклинивания метода покоординатного спуска невозможно, и все вычислительные проблемы при применении покоординатных стратегий поиска оптимума, связанные с большими значениями  $n$ , формально снимаются.

В действительности бывает задана не матрица  $A$ , а возмущенная матрица  $A + dA$ , где  $dA$  отражает как неопределенность задания исходной матрицы  $A$ , так и последующие ошибки округления при проведении собственно процесса диагонализации. В связи с этим вместо точной матрицы  $U$  оказывается доступной некоторая матрица  $V = \Lambda(A)$ . Свойства оператора  $\Lambda$  были рас-

смотрены выше. Замена переменных  $x = Uy$  уже не приводит к представлению (3). Для изучения создавшейся ситуации важное значение имеет следующая теорема.

**Теорема 2.** Пусть собственные значения  $\{\lambda_i\}$  и отвечающие им ортонормированные собственные векторы  $\{u_i\}$ ,  $i \in [1 : n]$ , некоторой симметричной матрицы  $A$  разделены произвольным образом на  $p$  групп

$\lambda_i^t, u^i, i \in [1 : k_t], t \in [1 : p], \sum_{t=1}^p k_t = n$  так, что  $u^{it} \neq u^{jt}, t' \neq t'', i \in [1 : k_{t'}], j \in [1 : k_{t''}]$ , где  $\lambda_j^t, u^{jt}$  означают  $j$ -е собственное число и соответствующий собственный вектор группы  $t$ . Тогда, если в каждом линейном многообразии  $M_t$  размерности  $k_t$  с базисом  $\{u^{it}\}$ ,  $i \in [1 : k_t]$  задать иной ортонормированный базис  $\{w^{it}\}$ ,  $i \in [1 : k_t]$ , связанный с исходным базисом линейным соотношением

$$w^{it} = \sum_{m=1}^{k_t} \alpha_{mi}^t u^{mt}, i \in [1 : k_t], \alpha_{mi}^t \in \mathbb{R}^1,$$

то существует такая матрица  $P$  перестановок столбцов, что

1. преобразование подобия

$$\overline{W}^T A \overline{W}, W = \{w^{11}, \dots, w^{k_1 1}, \dots, w^{k_p p}\}, \overline{W} = WP$$

приводит матрицу  $A$  к блочно-диагональному виду

$$\overline{W}^T A \overline{W} = \text{diag}(A_1, A_2, \dots, A_p), \overline{W}^{-T} = \overline{W}^{-1}$$

с квадратными  $(k_t \times k_t)$  матрицами  $A_t$  на главной диагонали;

2. собственные значения матрицы  $A_t$  есть

$$\lambda_i^t, i \in [1 : k_t], t \in [1 : p].$$

**Доказательство.** Первое утверждение проверяется непосредственно с учетом ортонормированности векторов базиса  $\{u^i\}$ . Для доказательства второго утверждения достаточно заметить, что вид и расположение матрицы  $A_m$  при фиксированном многообразии  $M_m$  не зависят от способа задания остальных многообразий  $M_t, t \neq m$ . Поэтому, предположив, что все  $k_t = 1$  при  $t \neq m$ , получим, что все оставшиеся многообразия будут содержать по одному собственному вектору и собственному числу. Тогда будем иметь:

$$\overline{W}^T A \overline{W} = \text{diag}(\lambda_1^1, \lambda_1^2, \dots, A_m, \dots, \lambda_1^p).$$

Учитывая, что преобразование подобия не изменяет спектр матрицы, приходим к требуемому заключению. Теорема доказана.

**Теорема 3.** Пусть  $V = \Lambda(A)$ , тогда

1. замена переменных  $x = Vy$  с точностью до нумерации компонентов вектора  $y$  приводит функционал (2) к блочно-сепарабельному виду

$$f(x) = f(Vy) = f_s(y) = \sum_{i=1}^p f_i(y^i), \quad (4)$$

где

$$y = (y_1, \dots, y_n) = (y^1, \dots, y^p),$$

$$y^i = (y_1^i, \dots, y_{k_i}^i),$$

$$f_i(y^i) \triangleq \frac{1}{2} \langle A_i y^i, y^i \rangle - \langle b^i, y^i \rangle + c_i, \quad c_i \in R^1;$$

2. собственные значения матрицы  $ff^T$  равны

$$\lambda_i^t(A), \quad i \in [1 : k_i], t \in [1 : p].$$

**Доказательство.** Имеем  $V = \overline{W}P$ , где  $P$  – некоторая матрица перестановок столбцов. Поэтому

$$f(x) = \frac{1}{2} \langle V^T A V y, y \rangle - \langle V^T b, y \rangle + c =$$

$$= \frac{1}{2} \langle P^T \overline{W}^T A \overline{W} P y, y \rangle - \langle P^T \overline{W}^T b, y \rangle + c =$$

$$= \frac{1}{2} \langle \overline{W}^T A \overline{W} z, z \rangle - \langle \overline{W}^T b, z \rangle + c, \quad z \triangleq P y.$$

Согласно теореме 2, матрица  $\overline{W}^T A \overline{W}$  имеет блочно-диагональную структуру, что и доказывает первое утверждение. Второе утверждение есть прямое следствие второго утверждения теоремы 2.

**Следствие.** Пусть собственные числа матрицы  $A$  удовлетворяют неравенствам  $\lambda_1 \geq \lambda_{n-r} \gg |\lambda_{n-r+1}| \geq \dots \geq |\lambda_n|$ ; тогда

1. замена переменных  $x = Vy$ ,  $V = \Lambda(A)$ , где  $V = (v^{11}, \dots, v^{1n-r}, v^{21}, \dots, v^{2r})^T$ ,

$$v^{li} = \sum_{m=1}^{n-r} \alpha_{mi}^1 u^m; \quad v^{2i} = \sum_{m=1}^r \alpha_{mi}^2 u^{n-r+m}$$

с точностью до нумерации компонентов вектора  $y$  приводит  $f(x)$  к виду

$$f_s(y) = f_1(y^1) + f_2(y^2), \quad y = (y^1, y^2), \quad (5)$$

где  $y^1 = (y_1, \dots, y_{n-r})$ ,  $y^2 = (y_{n-r+1}, \dots, y_n)$ ;

2.  $\eta_1 \ll \eta$ ,  $\eta_2 \ll \eta$ , где  $\eta_i$  – показатели жесткости задач  $f_i \rightarrow \min$ ,  $i = 1, 2$ .

Таким образом, исходная оптимизационная задача локально может быть сведена к двум эквивалентным задачам с существенно меньшими числами  $\eta_i$ . Представление (5) реализует принцип *частичной локальной декомпозиции* и является аналогом идеализированного соотношения (3).

Если собственные числа матрицы квадратичного функционала разделяются более, чем на две группы, то будет справедливо представление (5), содержащее соответствующее число слагаемых.

Согласно (5) появляется возможность независимого решения не связанных между собой оптимизационных задач для функционалов  $f_i$  с невысокими показателями жесткости.

Полученные результаты носят локальный характер и справедливы в рамках квадратичной аппроксимации исходного функционала  $J(x)$ . Для неквадратичных функционалов приближенное выполнение соотношений типа (5) позволяет говорить о существенном ослаблении связей между различными группами переменных, что определяет достаточно высокую эффективность обобщенного координатного спуска и в общем случае.

Исследование сходимости методов ОПС в предположении точной линейной оптимизации вдоль направляющих ортов может быть основано на общем подходе к исследованию алгоритмов нелинейного программирования [9]. Пусть решается задача  $J(x) \rightarrow \min$ ,  $x \in R^n$ ,  $J(x) \in C^1(R^n)$ . Рассмотрим произвольный алгоритм, строящий последовательность точек  $\{x^k\}$ , причем каждая точка  $x^{k+1}$  получается из  $x^k$  последовательной минимизацией функционала  $J(x)$  вдоль направлений  $d^1, \dots, d^n$ , начиная из точки  $x^k$ . Предполагается, что матрица  $D = (d^1, \dots, d^n)$  может зависеть от номера  $k$ , являясь при любом  $k$  ортогональной.

Легко видеть, что метод ЦПС, метод Розенброка, а также методы ОПС описываются приведенной общей схемой. Можно доказать, что все эти методы при некоторых естественных ограничениях сходятся, т. к. сходится базовый алгоритм.

**Лемма 1.** Пусть  $J(x) \in C^1(R^n)$  и пусть задана последовательность точек  $\{x^k, k = 1, 2, \dots\}$  из  $R^n$  такая, что

1.  $\forall k J(x^{k+1}) \leq J(x^k)$ ;
2.  $J(x^k) \rightarrow J, k \in K$ ,

где  $K$  – некоторое бесконечное подмножество индексов.

Тогда  $\lim_{k \rightarrow \infty} J(x^k) = \lim_{k \in K} J(x^k) = J$ .

Доказательство содержится в [10]. В лемме утверждается, что если последовательность значений функционала  $\{J_k\}$  является монотонно невозрастающей и ее подпоследовательность  $\{J_k, k \in K\}$  сходится к некоторому значению  $J$ , то вся последовательность  $\{J_k\}$ , сходится к этому же значению.

**Теорема 4.** Пусть

1.  $J(x) \in C^1(R^n)$ ;
2. множество решений, определяемое как  $X^* = \{x \in R^n \mid J'(x) = 0\}$  непусто;
3. множество  $\Theta = \{x \in R^n \mid J(x) \leq J(x^0)\}$ , где  $x^0$  – заданная начальная точка, ограничено и замкнуто (компактно) в  $R^n$ ;
4. минимум функционала  $J$  вдоль любой прямой в  $R^n$  единственен;
5. если  $J'(x^k) = 0$ , то алгоритм останавливается в  $x^k$ .

Тогда каждая предельная точка последовательности  $\{x^k\}$ , построенной базовым алгоритмом, принадлежит множеству  $X^*$ .

**Доказательство.** Если последовательность  $\{x^k\}$  конечна, то это значит, что алгоритм остановился в точке  $x^k \in X^*$ , и утверждение теоремы очевидно. Допустим поэтому, что алгоритм порождает бесконечную последовательность точек  $\{x^k\}$ . Согласно третьему предположению все точки расположены в компактном множестве  $\Theta$  и, следовательно, должна существовать подпоследовательность  $\{x^k, k \in K\}$ , сходящаяся при  $k \rightarrow \infty$  к некоторой точке  $x$ . Покажем, что  $x \in X^*$ , то есть  $J'(x) = 0$ .

Рассмотрим подпоследовательность  $\{x^{k+1}, k \in K\}$ . Она также содержится в  $\Theta$ , и поэтому существует  $K' \subset K$  такое, что  $x^{k+1} \rightarrow x'$  при  $k \in K'$ , где  $x'$  – некоторая точка из  $\Theta$ . Из сходимости последовательностей  $\{x^k, k \in K\}$ ,  $\{x^{k+1}, k \in K'\}$  следует в силу непрерывности  $J$  сходимости последовательностей  $\{J(x^k), k \in K\}$ ,  $\{J(x^{k+1}), k \in K'\}$ , соот-

ветственно, к значениям  $J(x)$ ,  $J(x')$ . В силу леммы 4.1 вся последовательность  $\{J(x^k)\}$  также оказывается сходящейся к некоторой величине  $\tilde{J}$  и при этом  $J(x) = J(x') = \tilde{J}$ .

Ниже будет показано, что точка  $x'$  получается из  $x$  последовательной минимизацией вдоль некоторых ортогональных направлений  $q^1, \dots, q^n$ . Поэтому в силу четвертого предположения из равенства  $J(x) = J(x')$  следует  $x = x'$ . Это значит, что убывания  $J$  нет ни в одном из направлений  $q^i$ , то есть проекции вектора градиента  $J'(x)$  на  $q^i, i \in [1: n]$  равны нулю. Отсюда следует, что  $J'(x) = 0$ .

Для завершения доказательства покажем, что существуют такие ортогональные направления  $q^1, \dots, q^n$ , которые позволяют получить точку  $x'$  из  $x$  с помощью последовательной минимизации по этим направлениям.

Пусть  $D_k$  – матрица порядка  $n \times n$ , столбцами которой являются построенные на  $k$ -й итерации направления спуска  $\{d^{ik}, i \in [1: n]\}$ ; то есть  $x^{k+1} = x^k + D_k \lambda^k$ , где  $\lambda^k = (\lambda_{k1}, \dots, \lambda_{kn})^T$  – вектор значений шагов вдоль направлений  $d^{1k}, \dots, d^{nk}$ , соответственно.

Обозначая  $y^{1k} = x^k$ ,  $y^{j+1, k} = y^{jk} + \lambda_{kj} d^{jk}$ ,  $j \in [1: n]$ , получим, что  $x^{k+1} = y^{n+1, k}$  и

$$J(y^{j+1, k}) \leq J(y^{jk} + \tilde{\lambda} d^{jk}), \tilde{\lambda} \in R^1, j \in [1: n]. \quad (6)$$

Рассмотрим последовательность матриц  $\{D_k\}$ ,  $k \in K'$ . Из этой последовательности можно выделить подпоследовательность, для которой первый столбец  $d^{1k}$  будет сходиться к некоторому вектору  $q^1$ . Это следует из того, что множество векторов, норма которых равна единице, компактно. Из этой подпоследовательности можно снова извлечь подпоследовательность, для которой не только  $d^{1k} \rightarrow q^1$ , но и  $d^{2k} \rightarrow q^2$  и т. д. В итоге существует такое  $K'' \subset K'$ , что  $D_k \rightarrow Q$ . Причем  $\det Q = 1 \neq 0$ , так как для всех  $k$   $\det D_k = 1$ . Имеем, таким образом,

$$x^k \rightarrow x, x^{k+1} \rightarrow x', D_k \rightarrow Q, k \in K''. \quad (7)$$

Поэтому из соотношения  $x^{k+1} = x^k + D_k \lambda^k$  имеем

$$\lambda^k \rightarrow \lambda = Q^{-1}(x' - x). \quad (8)$$

Следовательно,  $x' = x + Q\lambda$ . Пусть  $y_1 = x$

и  $y^{j+1} = y^j + \lambda_j d^j$ . Чтобы доказать, что вектор  $x'$  может быть получен из  $x$  последовательной минимизацией  $J$  по направлениям  $q^i$ ,  $i \in [1: n]$ , достаточно показать, что

$$J(y^{j+1}) \leq J(y^j + \tilde{\lambda} q^j), \tilde{\lambda} \in R^1, j \in [1: n]. \quad (9)$$

Из (7), (8) следует, что  $y^k \rightarrow y^j$ ,  $j \in [1: n + 1]$  при  $k \in K^n$ . Поэтому в силу непрерывности  $J$  неравенство (9) следует из (6). Теорема доказана полностью.

Сделаем несколько замечаний общего характера. Вопросам выяснения условий сходимости алгоритмов оптимизации уделяется большое внимание. При этом исследуется сходимость последовательности  $\{x^k\}$ , генерируемой алгоритмом, к некоторому фиксированному множеству  $X^*$ , которое называется множеством решений. Наиболее естественный подход к введению понятия множества решений заключается в определении  $X^* = \{x^* \in D \subset R^n \mid J(x^*) \leq J(x), x \in D\}$ , где  $D$  — множество допустимых точек из  $R^n$ . Иначе говоря,  $x^*$  — решение, если это точка глобального минимума. Однако в действительности такой подход не всегда возможен, и мы вынуждены останавливать процесс при выполнении более слабых предположений относительно полученных алгоритмом точек. Чаще всего для описания множества  $X^*$  применяются необходимые условия экстремума, и тогда полагают:  $X^* = \{x^* \in D \subset R^n \mid J'(x^*) = 0\}$ . Возможны и другие способы задания  $X^*$ , которые столь же легко позволяют проверить принадлежность точки  $x^k$  множеству решений. Типичное для приложений множество решений может быть задано следующим образом:

$$X^* = \{x^* \in D \subset R^n \mid J(x^*) \leq \tilde{J}\},$$

где  $\tilde{J}$  — некоторое приемлемое значение минимизируемого функционала. Очевидно, что сходящийся для фиксированного  $X^*$  алгоритм может оказаться несходящимся, если определить множество решений другим способом.

Сходимость алгоритма при заданном  $X^*$  является чрезвычайно желательным свойством. Однако полезность этого свойства часто переоценивается. Дело в том, что, исследуя бесконечные последовательности генерируемых алгоритмом точек, мы

фактически обращаемся к некоторым математическим фикциям. В действительности всегда имеют дело с конечной последовательностью и свойство сходимости алгоритма в общем случае не является ни необходимым, ни достаточным для окончательной оценки алгоритма.

Другим фактором, несколько снижающим ценность понятия сходимости, являются различные погрешности, которые всегда присутствуют в реальных вычислениях и почти никогда не фигурируют в доказательствах теорем о сходимости. Вместе с тем известны многочисленные примеры, когда влияние погрешностей оказывается решающим.

#### Реализация методов обобщенного покоординатного спуска

При реализации методов ОПС в первую очередь необходимо учитывать три фактора, связанных с нормализацией и масштабированием основных переменных задачи, построением аппроксимации матрицы  $G(x)$ , а также с выбором метода диагонализации матрицы  $G(x)$ .

Цель *нормализации* состоит в улучшении обусловленности задачи, а также в согласовании масштабов основных переменных, часто имеющих различный, физический смысл и измеряемых в различных единицах.

*Масштабирование* может выполняться на двух уровнях: на уровне пользователя и на уровне стандартных программных модулей, реализующих конкретный алгоритм параметрической оптимизации. Как показывает опыт решения реальных задач, эффективность процедуры масштабирования существенно зависит от конкретной структуры решаемой задачи. Поэтому, как правило, целесообразно выполнять масштабирование на уровне пользователя, несмотря на известное усложнение процесса подготовки задачи к ее компьютерной реализации. Наиболее часто переход от исходных переменных  $x = (x_1, \dots, x_n)$  к новым  $y = (y_1, \dots, y_n)$  осуществляется с помощью замены вида

$$x = Dy, D = \text{diag}(d). \quad (10)$$

Диагональная матрица масштабов  $D$

хранится в соответствующей «общей» области подпрограммы, вычисляющей значения  $J(x)$ , и модифицируется в начале каждого нового цикла оптимизации исходя из равенства

$$d_i = \min\{\delta_2, \max\{|\tilde{x}_i|, \delta_1\}\}, \delta_i > 0, \quad (11)$$

где  $\tilde{x}_i$  — лучшее значение  $i$ -й переменной, полученное к началу следующего цикла;  $\delta_1 \cong \varepsilon_M$  — заданное пороговое значение, исключающее появление нулевых  $d_i$ ;  $\delta_2 \cong \varepsilon_M^{-1}$ . В методах второго порядка, рассматриваемых в данной статье, считается, что новый цикл начинается с процедуры вычисления аппроксимации матрицы вторых производных  $G(x)$ . Начальное масштабирование проводится на основе заданных начальных значений  $x_i^0$ . В ряде случаев лучшие результаты дает комбинированный метод, использующий дополнительное масштабирование на основе принципа «равного влияния», согласно которому диагональная матрица масштабов  $D$  выбирается из условия балансирования производных:

$$d_i = k\gamma_i, \gamma_i = \left( \left| \frac{\partial J}{\partial x_i} \right| + \varepsilon_M \right)^{-1} + \varepsilon_M, \quad (12)$$

где, например,  $k = 1$ . В этом случае (при отсутствии влияния  $\varepsilon_M$ ) будем иметь

$$\frac{\partial J(Dy)}{\partial y} = \frac{D\partial J}{\partial x} = (1, \dots, 1)^T.$$

При реализации масштабов (12) в начале каждого цикла оптимизации необходимо проводить анализ чувствительности для получения грубой аппроксимации вектора  $\partial J/\partial x$  в окрестности точки  $\tilde{x}$ .

Двух указанных методов нормализации управляемых параметров, обычно применяемых совместно, бывает достаточно для решения большинства практических задач методами параметрической оптимизации.

Обратимся к нормализации значений минимизируемого функционала. Теоретически масштаб значений целевого функционала  $J(x)$  не оказывает влияния на процесс поиска. Однако при численной реализации алгоритма выбор масштаба  $J(x)$  является

важным фактором. Как показывает практика, целесообразно поддерживать значения  $J(x)$  на уровне  $J(x) \cong 1$  с целью уменьшения вычислительных погрешностей и предотвращения влияния концов диапазона чисел, представимых в памяти компьютера. Для этого в подпрограмме пользователя значения  $J(x)$  умножаются на постоянный для текущего цикла оптимизации множитель. Как правило, необходимо вначале нормализовать управляемые параметры, затем промасштабировать значения  $J(x)$ , а далее перейти к уравниванию производных.

Рассмотрим методы диагонализации. В качестве основной процедуры приведения симметричной матрицы к главным осям может быть использован метод Якоби [11], несмотря на наличие конкурирующих, вообще говоря, более эффективных вычислительных схем [12]. Данный выбор обусловлен следующими обстоятельствами. Во-первых, вычисленные методом Якоби собственные векторы всегда строго ортонормальны с точностью, определяемой точностью компьютера, даже при кратных собственным числах. Последнее весьма существенно при использовании этих векторов в качестве базиса, т. к. предотвращается возможность вырождения базиса, существующая, например, в методе Пауэлла [13]. Во-вторых, многие вычислительные схемы имеют преимущество перед методом Якоби лишь при решении частичной проблемы собственных значений. В нашем же случае всегда решается полная проблема, и поэтому выигрыш во времени оказывается несущественным при существенно более сложных вычислительных схемах. В-третьих, алгоритмы, основанные на методах Якоби, часто оказываются наиболее доступными, т. к. соответствующие программы имеются в большинстве вычислительных систем. И, наконец, определенное влияние на выбор алгоритма оказала простота логики метода Якоби, что приводит к компактности и надежности реализующих его программ.

В задачах большой размерности по



сравнению с методом Якоби более предпочтительным по объему вычислительных затрат оказывается метод, использующий преобразование Хаусхолдера для приведения матрицы к трехдиагональной форме с последующим обращением к  $QR$ -алгоритму определения собственных векторов симметричной трехдиагональной матрицы [12].

В методе Якоби исходная симметричная матрица  $A$  приводится к диагональному виду с помощью цепочки ортогональных преобразований вида

$$A_{k+1} = U_k^T A_k U_k, A_0 = A, k = 1, 2, \dots, \quad (13)$$

являющихся преобразованиями вращения. В результате надлежащего выбора последовательности  $\{U_k\}$  получаем  $\lim A_k = D = U^T A U, k \rightarrow \infty$ , где  $D = \text{diag}(\lambda_i)$  — диагональная матрица;  $U = U_0 U_1 U_2 \dots$  — ортогональная матрица. Так как (4) есть преобразования подобия, то на диагонали матрицы  $D$  расположены собственные числа матрицы  $A$ ; столбцы матрицы  $U$  есть собственные векторы матрицы  $A$ . В действительности вместо матрицы  $U$  получается некоторая матрица  $V = \Lambda(A)$ , отличная, вообще говоря, от истинной матрицы  $U$ . Основные характеристики оператора  $\Lambda$  были рассмотрены выше.

Элементарный шаг (13) процесса Якоби заключается в преобразовании посредством матрицы  $U_k = \{u_{ij}\}$ , отличающейся от единичной элементами  $u_{pp} = u_{qq} = \cos \varphi$ ,  $u_{pq} = u_{qp} = \sin \varphi$ . Угол вращения  $\varphi$  выбирается таким образом, чтобы сделать элемент  $a_{pq}$  матрицы  $A$  нулем. Вопросы сходимости различных численных схем, реализующих метод Якоби, рассмотрены в [14].

За основу может быть взят алгоритм *jacobi* из сборника алгоритмов линейной алгебры [15], реализующий так называемый *частный циклический метод Якоби*. В этом методе аннулированию подвергаются все элементы верхней треугольной части матрицы  $A$  с применением построчного выбора.

При таком выборе индексы элементов  $a_{pq}$  пробегают последовательность значений  $(1, 2), (1, 3), \dots, (1, n); (2, 3), (2, 4), \dots, (2, n); \dots; (n-1, n)$ . Затем начинается новый цикл перебора элементов в том же порядке. Эмпирическая оценка трудоемкости процесса построения матрицы  $\Lambda(A)$  позволяет выразить необходимое время  $T$  работы процессора через размерность  $n$  решаемой задачи. Известно, что для матриц до 50-го порядка и длин машинных слов от 32 до 48 двоичных разрядов общее число циклов в процессе вращений Якоби в среднем не превышает 6–10 (под циклом понимается любая последовательность из  $(n^2 - n)/2$  вращений). При этом  $T \cong kn^3$ , где коэффициент  $k$  определяется быстродействием применяемой вычислительной системы и приблизительно равен  $40t_y$ , где  $t_y$  — время выполнения операции умножения.

Полученная оценка, а также опыт практической работы показывают, что при умеренных  $n$  время реализации оператора  $\Lambda$  для многих практических случаев невелико и сравнимо с временем однократного вычисления значения минимизируемого функционала. Упомянутая выше комбинация метода Хаусхолдера и  $QR$ -алгоритма оказывается приблизительно в 1,5–2 раза быстрее, что может иметь значение при достаточных больших  $n$ .

Различные версии алгоритмов, учитывающие специфику решаемых классов прикладных задач, отличаются, в основном, методами построения аппроксимаций матриц  $G(x)$ . Соответствующие вопросы в данной статье не рассматриваются. Практическое применение изложенного подхода показывает, что при решении жестких оптимизационных задач эффективность методов последовательной релаксации существенно повышается за счет локального сведения исходной жесткой задачи к нескольким слабо связанным задачам с пониженными показателями жесткости.

#### СПИСОК ЛИТЕРАТУРЫ

1. Черноруцкий И.Г. Методы оптимизации. Компьютерные технологии. СПб.: БХВ-

Петербург, 2011. 384 с.

2. Черноруцкий И.Г. Методы параметриче-

ской оптимизации в задачах идентификации // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2009. № 2(76). С. 150–155.

3. **Черноруцкий И.Г.** Параметрические методы синтеза систем управления // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2009. № 2(76). С. 111–115.

4. **Черноруцкий И.Г.** Алгоритмические проблемы жесткой оптимизации // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2012. № 6(162). С. 141–152.

5. **Ракитский Ю.В., Устинов С.М., Черноруцкий И.Г.** Численные методы решения жестких систем. М.: Наука, 1979. 208 с.

6. **Черноруцкий И.Г.** Функции релаксации градиентных методов // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2012. № 3(150). С. 66–72.

7. **Черноруцкий И.Г.** Некоторые стандартные схемы параметрической оптимизации // Научно-

технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2012. № 6(162). С. 128–133.

8. **Черноруцкий И.Г.** Градиентные методы с экспоненциальной функцией релаксации // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2013. № 5(181). С. 58–66.

9. **Базара М., Шетти К.** Нелинейное программирование. М.: Мир, 1982.

10. **Зангвилл У.И.** Нелинейное программирование. М.: Сов. радио, 1973.

11. **Уилкинсон Дж.Х.** Алгебраическая проблема собственных значений. М.: Наука, 1970.

12. **Парлетт Б.** Симметричная проблема собственных значений. М.: Мир, 1983.

13. **Химмельблау Д.** Прикладное нелинейное программирование. М.: Мир, 1975.

14. **Фаддеев Д.К., Фаддеева В.Н.** Вычислительные методы линейной алгебры. М.: Физматгиз, 1963.

15. **Уилкинсон Дж.Х., Райнш С.** Справочник алгоритмов на языке АЛГОЛ. М.: Машиностроение, 1976.

#### REFERENCES

1. **Chernorutskiy I.G.** *Metody optimizatsii. Kompyuternyye tekhnologii.* St. Petersburg: VKhV-Petersburg Publ., 2011, 384 p. (rus)

2. **Chernorutskiy I.G.** Metody parametricheskoy optimizatsii v zadachakh identifikatsii [Methods of parametrical optimisation in identification problems], *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie.* St. Petersburg: SPbGPU Publ., 2009, No. 2(76), Pp. 150–155. (rus)

3. **Chernorutskiy I.G.** Parametricheskiye metody sinteza sistem upravleniya [Parametrical methods of synthesis of control systems], *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie.* St. Petersburg: SPbGPU Publ., 2009, No. 2(76), Pp. 111–115. (rus)

4. **Chernorutskiy I.G.** Algorithmicheskiye problemy zhestkoy optimizatsii [Algorithmic problems of stiff optimization], *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie.* St. Petersburg: SPbGPU Publ., 2012, No. 6(162), Pp. 141–152. (rus)

5. **Rakitskiy Yu.V., Ustinov S.M., Chernorutskiy I.G.** *Chislennyye metody resheniya zhestkikh sistem.* Moscow: Nauka Publ., 1979, 208 p. (rus)

6. **Chernorutskiy I.G.** Funktsii relaksatsii gradiyentnykh metodov [Relaxation function

of gradient methods], *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie.* St. Petersburg: SPbGPU Publ., 2012, No. 3(150), Pp. 66–72. (rus)

7. **Chernorutskiy I.G.** Nekotoryye standartnyye skhemy parametricheskoy optimizatsii [Standard schemes of parametric optimization], *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie.* St. Petersburg: SPbGPU Publ., 2012, No. 6(162), Pp. 128–133. (rus)

8. **Chernorutskiy I.G.** Gradiyentnyye metody s eksponentsialnoy funktsiyey relaksatsii [Gradient methods with exponent relaxation function], *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie.* St. Petersburg: SPbGPU Publ., 2013, No. 5 (181), Pp. 58–66. (rus)

9. **Bazara M, Shetti K.** *Nelineynoye programmirovaniye.* Moscow: Mir Publ., 1982. (rus)

10. **Zangvill U.I.** *Nelineynoye programmirovaniye.* Moscow: Sovetscoe radio Publ., 1973. (rus)

11. **Uilkinson Dzh.Kh.** *Algebraicheskaya problema sobstvennykh znacheniy.* Moscow: Nauka Publ., 1970. (rus)

12. **Parlett B.** *Simmetrichnaya problema sobstvennykh znacheniy.* Moscow: Mir Publ., 1983. (rus)

13. **Khimmelblau D.** *Prikladnoye nelineynoye programmirovaniye.* Moscow: Mir Publ., 1975. (rus) 1963. (rus)
14. **Faddeyev D.K., Faddeyeva V.N.** *Vychislitelnyye metody lineynoy algebrы.* Moscow: Fizmatgiz Publ., 1963. (rus)
15. **Uilkinson Dzh.Kh., Raynsh S.** *Spravochnik algoritmov na yazyke ALGOL.* Moscow: Mashinostroyeniye Publ., 1976. (rus)
- 

**ЧЕРНОРУЦКИЙ Игорь Георгиевич** — заведующий кафедрой информационных и управляющих систем Института информационных технологий и управления Санкт-Петербургского государственного политехнического университета, доктор технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: igcher1946@mail.ru

**CHERNORUTSKIY, Igor G.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: igcher1946@mail.ru