

**СУБГРАДИЕНТНЫЙ МЕТОД МИНИМИЗАЦИИ С КОРРЕКЦИЕЙ ВЕКТОРОВ СПУСКА  
НА ОСНОВЕ ПАР ОБУЧАЮЩИХ СООТНОШЕНИЙ**

*В. Н. Крутиков, Я. Н. Вершинин*

**SUBGRADIENT MINIMIZATION METHOD WITH DESCENT VECTORS CORRECTION  
BY MEANS OF TRAINING RELATIONS PAIRS**

*V. N. Krutikov, Ya. N. Vershinin*

Предложен релаксационный метод сопряженных субградиентов, направление спуска которого корректируется на основе пары текущих обучающих соотношений. Доказана сходимость метода на строго выпуклых функциях. Как показывает численный эксперимент, метод эффективен в задачах минимизации негладких функций высокой размерности. По затратам памяти на хранение информации алгоритм сходен с методом сопряженных градиентов. На гладких функциях большой размерности с высокой степенью вытянутости поверхностей уровня он соизмерим с последним в скорости сходимости.

The paper introduces a conjugate subgradient method whose descent is corrected by a pair of current training relations. The convergence of the method is proved on strictly convex functions. According to the numerical experiment, the method is effective at non-smooth high-dimensional minimization problems. By memory cost, the proposed method is similar to the conjugate gradient method, and at smooth high-dimensional singular functions its convergence rate is not inferior to that of the conjugate gradient method.

**Ключевые слова:** метод минимизации, релаксационный субградиентный метод, метод сопряженных субградиентов, алгоритм Качмажа.

**Keywords:** minimization method, relaxation conjugate subgradients method, Kaczmarz algorithm.

**Введение**

В связи с ростом приложений методов оптимизации в технике, экономике, управлении потребность в методах решения гладких и негладких задач минимизации большой размерности постоянно растет. Как отмечено в [1], метод сопряженных градиентов (МСГ) (см., например [1; 2]) является единственным универсальным средством решения подобных гладких задач. В работах [3 – 7] развит подход построения релаксационных субградиентных методов минимизации (PCM) [8; 9]  $\varepsilon$ -субградиентного типа [9] для решения негладких задач, обладающих при определенных условиях свойствами МСГ [3; 6; 7]. В методах этого типа, в отличие от МСГ, заложено улучшение локальных свойств сходимости в виде цели алгоритма обучения для формирования направления спуска [3 – 7]. Имеющиеся на настоящий момент PCM с растяжением пространства [4 – 7; 10], основываются на взвешенной обучающей информации, собранной на всей траектории минимизации. Они соизмеримы по скорости сходимости на гладких функциях с квазиньютоновскими методами и эффективны при решении негладких задач овражного типа [4 – 7; 10]. В силу необходимости хранения и преобразования матриц их эффективность по затратам времени счета резко снижается на задачах высокой размерности.

Существующие многошаговые PCM [3; 6 – 9], не имеющие в своем составе матриц, используют незначительный объем памяти, необходимой для хранения, имеют незначительные вычислительные затраты на обслуживание итерации метода, что позволяет использовать их при решении большеразмерных задач, но при этом они существенно уступают в скорости сходимости субградиентным методам с растяжением пространства и подвержены закликиванию на овражных задачах негладкой оптимизации, что определяет

актуальность их совершенствования на базе повышения объема используемой обучающей информации. Предлагаемый в работе PCM, в отличие от метода из [3], для цели повышения эффективности алгоритма обучения, основан на использовании дополнительных обучающих соотношений на итерации. Отмеченное нововведение, сравнительно с методом из [3], обеспечивает более широкий диапазон решаемых задач негладкой оптимизации и более высокую скорость сходимости при решении гладких и негладких задач минимизации.

**1. Постановка задачи**

Пусть решается задача минимизации выпуклой на  $R^n$  функции  $f(x)$ . В PCM последовательные приближения строятся по формулам:

$$x_{i+1} = x_i - \gamma_i s_i, \quad \gamma_i = \arg \min_{\gamma} f(x_i - \gamma s_i),$$

где направление спуска  $s_i$  выбирается как решение системы неравенств [9]:

$$(s, g) > 0, \quad \forall g \in G. \quad (1)$$

Здесь множество  $G = \partial_{\varepsilon} f(x_i) - \varepsilon$  –

субградиентное множество в точке  $x_i$ . Обозначим  $S(G)$  – множество решений (1),  $\partial f(x) \equiv \partial f_0(x)$  – субградиентное множество в точке  $x$ . В PCM для решения систем неравенств (1) применяют итерационные методы (алгоритмы обучения), где в качестве элементов  $\varepsilon$ -субградиентных множеств, поскольку их явное задание отсутствует, используют субградиенты, вычисляемые на траектории спуска алгоритма минимизации.

В работах [3 – 7] предложен и используется следующий подход сведения системы (1) к системе ра-

венств. Пусть  $G \subset R^n$  принадлежит некоторой гиперплоскости, а его ближайший к началу координат вектор  $\eta(G)$  является также и ближайшим к началу координат вектором гиперплоскости. Тогда решение системы  $(s, g) = 1, \forall g \in G$  является также решением и для (1). Его можно найти как решение системы [4–7]

$$(s, g_i) = y_i, \quad i = 0, 1, \dots, k, \quad y_i \equiv 1. \quad (2)$$

В [3] (см. также [6; 7]) предложен метод минимизации, в котором для решения системы (2) используется алгоритм Качмажа [11] (см. также [12]):

$$s_{k+1} = s_k + \frac{1 - (s_k, g_k)}{(g_k, g_k)} g_k. \quad (3)$$

Метод (3) дает приближение  $s_{k+1}$ , удовлетворяющее уравнению  $(s, g_k) = 1$ , т. е. последнему поступившему обучающему соотношению из (2). Алгоритм решения задач негладкой оптимизации на основе (3) при точном одномерном спуске на дифференцируемых функциях обладает свойствами метода сопряженных градиентов [3; 6; 7].

В случае ортогональности векторов  $g_i$  в (2) метод (3) на последовательности уравнений из (2) сходится не более чем за  $n$  итераций [6–7]. Можно построить метод решения системы уравнений (2) на последовательной ортогонализации векторов  $g_i$ . В методах оптимизации, в силу несовместности систем равенств, возможной линейной зависимости векторов  $g_i$ , значительных вычислительных затрат на ортогонализацию и необходимости хранения системы векторов, прямые методы полной ортогонализации для решения системы (2) неприемлемы. При определенных ограничениях на ошибки в (2) алгоритмы решения системы равенств с последовательной ортогонализацией векторов  $g_k$  предложены в [13]. В настоящей работе для решения системы неравенств (1) используется схема коррекции направления спуска на основе точного решения двух последних равенств из (2) для пары индексов  $k-1, k$ , что реализуется посредством использования на каждой итерации ортогонализации пары векторов  $g_k, g_{k-1}$  с последующим применением (3).

$$p_k = \begin{cases} g_k, & \text{если } (g_k, g_{k-1}) \geq 0, \\ g_k - \frac{(g_k, g_{k-1})}{\|g_{k-1}\|^2} g_{k-1}, & \text{если } (g_k, g_{k-1}) < 0. \end{cases} \quad (4)$$

где

4. Положить  $k = k + 1$ . Перейти на пункт 2.

Алгоритм А1 отличается от алгоритма решения неравенств на основе алгоритма Качмажа (обозначим его А0) [3; 6; 7] реализацией пункта 3. В А0 вместо (4), (5) используется формула (3).

Согласно (4), (5) равенство  $(s_{k+1}, g_k) = 1$  выполняется всегда, а равенство  $(s_{k+1}, g_{k-1}) = 1$  сохраняется в случае преобразования (5 б), что соответствует точному решению двух последних равенств из (2). В

Основной целью построения направления  $s_k$  в релаксационных субградиентных методах является поиск такого направления, которое обеспечивало бы возможность уменьшения функции из любой точки некоторой окрестности текущего приближения, т. е. решение системы неравенств (1), где множество  $G$  составлено из субградиентов окрестности текущего приближения  $x_k$ . Это означает возможность выхода из этой окрестности посредством минимизации функции вдоль этого направления. Чем шире окрестность, тем больше продвижение в направлении к экстремуму, выше устойчивость метода к ошибкам округления, помехам и способность преодоления малых локальных экстремумов. В этой связи особую важность приобретают изучаемые в работе методы минимизации, в которых, в отличие от метода из [8] и его модификации [9], встроенные алгоритмы решения систем неравенств допускают использование субградиентов достаточно широкой окрестности текущего приближения минимума и не требуют точного одномерного спуска.

### 2. Метод решения неравенств

В излагаемом ниже алгоритме строятся последовательные приближения решения системы неравенств (1) посредством коррекции текущего приближения, обеспечивающей точное решение двух последних равенств типа (2), что реализовано с использованием последовательной ортогонализации пар векторов, предложенной в [13] для решения систем равенств.

Алгоритм А1.

1. Положить  $k = 0, g_{k-1} = 0$ . Задать начальное приближение  $s_0 \in R^n$ .
2. Выбрать произвольно  $g_k \in G$ , удовлетворяющий условию  $(s_k, g_k) \leq 0$ , если такого вектора не существует, то  $s_k \in S(G)$ , закончить работу алгоритма.

3. Получить новое приближение  $s_{k+1}$ :

$$s_{k+1} = s_k + \frac{1 - (s_k, g_k)}{(p_k, g_k)} p_k, \quad (4)$$

случае преобразования (5 б) производится ортогонализация векторов  $g_k, g_{k-1}$ , что отражено в равенстве (6 а). Непосредственно из (5) следует соотношения (6 б). Из (5), с учетом (6 а), получим (6 с):

$$\begin{aligned} \text{а)} & (p_k, g_{k-1}) = 0, \text{ если } (g_k, g_{k-1}) < 0; \\ \text{б)} & (p_k, g_k) \leq (g_k, g_k); \\ \text{с)} & (p_k, p_k) = (p_k, g_k). \end{aligned} \quad (6)$$

Обозначим  $\eta_G \equiv \eta(G)$  – ближайший к началу координат вектор множества  $G$ ,

$$\rho_G \equiv \rho(G) = \|\eta(G)\|, \mu_G = \eta(G) / \|\eta(G)\|,$$

$$s^* = \mu_G / \rho_G, R_G \equiv R(G) = \max_{g \in G} \|g\|.$$

Сделаем предположение относительно множества  $G$ .

**Предположение 1.** Множество  $G$  не пустое, выпуклое, замкнутое, ограниченное  $R_G < \infty$  и удовлетворяет условию отделимости, то есть  $\rho_G > 0$ .

При этих условиях векторы  $\mu_G$  и  $s^*$  являются решениями (1), а для векторов  $g \in G$  выполняются ограничения [4 – 7]

$$1 \leq (s^*, g) \leq R_G / \rho_G, \forall g \in G. \quad (7)$$

Мы изучим сходимость алгоритм А1 к решению  $s^*$ . Обозначим  $\Delta_k = s_k - s^*$  – вектор невязки. Следующие результаты относительно алгоритма А1 получены в условиях справедливости предположения 1.

**Лемма 1.** Пусть последовательность  $\{s_k\}$  получена в результате работы алгоритмом А1. Тогда, для  $k = 0, 1, 2, \dots$  имеют место оценки:

$$(\Delta_{k+1}, p_k) \leq 0, \quad (8)$$

$$(\Delta_k, p_k) \leq (\Delta_k, g_k). \quad (9)$$

Доказательство проведем по индукции. В силу равенства  $p_k = g_k$  при  $k = 0$  выполнено (9). Учитывая левое из неравенств (7) и условие пункта 2  $(s_k, g_k) \leq 0$  получим:

$$-(\Delta_k, g_k) = (s^*, g_k) - (s_k, g_k) \geq 1 - (s_k, g_k) \geq 1. \quad (10)$$

Вычтем из обеих частей (4)  $s^*$ , умножим обе части равенства скалярно на  $p_k$  и преобразуем правую часть с учетом (6 с), (9) и (10):

$$\begin{aligned} (\Delta_{k+1}, p_k) &= (\Delta_k, p_k) + \frac{1 - (s_k, g_k)}{(p_k, g_k)} (p_k, p_k) \leq \\ &\leq (\Delta_k, p_k) - \frac{(\Delta_k, g_k)}{(p_k, g_k)} (p_k, g_k) \leq 0. \end{aligned} \quad (11)$$

Здесь мы предположили, что последний переход в цепочке неравенств произведен при условии (9) для текущего  $k$ . Поскольку неравенство (9) выполняется при  $k = 0$ , то справедливо (11), откуда следует (8) при  $k = 0$ .

Предположим, что неравенства (8), (9) выполнены при  $k = 0, 1, \dots, l - 1$ , где  $l \geq 1$ . Покажем, что они выполняются при  $k = l$ . В случае (5 а) выполнено (9). Для доказательства (9) в случае (5 б) умножим скалярно на  $\Delta_k$  обе части равенства (5 б). Отсюда, в силу справедливости (8) при  $k = l - 1$  и условия  $(g_k, g_{k-1}) < 0$  из (5 б), получим обоснование (9) при  $k = l$

$$(\Delta_k, p_k) = (\Delta_k, g_k) - \frac{(g_k, g_{k-1})}{(g_{k-1}, g_{k-1})} (\Delta_k, g_{k-1}) \leq (\Delta_k, g_k).$$

Из (9) и (11) следует (8) при  $k = l$ . Лемма доказана.

В следующей теореме утверждается, что преобразование (5 б) дает направление  $p_k$  на точку решения  $s^*$  с более острым углом по сравнению с  $g_k$ .

**Теорема 1.** Пусть последовательность  $\{s_k\}$  получена в результате работы алгоритма А1. Тогда, для  $k = 0, 1, 2, \dots$  имеет место оценка:

$$\frac{(-\Delta_k, p_k)}{(p_k, p_k)^{0.5}} \geq \frac{(-\Delta_k, g_k)}{(g_k, g_k)^{0.5}} \geq \frac{1}{R_G}. \quad (12)$$

Доказательство. Из (9) и (10) следует

$$(-\Delta_k, p_k) \geq (-\Delta_k, g_k) \geq 1. \quad (13)$$

Отсюда, с учетом (6 б), (6 с), (9), (10) и определения величины  $R_G$  имеем:

$$\frac{(-\Delta_k, p_k)}{(p_k, p_k)^{0.5}} \geq \frac{(-\Delta_k, p_k)}{(g_k, g_k)^{0.5}} \geq \frac{(-\Delta_k, g_k)}{(g_k, g_k)^{0.5}} \geq \frac{1}{R_G}.$$

Теорема доказана.

Для обоснования сходимости алгоритма А1 нам потребуется следующий результат.

**Лемма 2** [4, 6, 7]. Пусть множество  $G$  удовлетворяет предположению 1. Тогда  $s_k \in S(G)$ , если  $\|\Delta_k\| < 1/R_G$ .

В следующей теореме обосновывается конечная сходимость алгоритма А1. Отметим, что полученные оценки полностью эквивалентны оценкам для алгоритма А0.

**Теорема 2.** Пусть множество  $G$  удовлетворяет предположению 1. Тогда для оценки скорости сходимости последовательности  $\{s_k\}$  к точке  $s^*$ , генерируемой алгоритмом А1 до момента останова, справедливо соотношение:

$$\|s_k - s^*\|^2 \leq (\|s_0\| + \rho_G^{-1})^2 - k/R_G^2, \quad (15)$$

для оценки величины  $\rho_G^{-1}$  имеет место оценка

$$\rho_G^{-1} \geq \left( \sum_{j=0}^k (g_j, g_j)^{-1} \right)^{0.5} - \|s_0\| \geq \frac{k^{0.5}}{R_G} - \|s_0\|, \quad (16)$$

а при некотором значении  $k$ , удовлетворяющем неравенству

$$k \leq k^* \equiv R_G^2 (\|s_0\| + \rho_G^{-1})^2 + 1, \quad (17)$$

будет получен вектор  $s_k \in S(G)$ .

**Доказательство.** Найдем невязку  $\Delta_{k+1}$  вычитанием  $s^*$  из обеих частей (4) и получим выражение квадрата ее нормы. Правую часть полученного выражения преобразуем с учетом неравенств (6 б), (6 с), (9), (10) и определения величины  $R_G$ :

$$\begin{aligned}
 (\Delta_{k+1}, \Delta_{k+1}) &= (\Delta_k, \Delta_k) + 2(\Delta_k, p_k) \frac{1 - (s_k, g_k)}{(p_k, g_k)} + (p_k, p_k) \frac{(1 - (s_k, g_k))^2}{(p_k, g_k)^2} \leq \\
 &\leq (\Delta_k, \Delta_k) - 2 \frac{(1 - (s_k, g_k))^2}{(p_k, g_k)} + \frac{(1 - (s_k, g_k))^2}{(p_k, g_k)} \leq (\Delta_k, \Delta_k) - \frac{1}{(g_k, g_k)} \leq (\Delta_k, \Delta_k) - \frac{1}{R_G^2}.
 \end{aligned}$$

Отсюда, используя неравенство

$$\|s_0 - s^*\|^2 \leq (\|s_0\| + \|s^*\|)^2 = (\|s_0\| + \rho_G^{-1})^2,$$

которое следует из свойств нормы, получим оценки (15) и (16).

Согласно оценке (15) величина  $\|\Delta_k\| \rightarrow 0$ . Поэтому на некотором шаге  $k$  для вектора  $s_k$  будет выполнено неравенство (14), т. е. будет получен вектор  $s_k \in S(G)$ , являющийся решением системы (1). В качестве верхней оценки необходимого числа шагов можно взять  $k^*$ , равное значению  $k$ , при котором правая часть (15) обращается в нуль, увеличенному на 1. Это дает оценку (17). Теорема доказана.

### 3. Субградиентный метод минимизации

Техника обоснования алгоритма соответствует [3; 6; 7]. Пусть функция  $f(x)$ ,  $x \in R^n$  выпукла. Обозначим

$$d(x) = \rho(\partial f(x)), \quad D(z) = \{x \in R^n \mid f(x) \leq f(z)\}.$$

**Примечание 1.** Для выпуклой на  $R^n$  функции, при ограниченности множества  $D(x_0)$  для точек  $x^* \in D(x_0)$ , удовлетворяющих условию  $d(x^*) < d_0$  справедлива оценка [9, с. 291],

$$f(x^*) - f^* \leq D d_0, \quad (18)$$

где  $D$  – диаметр множества  $D(x_0)$ ,  $f^* = \inf_{x \in R^n} f(x)$ .

Дадим описание метода минимизации на основе алгоритма А1 для нахождения точек  $x^* \in R^n$ , таких, что  $d(x^*) \leq E_0$ , где  $E_0 > 0$ .

Алгоритм М1.

1. Задать начальное приближение  $x_0 \in R^n$ , целые  $k = i = 0$ .

2. Положить  $k = k + 1$ ,  $q_k = i$ ,  $s_i = 0$ ,

$$g_{i-1} = 0, \quad \Sigma_i = 0.$$

3. Задать  $\varepsilon_k, m_k$ .

4. Вычислить субградиент  $g_i \in \partial f(x_i)$ , удовлетворяющий условию  $(s_i, g_i) \leq 0$ , если  $g_i = 0$ , то закончить работу алгоритма.

5. Получить новое приближение

$$s_{i+1} = s_i + \frac{1 - (s_i, g_i)}{(p_i, g_i)} p_i, \quad \text{где}$$

$$p_i = \begin{cases} g_i, & \text{если } (g_i, g_{i-1}) \geq 0, \\ g_i - \frac{(g_i, g_{i-1})}{\|g_{i-1}\|^2} g_{i-1}, & \text{если } (g_i, g_{i-1}) < 0. \end{cases}$$

6. Вычислить новое приближение критерия  $\Sigma_{i+1} = \Sigma_i + (g_i, g_i)^{-1}$ .

7. Вычислить новое приближение точки минимума:

$$x_{i+1} = x_i - \gamma_i s_{i+1}, \quad \gamma_i = \arg \min_{\gamma \in R} f(x_i - \gamma_i s_{i+1}).$$

8. Положить  $i = i + 1$ .

9. Если  $1/\sqrt{\Sigma_i} < \varepsilon_k$ , то перейти на пункт 2.

10. Если  $i - q_k > m_k$ , то перейти на пункт 2, иначе перейти на пункт 4.

Алгоритм М1 отличается от известного метода минимизации [3; 6; 7], основанного на алгоритме решения неравенств А0 с формулой Качмажа (3) (назовем его М0), реализацией пункта 5, где вместо формул (4), (5) используется преобразование (3). В алгоритм М1 в пунктах 2,4,5 встроены алгоритмы решения неравенств. Индекс  $q_k$ ,  $k = 0,1,2,\dots$  введен с целью обозначения номеров итераций  $i$ , при которых в пункте 2 при выполнении критериев пунктов 9, 10 происходит обновление для алгоритма решения неравенств ( $p_{i-1} = 0$ ). Согласно (15), (17) алгоритм решения неравенств при  $s_0 = 0$  имеет наилучшие оценки скорости сходимости. Поэтому при обновлении в пункте 2 алгоритма М1 задаем  $s_i = 0$ . Потребность в обновлении возникает вследствие того, что в результате смещения в пункте 7 происходит смена субградиентных множеств окрестности текущей точки достигнутого минимума, что приводит к необходимости решения системы неравенств на основе новой информации.

Отметим, что в силу условия точного одномерного спуска вдоль направления  $(-s_{i+1})$  в пункте 7, в новой точке  $x_{i+1}$  вектор  $g_{i+1} \in \partial f(x_{i+1})$ , такой, что  $(g_{i+1}, s_{i+1}) \leq 0$ , всегда существует согласно необходимому условию минимума одномерной функции (см., например, [9, с. 287]). Следовательно, с учетом роста индекса  $i$  в пункте 8, условие  $(g_i, s_i) \leq 0$  пункта 4 всегда удовлетворяется.

Доказательство сходимости метода М1 опирается на следующую лемму.

**Лемма 3** [9]. Пусть функция  $f(x)$  строго выпукла на  $R^n$ , множество  $D(x_0)$  ограничено, а последовательность  $\{x_k\}_{k=0}^\infty$  такова, что

$$f(x_{k+1}) = \min_{\alpha \in [0,1]} f(x_k + \alpha(x_{k+1} - x_k)).$$

Тогда  $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$ .

Обозначим

$$S_\varepsilon(G) = \{z \in R^n \mid \|z - x\| \leq \varepsilon, \forall x \in G\} - \varepsilon -$$

окрестность множества  $G$ ,

$U_\delta(x) = \{z \in R^n \mid \|z - x\| \leq \delta\} - \delta$  - окрестность точки  $x$ ,  $z_k = x_{q_k}$ ,  $Q_k = \Sigma_{q_k}$ ,  $k = 1, 2, \dots$ , т. е. точки  $x_i$  и значения показателя  $\Sigma_i$ , соответствующие индексам  $i$  на момент обновления в пункте 2 алгоритма М1.

**Теорема 3.** Пусть функция  $f(x)$  строго выпукла на  $R^n$ , множество  $D(x_0)$  ограничено и параметры  $\varepsilon_k, m_k$ , задаваемые в пункте 2 алгоритма М1, фиксированы:

$$\varepsilon_k = E_0 > 0, \quad m_k = M_0, \quad (19)$$

Тогда, если  $x^*$  предельная точка последовательности  $\{x_{q_k}\}_{k=1}^\infty$ , генерируемой алгоритмом М1, то

$$d(x^*) \leq \max\{E_0, R(x_0)/\sqrt{M_0}\} \equiv d_0, \quad (20)$$

где  $R(x_0) = \max_{x \in D(x_0)} \max_{v \in \partial f(x)} \|v\|$ . В частности, если  $M_0 \geq R^2(x_0)E_0^{-2}$ , то  $d(x^*) \leq E_0$ .

**Доказательство.** Существование предельных точек последовательности  $\{z_k\}$  следует из ограниченности множества  $D(x_0)$  и  $z_k \in D(x_0)$ . Допустим, что утверждение теоремы неверно: предположим, что подпоследовательность  $z_{k_s} \rightarrow x^*$ , но

$$d(x^*) = d^* > d_0 > 0. \quad (21)$$

Положим:

$$\varepsilon = (d^* - d_0)/2. \quad (22)$$

Обозначим  $S_\varepsilon^* = S_\varepsilon(\partial f(x^*))$ . Выберем  $\delta > 0$ , такое, что

$$\partial f(x) \subset S_\varepsilon^* \quad \forall x \in S_\delta(x^*). \quad (23)$$

Такой выбор возможен в силу полунепрерывности сверху точно-множественного отображения  $\partial f(x)$  (см. [9, с. 289]).

Выберем номер  $K$ , такой, что при  $k_s > K$  будет справедливо:

$$z_{k_s} \in S_{\delta/2}(x^*), \quad x_i \in S_\delta(x^*), \quad q_{k_s} \leq i \leq q_{k_s} + M_0, \quad (24)$$

т. е. такой номер  $K$ , что точки  $x_i$  остаются в окрестности  $S_\delta(x^*)$  в течение, по крайней мере,  $M_0$  шагов алгоритма. Такой выбор возможен в силу сходимости  $z_{k_s} \rightarrow x^*$  и результата леммы 3, условия которой выполняются при условиях теоремы 3 и наличии точного одномерного спуска в пункте 7 алгоритма М1.

Согласно предположению (21), условиям выбора  $\varepsilon$  (22),  $\delta$  (23) и  $k$  (24) при  $k_s > K$  будет выполняться неравенство:

$$\begin{aligned} \rho(S_\varepsilon^*) &\geq \rho(\partial f(x^*)) - \\ -\varepsilon &= d^* - (d^* - d_0)/2 > d_0. \end{aligned} \quad (25)$$

При  $k_s > K$ , в силу справедливости соотношений (24), из (23) следует:

$$g_i \in S_\varepsilon^*, \quad q_{k_s} \leq i \leq q_{k_s} + M_0.$$

Алгоритм М1 содержит в своем составе алгоритм А1. Поэтому, с учетом оценок из (16), в зависимости от того, в каком из пунктов алгоритма М1 (9 или 10) произойдет обновление при некотором  $i = j$ , будет выполнено одно из неравенств:

$$\rho(S_\varepsilon^*) \leq \Sigma_j^{-0.5} \leq \varepsilon_k \leq E_0 \leq d_0, \quad (26)$$

$$\rho(S_\varepsilon^*) \leq R(x_0)/\sqrt{m_k} \leq R(x_0)/\sqrt{M_0} \leq d_0, \quad (27)$$

где последний переход в неравенствах вытекает из определения величины  $d_0$  в (20). Но (25) противоречит как (26), так и (27). Полученное противоречие доказывает теорему.

Согласно оценке (20), для любой предельной точки последовательности  $\{z_k\}$ , генерируемой алгоритмом М1, будет выполнено  $d(x^*) < d_0$ , а следовательно, будет справедлива оценка (18).

В следующей теореме определяются условия, при которых алгоритм М1 генерирует последовательность  $\{x_i\}$ , сходящуюся к точке минимума.

**Теорема 4.** Пусть функция  $f(x)$  строго выпукла, множество  $D(x_0)$  ограничено и

$$\varepsilon_k \rightarrow 0, \quad m_k \rightarrow \infty. \quad (28)$$

Тогда любая предельная точка последовательности  $\{x_{q_k}\}$ , генерируемая алгоритмом М1, является точкой минимума функции  $f(x)$  на  $R^n$ .

**Доказательство.** Допустим, что утверждение теоремы неверно: предположим, что подпоследовательность  $z_{k_s} \rightarrow x^*$ , но при этом найдется такое  $d_0 > 0$ , что будет выполняться неравенство (21). Как и ранее зададим  $\varepsilon$  согласно (22). Выберем  $\delta > 0$  такое, что будет выполнено (23).

В силу условий (28), найдется такое  $K_0$ , что при  $k > K_0$  будет выполняться соотношение:

$$\max\{\varepsilon_k, R(x_0)/\sqrt{m_k}\} \leq d_0. \quad (29)$$

Обозначим  $E_0 = d_0$  и  $M_0$  - наименьшее значение  $m_k$  при  $k > K_0$ . Дальнейшие рассуждения аналогичны доказательствам теоремы 3.

Выберем номер  $K > K_0$ , такой, что при  $k_s > K$  будет справедливо (24), т. е. такой номер  $K$ , что точ-

ки  $x_i$  остаются в окрестности  $S_\delta(x^*)$  в течение, по крайней мере,  $M_0$  шагов алгоритма. Согласно предположению (21), условиям выбора  $\varepsilon$  (22),  $\delta$  (23) и  $k$  (24) при  $k_s > K$  будет выполняться неравенство (25). При  $k_s > K$ , в силу справедливости соотношений (24), из (23) следует:  $g_i \in S_\varepsilon^*$ ,  $q_{k_s} \leq i \leq q_{k_s} + M_0$ . Алгоритм М1 содержит в своем составе алгоритм А1. Поэтому, с учетом оценок из (16), в зависимости от того, в каком из пунктов алгоритма М1 (9 или 10) произойдет обновление при некотором  $i = j$ , будет выполнено одно из неравенств (26), (27), где последний переход в неравенствах следует из определения величин  $E_0$  и  $M_0$ . Но (25) противоречит как (26), так и (27). Полученное противоречие доказывает теорему.

**4. Реализация алгоритма минимизации**

Алгоритм М1 реализован согласно технике реализации РСМ [3; 6; 7]. Рассмотрим версию алгоритма М1, включающую в себя процедуру одномерной минимизации вдоль направления  $s$ , функции которой заключаются в построении: а) текущего приближения минимума  $x_m$ ; б) точки  $y$  из окрестности  $x_m$ , такой, что для  $g_1 \in \partial f(y)$  выполняется неравенство  $(s, g_1) \leq 0$ . Субградиент  $g_1$  используется для решения системы неравенств.

Обращение к процедуре обозначим:

$$OM(\{x, s, g_x, f_x, h_0\}; \{\gamma_m, f_m, g_m, \gamma_1, g_1, h_1\}).$$

Блок входных параметров состоит из точки текущего приближения минимума  $x$ , направления спуска  $s$ ,  $g_x \in \partial f(x)$ ,  $f_x = f(x)$ , начального шага  $h_0$ . Предполагается, что выполняется необходимое условие возможности спуска  $(g_x, s) > 0$  в направлении  $s$ .

Блок выходных параметров включает в себя  $\gamma_m$  – шаг в точку полученного приближения минимума  $x^+ = x - \gamma_m s$ ,  $f_m = f(x^+)$ ,  $g_m \in \partial f(x^+)$ ,  $\gamma_1$  – шаг вдоль  $s$ , такой, что в точке  $y^+ = x - \gamma_1 s$  для  $g_1 \in \partial f(y^+)$  выполняется неравенство  $(g_1, s) \leq 0$ , и  $h_1$  – начальный шаг спуска для следующей итерации. В излагаемом ниже алгоритме векторы  $g_1 \in \partial f(y^+)$  используются для решения множества неравенств, а точки  $x^+ = x - \gamma_m s$  как точки приближений минимума.

$$s_{i+1} = \begin{cases} \tilde{s}_{i+1}, & \text{если } (\tilde{s}_{i+1}, g_i) \geq 1, \\ \tilde{s}_{i+1} + g_i(1 - (\tilde{s}_{i+1}, g_i)) / (g_i, g_i), & \text{если } (\tilde{s}_{i+1}, g_i) < 1. \end{cases} \quad (32)$$

Алгоритм одномерного спуска (ОМ). Пусть требуется найти приближение минимума одномерной функции  $\varphi(\beta) = f(x - \beta s)$ , где  $x$  – некоторая точка,  $s$  – направление спуска. Возьмем возрастающую последовательность  $\beta_0 = 0$  и  $\beta_i = h_0 q_M^{i-1}$  при  $i \geq 1$ . Обозначим  $z_i = x - \beta_i s$ ,  $r_i \in \partial f(z_i)$ ,  $i = 0, 1, 2, \dots, l$  – номер  $i$ , при котором впервые выполнится соотношение  $(r_i, s) \leq 0$ . Зададим параметры отрезка локализации  $[\gamma_0, \gamma_1]$  одномерного минимума:  $\gamma_0 = \beta_{l-1}$ ,  $f_0 = f(z_{l-1})$ ,  $g_0 = r_{l-1}$ ,  $\gamma_1 = \beta_l$ ,  $f_1 = f(z_l)$ ,  $g_1 = r_l$  и найдем точку минимума  $\gamma^*$  одномерной кубической аппроксимации функции на отрезке локализации. Вычислим:

$$\gamma_m = \begin{cases} q_\gamma \gamma_1, & \text{если } l = 1 \text{ и } \gamma^* \leq q_\gamma \gamma_1, \\ \gamma_1, & \text{если } \gamma_1 - \gamma^* \leq q_\gamma (\gamma_1 - \gamma_0), \\ \gamma_0, & \text{если } l > 1 \text{ и } \gamma^* - \gamma_0 \leq q_\gamma (\gamma_1 - \gamma_0), \\ \gamma^*, & \text{в остальных случаях.} \end{cases} \quad (30)$$

Вычислим начальный шаг спуска для следующей итерации:

$$h_1 = q_m (h_0 \gamma_m)^{1/2}. \quad (31)$$

**Алгоритм минимизации.** В предлагаемом ниже варианте реализации алгоритма М1 обновление для метода решения неравенств не производится, а точный одномерный спуск заменен на приближенный.

**Алгоритм**

1. Задать начальное приближение  $x_0 \in R^n$ , начальный шаг одномерного спуска  $h_0$ . Положить:  $i = 0$ ,  $g_0 = \tilde{g}_0 \in \partial f(x_0)$ ,  $g_{i-1} = 0$ ,  $f_0 = f(x_0)$ ,  $s_0 = \tilde{s}_0 = 0$ . Задать параметры останова:  $N$  – максимально допустимое число итераций,  $\varepsilon_x$  – точность минимизации по аргументу,  $\varepsilon_g$  – точность минимизации по градиенту.

2. Получить приближение

$$\tilde{s}_{i+1} = s_i + \frac{1 - (s_i, \tilde{g}_i)}{(p_i, \tilde{g}_i)} p_i,$$

$$\text{где } p_i = \begin{cases} \tilde{g}_i, & \text{если } (\tilde{g}_i, g_{i-1}) \geq 0, \\ \tilde{g}_i - \frac{(\tilde{g}_i, g_{i-1})}{\|g_{i-1}\|^2} g_{i-1}, & \text{если } (\tilde{g}_i, g_{i-1}) < 0. \end{cases}$$

Здесь осуществляются шаг метода решения неравенств.

3. Получить направление спуска.

4. Произвести одномерный спуск вдоль

$$w_{i+1} = s_{i+1}(s_{i+1}, s_{i+1})^{-1/2};$$

$$OM(\{x_i, w_{i+1}, g_i, f_i, h_i\}; \\ \{\gamma_{i+1}, f_{i+1}, g_{i+1}, \tilde{\gamma}_{i+1}, \tilde{g}_{i+1}, h_{i+1}\}).$$

Вычислить приближение точки минимума

$$x_{i+1} = x_i - \gamma_{i+1}w_{i+1}.$$

5. Если  $i > N$  или  $\|x_{i+1} - x_i\| \leq \epsilon_x$ ,

или  $\|g_{i+1}\| \leq \epsilon_g$ , то закончить вычисления, иначе положить  $i = i + 1$  и перейти на пункт 2.

Поясним действия алгоритма. Из *OM* поступают два субградиента  $\tilde{g}_{i+1}$  и  $g_{i+1}$ . Первый из них используется для решения неравенств в пункте 2, а второй – в пункте 3, для коррекции направления спуска с помощью формулы (3) с целью обеспечения необходимого условия  $(s_{i+1}, g_i) > 0$  возможности спуска в направлении  $(-s_{i+1})$ . Как показано в [3; 4; 6; 7] итерация (3) в (32) при  $(\tilde{s}_{i+1}, g_i) < 1$  не ухудшает текущее приближения  $\tilde{s}_i$  решения системы неравенств, поэтому это преобразование проводится. В пункте 3, когда  $(\tilde{s}_{i+1}, g_i) > 1$ , условие  $(\tilde{s}_{i+1}, g_i) > 0$  выполнено и, согласно результатам работ [3; 4; 6; 7], нет теоретических рекомендаций по улучшению решения системы неравенств. Поэтому преобразование коррекции не производится.

Хотя обоснование сходимости идеализированных версий РСМ [3 – 10] производится при условии точного одномерного спуска, реализации этих алгоритмов осуществляется с процедурами одномерной минимизации, в которых начальный шаг, в зависимости от прогресса, может увеличиваться или уменьшаться, что определяется заданными коэффициентами

$$q_M > 1 \text{ и } q_m < 1.$$

При этом минимальный шаг на итерации не может быть меньше некоторой доли начального шага, величина которой задана в (30) параметрами  $q_{\gamma_1} = 0.1$  и  $q_{\gamma} = 0.2$ , приведенные значения которых использовались нами при расчетах.

### 5. Численный эксперимент

Алгоритм М1 реализован согласно технике реализации алгоритма М0 [3; 6; 7], в которой ключевое значение играют коэффициенты уменьшения  $q_m < 1$  и увеличения  $q_M > 1$  начального шага одномерного спуска на итерации. Значения  $q_m$  близкие к 1 обеспечивают малую скорость убывания шага и, соответственно, малую скорость сходимости метода. При этом малая скорость убывания шага устраняет заикливание метода в силу того, что субградиенты функции, участвующие в решении неравенств берутся из более широкой окрестности. Выбор параметра  $q_m$  должен соизмеряться с возможной скоростью сходимости метода минимизации. Чем выше скоростные возможности алгоритма, тем меньшим может быть выбран этот

параметр. Например, в РСМ с растяжением пространства [4 – 7, 10] выбирается  $q_m = 0.8$ . Для гладких функций выбор этого параметра не критичен и его можно брать из интервала [0.8 – 0.98]. От параметра возрастания шага скорость сходимости практически не зависит, поэтому его можно взять постоянным  $q_M = 1.5$ .

Проведенное тестирование алгоритма М1 на известных негладких и невыпуклых малоразмерных функциях свидетельствует о его работоспособности и достижении задаваемой точности. Например, тестирование на негладких функциях малой размерности из [14], показало что алгоритмы М0 и М1 при  $n = 50$  практически не уступают в скорости сходимости известному методу с растяжением пространства в направлении разности субградиентов [10]. Основная задача приводимого здесь эксперимента состоит в изучении закономерности поведения метода на больших размерных функциях и его сравнении с МСГ на квадратичных функциях, где последний является конечным.

В таблицах приведены результаты для квазиьютоновского метода с формулой BFGS (КНМ) [1], метода сопряженных градиентов (МСГ) [1, 2], алгоритма с растяжением пространства в направлении разности субградиентов (МШ) [10] и алгоритма с растяжением-сжатием пространства (М2) [5; 7]. В методах КНМ и МСГ реализован одномерный поиск с квадратичной интерполяцией. В методах М0, М1, МШ и М2 используется одна и та же процедура одномерной минимизации.

Исследование проводилось на следующих функциях:

$$\begin{aligned} 1) \quad & f_1 = \sum_{k=1}^n |x_k| \cdot k, \quad f_1^* = 0.0, \\ & x^* = (0, 0, \dots, 0), \quad x_0 = (10, \frac{10}{2}, \frac{10}{3}, \dots, \frac{10}{n}); \\ 2) \quad & f_2 = \sum_{k=1}^n x_k^2 \cdot k^2, \quad f_2^* = 0.0, \\ & x^* = (0, 0, \dots, 0), \quad x_0 = (10, \frac{10}{2}, \frac{10}{3}, \dots, \frac{10}{n}); \\ 3) \quad & f_3(x) = \sum_{i=1}^{n-1} [1000(x_i - x_{i+1})^2 + (1 - x_{i+1})^2], \\ & f_3^* = 0.0, \quad x^* = (1, 1, \dots, 1), \quad x_0 = (0, 0, \dots, 0). \end{aligned}$$

Функция  $f_3(x)$  взята из [14].

В таблицах 1 и 2 приведено количество затраченных методом вычислений функции и субградиента, которое соответствует моменту выполнения условия  $f_k - f^* \leq \epsilon$ . Знаком N помечены задачи, которые не удалось решить за количество итераций не превышающее заданное максимальное, т. е. найти точку приближения  $x_k$ , в которой бы выполнялось условие останова алгоритма по значению функции, т. е.  $f_k - f^* \leq \epsilon$ . При тестировании другие критерии останова не использовались.

Результаты численного эксперимента для функции  $f_2$ 

| $n$  | $f_2, \varepsilon = 10^{-10}$ |           |           |            | $f_2, \varepsilon = 10^{-10}, q_{MIN} = 0.98, q_{MAX} = 1.5$ |           |
|------|-------------------------------|-----------|-----------|------------|--|-----------|
|      | <i>КНМ</i>                    | <i>М2</i> | <i>МШ</i> | <i>МСГ</i> | <i>М0</i>  | <i>М1</i> |
| 100  | 207                           | 266       | 595       | 938        | 2064   | 1709      |
| 200  | 409                           | 420       | 1257      | 2359       | 4008   | 2668      |
| 300  | 624                           | 585       | 2059      | 3891       | 5781   | 3729      |
| 400  | 814                           | 732       | 2887      | 5929       | 7804   | 4898      |
| 500  | 1018                          | 893       | 3734      | 7632       | 10086  | 5904      |
| 600  | 1225                          | 1037      | 4523      | 9264       | 12457  | 7269      |
| 700  | 1419                          | 1203      | 5365      | 10914      | 14837  | 8705      |
| 800  | 1624                          | 1356      | 6214      | 12563      | 17345  | 10201     |
| 900  | 1831                          | 1513      | 6967      | 14272      | 19839  | 11816     |
| 1000 | 2030                          | 1688      | 7825      | 16008      | 22478  | 13138     |

Функция 2 квадратичная с отношением собственных значений  $1/n^2$ . На функции 2 МСГ требуется существенно большее количество итераций для достижения заданной точности, чем  $n$ . В методе КНМ количество итераций равно  $n$ . В методе М2 при  $n > 200$  за счет использования грубого одномерного поиска общие затраты NFG меньше, чем в методе КНМ. Тем не менее при существенно больших затратах NFG в методах М0, М1 и МСГ сравнительно с методами КНМ, М2 и МШ, они затрачивают существенно меньше времени счета, чем алгоритмы, использующие матрицы преобразования пространства. Подобные преимущества по времени счета могут иметь место при решении задач, где затраты на вычисление функции и

градиента пропорциональны размерности. На функции 2 алгоритм М1 эффективнее метода М0 и сравним с МСГ при размерностях выше 300 на функции 2.

На квадратичной функции 3 с небольшим разбросом собственных значений методы М0 и М1 практически эквивалентны, причем при росте размерности затраты М1 соизмеримы с затратами МСГ. В силу малой сложности функции количество итераций МСГ меньше размерности решаемой задачи.

На основании этих результатов можно сделать вывод, что при решении гладких задач минимизации высокой размерности с высокой степенью вытянутости поверхностей уровня наряду с МСГ возможно и применение многошаговых РСМ.

Таблица 2

Результаты численного эксперимента для функций  $f_1$  и  $f_3$ 

| $n$  | $f_1, \varepsilon = 10^{-5}$ | $f_1, \varepsilon = 10^{-5}$ | $f_1, \varepsilon = 10^{-5}, q_{MIN} = 0.99905, q_{MAX} = 1.5$ |           | $f_3, \varepsilon = 10^{-10}$ | $f_3, \varepsilon = 10^{-10}, q_{MIN} = 0.85, q_{MAX} = 1.5$ |           |
|------|------------------------------|------------------------------|--|-----------|-------------------------------|--|-----------|
|      | <i>М2</i>                    | <i>МШ</i>                    | <i>М0</i>  | <i>М1</i> | <i>МСГ</i>                    | <i>М0</i>  | <i>М1</i> |
| 100  | 930                          | 2258                         | 28995  | 28759     | 203                           | 760  | 457       |
| 200  | 1996                         | 4250                         | 89281 (N)  | 30913     | 404                           | 869  | 562       |
| 300  | 3137                         | 8251                         | 89261 (N)  | 32185     | 604                           | 903  | 633       |
| 400  | 4248                         | 10237                        | 89289 (N)  | 33283     | 715                           | 885  | 603       |
| 500  | 5235                         | 12932                        | 89211 (N)  | 33981     | 723                           | 947  | 697       |
| 600  | 6467                         | 16156                        | 89101 (N)  | 34593     | 727                           | 935  | 657       |
| 700  | 7456                         | 19670                        | 89205 (N)  | 35105     | 729                           | 975  | 672       |
| 800  | 8786                         | 24201                        | 89469 (N)  | 35371     | 729                           | 960  | 704       |
| 900  | 9717                         | 26184                        | 89415 (N)  | 36013     | 731                           | 948  | 673       |
| 1000 | 11342                        | 28439                        | 89167 (N)  | 36013     | 731                           | 967  | 671       |

Кусочно-линейная функции 1 имеет одинаковую вытянутость линий уровня с функцией 2, но неизмеримо сложнее для метода минимизации. Здесь в методе М0 происходит заикливание, а увеличение числа итераций не приводит к решению задачи. На этом примере заметно существенное повышение эффективности алгоритма М1 сравнительно с М0 за счет использования дополнительного обучающего соотношения при решении неравенств.

Анализируя результаты для МШ и М1 приходим к выводу об уменьшении разрыва между ними по затратам NFG при росте размерности. При решении этой задачи для  $n = 1000$  уже не наблюдается десятикратного превосходства метода М2, как это было на функции 2. При этом затраты времени решения задачи 1 у метода М1 существенно меньше, чем у методов М2 и МШ. На основе полученных результатов можно сделать вывод, что при решении задач, где затраты на



вычисление функции и градиента пропорциональны размерности, метод М1 может оказаться предпочтительнее методов с изменением метрики пространства. Тем не менее, следует отметить, что при росте степени вырожденности задачи, граница этой степени, при которой метод в состоянии решить задачу, выше у субградиентных методов минимизации с растяжением пространства по сравнению с границей для М1. На негладких функциях оказывается существенным выбор коэффициента  $q_m$ , который определяет широту окрестности, из которой берутся субградиенты для встроенного в метод минимизации алгоритма решения неравенств. В силу того, что практически важную задачу минимизации, как правило, приходится решать многократно, то параметр  $q_m$  можно предварительно подобрать экспериментально.

#### Заключение

В работе итерационный метод решения системы равенств, использующий на итерации коррекцию ре-

шения, обеспечивающую точное решение двух последних из них [13], распространен на задачу решения множества неравенств. Разработанный метод обоснован теоретически, получены оценки его скорости сходимости. На основе предложенного метода сформулирован и обоснован новый релаксационный субградиентный алгоритм минимизации, который пригоден для решения, в том числе и невыпуклых задач.

По свойствам сходимости на квадратичных функциях высокой размерности, при больших разбросах собственных значений, разработанный алгоритм эквивалентен методу сопряженных градиентов. Новый метод позволяет решать негладкие большеразмерные задачи минимизации с высокой степенью вытянутости поверхностей уровня, где он при затратах на вычисление функции и субградиента пропорциональных размерности, может превосходить по времени счета известные методы минимизации с растяжением пространства.

#### Литература

1. Гилл, Ф. Практическая оптимизация / Ф. Гилл, У. Мюррей, М. Райт. – М.: Мир, 1985. – 509 с.
2. Поляк, Б. Т. Введение в оптимизацию / Б. Т. Поляк. – М.: Наука. – 1983. – 384 с.
3. Крутиков, В. Н. Новый релаксационный метод недифференцируемой минимизации / В. Н. Крутиков, Т. В. Петрова // Математические заметки ЯГУ. – 2001. – Т. 8. – Вып. 1. – С. 50 – 60.
4. Крутиков, В. Н. Релаксационный метод минимизации с растяжением пространства в направлении субградиента / В. Н. Крутиков, Т. В. Петрова // Экономика и мат. методы. – 2003. – Т. 39. – Вып. 1. – С. 33 – 49.
5. Крутиков, В. Н. Семейство релаксационных субградиентных методов с двухэтапной коррекцией матриц метрики / В. Н. Крутиков, Т. А. Горская // Экономика и мат. методы. – 2009. – Т. 45. – № 4. – С. 37 – 80.
6. Крутиков, В. Н. Релаксационные методы безусловной оптимизации, основанные на принципах обучения: учебное пособие / В. Н. Крутиков. – Кемерово: Кузбассвузиздат, 2004. – 171 с.
7. Крутиков, В. Н. Обучающиеся методы безусловной оптимизации и их применение / В. Н. Крутиков. – Томск: Изд-во Том. гос. пед. ун-та, 2008. – 264 с.
8. Wolfe, P. Note on a method of conjugate subgradients for minimizing nondifferentiable functions / P. Wolfe // Math. Programming. – 1974. – V. 7. – № 3. – P. 380 – 383.
9. Демьянов, В. Ф. Недифференцируемая оптимизация / В. Ф. Демьянов, Л. В. Васильев. – М.: Наука, 1972. – 368 с.
10. Шор, Н. З. Методы минимизации недифференцируемых функций и их приложения / Н. З. Шор. – Киев: Наукова думка, 1979. – 199 с.
11. Kaczmarz, S. Approximate solution of systems of linear equations / S. Kaczmarz // Internat. J. Control. – 1993. – V. 54. – № 3. – P. 1239 – 1241.
12. Цыпкин, Я. З. Основы теории обучающихся систем / Я. З. Цыпкин. – М.: Наука, 1981. – 251 с.
13. Крутиков, В. Н. Алгоритмы обучения на основе ортогонализации последовательных векторов / В. Н. Крутиков, Я. Н. Вершинин // Вестник КемГУ. – 2012. – Вып. 2(50). – С. 37 – 42.
14. Скоков, В. А. Варианты метода уровней для минимизации негладких выпуклых функций и их численное исследование / В. А. Скоков // Экономика и математические методы. – 1997. – Т. 33. – № 1.
15. Банди, Б. Методы оптимизации. Вводный курс / Б. Банди; пер. с англ. – М.: Радио и связь, 1968. – 128 с.

#### Информация об авторах:

**Крутиков Владимир Николаевич** – доктор технических наук, профессор кафедры математической кибернетики КемГУ, 8-905-077-53-48, krutikovvn@gmail.com.

**Vladimir N. Krutikov** – Doctor of Techniocal Science, Professor at the Department of Mathematical Cybernetics, Kemerovo State University.

**Вершинин Ярослав Николаевич** – аспирант кафедры математической кибернетики КемГУ, 8-960-919-74-13, Azimus88@gmail.com.

**Yaroslav N. Vershinin** – post-graduate student at the Department of Mathematical Cybernetics, Kemerovo State University.

Статья поступила в реколлегию 21.10.2013 г.