КОМПЬЮТЕРНАЯ МАТЕМАТИКА

УДК 519.85

П.И. СТЕЦЮК, А.В. ИВЛИЧЕВ, А.А. ИЩЕНКО

О СХОДИМОСТИ $r_{«}(\sim)$-АЛГОРИТМА

Приводится описание $r_{«}(\sim)$-алгоритма для минимизации почти-дифференцируемой функции. Рассматривается кусочно-линейная выпуклая функция, для которой две точки с линейно зависимыми почти-градиентами могут служить «ловушками» для $r_{«}(\sim)$-алгоритма. Для минимизации выпуклой функции предложен $r_0(\sim)$-алгоритм. Показано, что его нельзя «зациклить» в точках-ловушках рассмотренной кусочно-линейной выпуклой функции.

Введение. $r_\mu(\alpha)$-алгоритм – это один из вариантов r-алгоритмов [1]. Он был предложен Н.З. Шором в 1975 году [2] для минимизации почти-дифференцируемых функций. С $r_\mu(\alpha)$-алгоритмом связаны самые сильные результаты о сходимости r-алгоритмов: при определенных условиях $r_\mu(\alpha)$-алгоритм сходится к локальному минимуму почти-дифференцируемой кусочно-гладкой функции [1, 2]. Эти условия не всегда выполняются даже для кусочно-линейных выпуклых функций [3].

Далее опишем $r_\mu(\alpha)$-алгоритм и условия его применения для минимизации почти-дифференцируемых функций. Обсудим пример кусочно-линейной выпуклой функции от двух переменных, такой, что $r_\mu(\alpha)$-алгоритм «зацикливается» в точке, которая не является точкой минимума. Центральный результат статьи – это построенная модификация $r_\mu(\alpha)$-алгоритма, преодолевающая точки-ловушки для кусочно-линейной выпуклой функции.

Материал статьи изложен в таком порядке. В разделе 1 описаны почти-дифференцируемые функции и их свойства. В разделе 2 описан $r_\mu(\alpha)$-алгоритм и приведена основная теорема о его сходимости. В разделе 3 обсуждается ситуация с зацикливанием $r_\mu(\alpha)$-алгоритма для кусочно-линейной выпуклой функции. В разделе 4 описана модификация $r_\mu(\alpha)$-алгоритма – $r_0(\alpha)$-алгоритм и показано, что его нельзя «зациклить» в точках-ловушках для кусочно-линейной выпуклой функции от двух переменных.

1. Почти-дифференцируемые функции и их свойства. Понятия почти-дифференцируемой функции и почти-градиента введены Н.З. Шором в [4]. Дадим их определения и теоремы о свойствах почти-дифференцируемых функций, которые нам понадобятся в дальнейшем.

Определение 1 [4]. Непрерывная функция $f(x)$, заданная в открытой области S n-мерного эвклидового пространства E^n, называется почти-дифференцируемой, если: а) в любой ограниченной области $f(x)$ удовлетворяет условию Липшица; б) функция $f(x)$ почти везде дифференцируема; в) градиент функции $f(x)$ непрерывен на множестве $M \subseteq S$, где он определен.

Определение 2 [4]. Почти-градиентом функции $f(x)$ в точке x называется вектор, являющийся предельной точкой некоторой последовательности градиентов $g(x_1)$, $g(x_2)$, …, $g(x_k)$, …, где $\{x_k\}_{k=1}^{\infty}$ – последовательность точек сходящаяся к точке x, и такая, что во всех точках этой последовательности функция $f(x)$ дифференцируема.

Множество $G_f(x)$ – множество почти-градиентов в любой точке x почти-дифференцируемой функции $f(x)$ является непустым, ограниченным и замкнутым.

Теорема 1 [4]. Произвольная выпуклая функция $f(x)$, определенная в E^n, является почти-дифференцируемой, а ее любой почти-градиент в точке x совпадает с некоторым субградиентом.

Теорема 2 [4]. Если две функции $f_1(x)$ и $f_2(x)$ – почти-дифференцируемы, то $f_3(x) = f_1(x) + f_2(x)$ и $f_4(x) = f_1(x) \cdot f_2(x)$ – также почти-дифференцируемы.

Теорема 3 [4]. Если функции $f_1(x)$ и $f_2(x)$ – почти-дифференцируемы, то функция $f(x) = \max(f_1(x),\ f_2(x))$ – также почти-дифференцируема.

Ввести класс почти-дифференцируемых функций Н.З. Шора побудило отсутствие свойств выпуклости и гладкости для широкого круга минимаксных задач и задач нелинейного программирования, а также задач нахождения решений систем нелинейных уравнений. Так, например, решение системы нелинейных уравнений: $\varphi_i(x) = 0$, $x \in E^n$, $i = 1,...,m$, где $\varphi_i(x)$ – почти-дифференцируемые функции, можно свести к нахождению точек минимума таких функций $\psi_1(x) = \max\limits_{1 \le i \le m} |\varphi_i(x)|$, либо $\psi_2(x) = \sum\limits_{i=1}^{m} |\varphi_i(x)|$, либо $\psi_3(x) = \sum\limits_{i=1}^{m} \varphi_i^2(x)$. Легко увидеть (см. теоремы 2 и 3), что функции $\psi_1(x)$, $\psi_2(x)$, $\psi_3(x)$ будут почти-дифференцируемыми. Точка локального минимума x^*, для которой $\psi_j(x^*) = 0$, $j = 1, 2, 3$, соответствует одному из решений системы нелинейных уравнений.

2. $r_\mu(\alpha)$-алгоритм. Для минимизации почти-дифференцируемых функций в [2] Н.З. Шор предложил $r_\mu(\alpha)$-алгоритм – метод градиентного типа с растяжением пространства переменных в направлении разности двух последовательных почти-градиентов. Ключевую роль здесь играет оператор растяжения пространства, который в матрично-векторной форме имеет вид

$$R_\alpha(\xi) = I_n + (\alpha - 1)\xi\xi^T, \quad \xi \in E^n, \quad \|\xi\| = 1, \quad \alpha > 1,$$

где $(\cdot)^T$ означает транспонирование, I_n – единичная $n \times n$-матрица, α – коэффициент растяжения пространства, ξ – направление растяжения. Подробно свойства оператора $R_\alpha(\xi)$ изложены в монографии [1, стр. 68 – 69].

При описании алгоритма в B-форме (на каждой итерации корректируется матрица B_k) используется оператор

$$R_\beta(\xi) = R_\alpha^{-1}(\xi) = I_n + (\beta - 1)\xi\xi^T, \quad \beta = \frac{1}{\alpha} < 1.$$

Оператор $R_\beta(\xi)$ является обратным к оператору $R_\alpha(\xi)$ и в методах с растяжением пространства переменных он обеспечивает одноранговый пересчет матрицы B_{k+1} по такому правилу:

$$B_{k+1} = B_k R_\beta(\xi) = B_k(I_n + (\beta - 1)\xi\xi^T) = B_k + (\beta - 1)(B_k\xi)\xi^T.$$

Пусть $f(x)$ – почти-дифференцирумая функция; $G_f(x)$ – множество почти-градиентов $f(x)$ в точке x; $\alpha > 1$ – коэффициент растяжения пространства; μ – константа, такая что $0 \le \mu < 1$; x_0 – начальная точка; $g_f(x_0)$ – почти-градиент в точке x_0; B_0 – невырожденная $n \times n$-матрица (в качестве матрицы B_0 часто выбирают либо единичную матрицу I_n, либо диагональную матрицу D_n с положительными элементами на диагонали, с помощью которой осуществляется масштабирование переменных).

$r_\mu(\alpha)$-алгоритмом называется итерационная процедура построения последовательностей векторов $\{x_k\}_{k=0}^\infty$ и матриц $\{B_k\}_{k=0}^\infty$, где переход от k-ой итерации к $(k+1)$-ой итерации осуществляется по следующему правилу:

1) вычислим очередную точку

$$x_{k+1} = x_k - h_k B_k g_{\varphi_k}(y_k), \tag{1}$$

где $g_{\varphi_k}(y_k) = B_k^T g_f(x_k)$, а величина шага h_k выбирается из условий:

a) на отрезке $[0, h_k]$ функция $\varphi_k(h) = f(x_{k+1}(h))$ не возрастает;

b) существует $g \in G_f(x_{k+1})$ такой, что $\dfrac{(B_k^T g, g_{\varphi_k}(y_k))}{\|B_k^T g\| \|g_{\varphi_k}(y_k)\|} \le \mu$;

2) пересчитаем матрицу (преобразования пространства)

$$B_{k+1} = B_k R_\beta(\xi_k) = B_k + (\beta - 1)\left(B_k \xi_k\right)\xi_k^T, \qquad (2)$$

где

$$\xi_k = \frac{B_k^T r_k}{\| B_k^T r_k \|}, \quad r_k = g - g_f(x_k), \quad \beta = \frac{1}{\alpha} < 1;$$

3) переходим к очередной итерации с x_{k+1}, B_{k+1} и $g_f(x_{k+1}) = g$.

Здесь h_k – неотрицательный шаг на k-ой итерации (может равняться нулю), $R_\beta(\xi) = I_n + (\beta - 1)\xi\xi^T$ – оператор «сжатия» пространства почти-градиентов в нормированном направлении ξ с коэффициентом $\beta = \dfrac{1}{\alpha} < 1$, $g_f(x_k)$ и $g_f(x_{k+1})$ – почти-градиенты функции $f(x)$ в точках x_k и x_{k+1}.

Комментарий. Если обе части формулы (1) умножить слева на матрицу $A_k = B_k^{-1}$, то получаем

$$y_{k+1} = A_k x_{k+1} = A_k x_k - h_k g_{\varphi_k}(y_k) = y_k - h_k g_{\varphi_k}(y_k).$$

Таким образом формула (1) фактически реализует шаг градиентного спуска для функции $\varphi_k(y) = f(B_k y)$. При $\mu = 0$ получается аналог шага наискорейшего спуска. Введение константы $\mu \ge 0$ дает возможность рассматривать алгоритмы, в которых поиск минимума по направлению производится приближенно.

Теорема 4 [2]. Пусть $f(x)$ – почти-дифференцируемая функция, такая, что $\lim\limits_{\|x\| \to \infty} f(x) = +\infty$, и последовательность $\{x_k\}_{k=0}^{\infty}$, полученная в результате реализации $r_\mu(\alpha)$-алгоритма, удовлетворяет условию

$$\lim_{k \to +\infty} \left\| x_{k+1} - x_k \right\| = 0. \qquad (3)$$

Если x^* – изолированная точка локального минимума, а точка x_0 – такая, что связная компонента множества $\left\{x : f(x^*) \le f(x) \le f(x_0)\right\}$, содержащая x_0 и x^*, не содержит, кроме x^*, других точек z, в которых семейство $G_f(z)$ линейно зависимо, то последовательность $\{x_k\}_{k=0}^{\infty}$ сходится к точке x^*.

Теорема 4 определяет достаточные условия, при которых $r_\mu(\alpha)$-алгоритм сходится к локальному минимуму почти-дифференцируемой кусочно-гладкой функции $f(x)$. Однако, эти условия слишком сильные и не выполняются даже для кусочно-гладких выпуклых функций. Наиболее типичная ситуация, когда это происходит, связана с нарушением линейной независимости множества почти-градиентов $G_f(x)$.

3. О зацикливании $r_\mu(\alpha)$-алгоритма. В работе [3] построена кусочно-линейная выпуклая функция от двух переменных, для которой возможно «зацикливание» $r_\mu(\alpha)$-алгоритма. Если коэффициент растяжения пространства $\alpha = 3,$ то примером такой функции является выпуклая функция

$$f(x_1,x_2) = \max_{i=1,\dots,8}\left\{ f_i(x_1,x_2)\right\} = \max\left\{\max_{i=1,\dots,4}\left\{ f_i(x_1,x_2)\right\}, \max_{i=5,\dots,8}\left\{ f_i(x_1,x_2)\right\}\right\}, \quad (4)$$

где линейные функции $f_i = f_i(x_1,x_2),\ i=1,\dots,8,$ имеют следующий вид:

$$f_1 = -10x_1 - x_2 - 1,\ \ f_2 = 6x_1 - 9x_2 - 9,\ \ f_3 = 10x_1 - x_2 - 1,\ \ f_4 = -6x_1 - 9x_2 - 9,$$

$$f_5 = 10x_1 + x_2 - 1,\ \ f_6 = -6x_1 + 9x_2 - 9,\ \ f_7 = -10x_1 + x_2 - 1,\ \ f_8 = 6x_1 + 9x_2 - 9.$$

Ее поверхности уровня показаны на рис. 1.

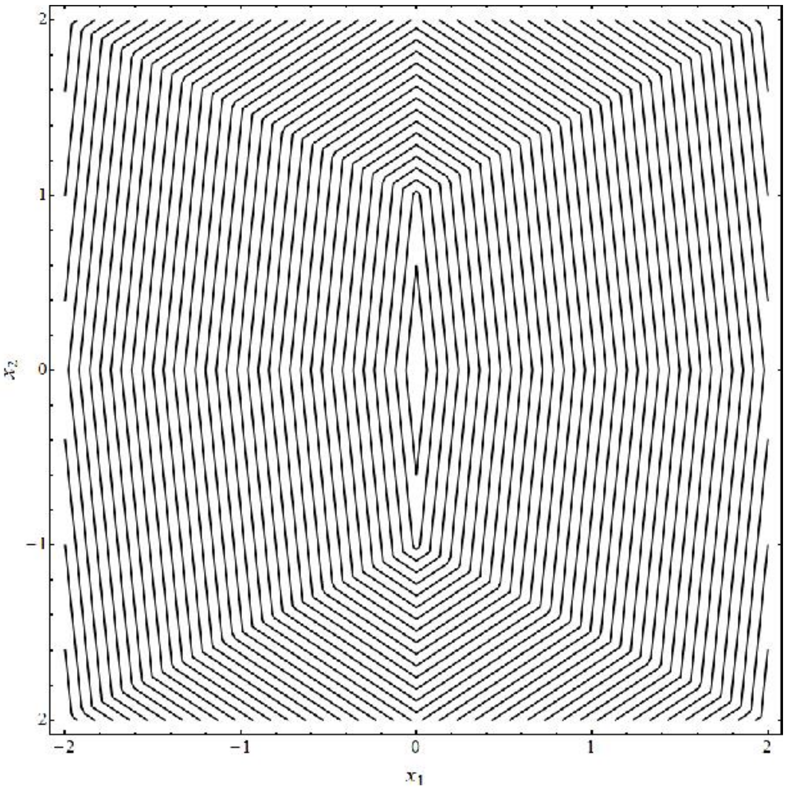

РИС. 1. Поверхности уровня функции (4)

Минимальное значение функция (4) принимает в точке $x^* = (0,0)^T$ и $f^* = -1.$ Кроме оптимальной, для нее имеются точки $z_1 = (0,1)^T$ и $z_2 = (0,-1)^T$: $f(z_1) = f(z_2) = 0,$ в которых нарушено условие линейной независимости семейств почти-градиентов $G_f(z_1)$ и $G_f(z_2)$.

Множество $G_f(z_1)$ состоит из четырех почти-градиентов: $g_1 = (10,1)^T$; $g_2 = (-6,9)^T$; $g_3 = (-10,1)^T$; $g_4 = (6,9)^T$. Им сответствуют градиенты линейных функций f_5, f_6, f_7 и f_8, значения которых в точке $z_1 = (0,1)^T$ равны нулю. Множество $G_f(z_1) = \{g_1, g_2, g_3, g_4\}$ показано на рис. 2, где выпуклая оболочка почти-градиентов закрашена в синий цвет. Любой из анти-почти-градиентов в точке z_1 не является направлением убывания функции (4), так как существует антипочти-градиент, образующий с ним тупой угол.

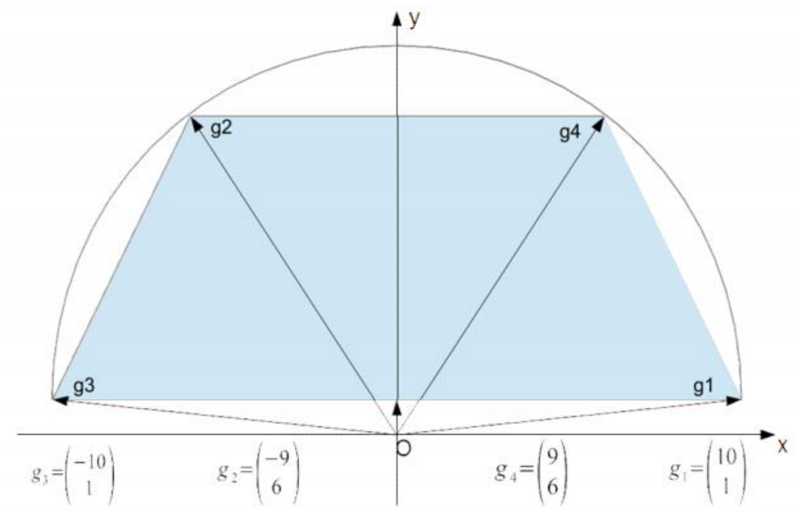

РИС. 2. Множество $G_f(z_1) = \{g_1, g_2, g_3, g_4\}$ для функции (4)

Множество $G_f(z_2)$ состоит из четырех таких почти-градиентов: $g_1 = (-10,-1)^T$; $g_2 = (6,-9)^T$; $g_3 = (10,-1)^T$; $g_4 = (-6,-9)^T$, которым соответствуют градиенты линейных функций f_1, f_2, f_3 и f_4. Множеству $G_f(z_2)$ будет соответствовать рисунок, который является «зеркальным» отражением рис. 2. относительно оси Ox.

Пусть $x_0 = z_1$ – стартовая точка для $r_\mu(\alpha)$-алгоритма, $\alpha = 3$, $B_0 = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$ и $g_0 = g_1$ – почти-градиент в точке x_0. Тогда в качестве последовательных почти-градиентов по итерациям выберем такую последовательность $\{g_1, g_2\}$, $\{g_2', g_3'\}$, $\{g_3'', g_4''\}$, $\{g_4''', g_1'''\}$. Здесь «штрихами» обозначены почти-градиенты в преобразованном пространстве: одним – после первого растяжения, двумя – после двух и т. д. Такая последовательность удовлетворяет требованиям $r_\mu(\alpha)$-алгоритма при $\alpha = 3$ и произвольном $\mu \geq 0$ [3]. При этом шаг на каждой итерации будет нулевым, т. е. сдвига с точки x_0 происходить не будет. Картина почти-градиентов после четырех итераций останется такой же, как и вначале, только длины всех векторов уменьшатся в $\alpha^2 = 9$ раз. Это связано с тем, что растяжение пространства реализуется по взаимно-ортогогальным направлениям.

Если этот цикл повторять, то $r_\mu(\alpha)$-алгоритм с коэффициентом растяжения пространства $\alpha = 3$ не сможет выбраться из точки $x_0 = (0,1)$, хотя она не является оптимальной для функции (4). Поэтому, точки z_1 и z_2 могут служить «ловушками» для минимизирующей последовательности $r_\mu(\alpha)$-алгоритма.

4. $r_\mu(\alpha)$-алгоритм. Далее рассмотрим $r_0(\alpha)$-алгоритм – модификацию $r_\mu(\alpha)$-алгоритма для выпуклой функции $f(x)$, $x \in E^n$, которая для функции (4) позволяет преодолевать стартовые точки $x_0 = z_1$ и $x_0 = z_2$. Выпуклая функция $f(x)$ – почти-дифференцируемая функция (см. теорему 1). Будем предполагать, что множество $G_f(x)$ состоит из конечного количества почти-градиентов функции $f(x)$ в точке x.

Пусть $\alpha > 1$, x_0 – начальная точка; $g_f(x_0)$ – почти-градиент функции $f(x)$ в точке x_0; $B_0 = I_n$ – единичная $n \times n$-матрица. $r_0(\alpha)$-алгоритм строит последовательности векторов $\{x_k\}_{k=0}^{\infty}$ и матриц $\{B_k\}_{k=0}^{\infty}$ по следующему правилу:

1) вычислим очередную точку

$$x_{k+1} = x_k - h_k B_k g_{\varphi_k}(y_k), \tag{5}$$

где $g_{\varphi_k}(y_k) = B_k^T g_f(x_k)$, а величина шага h_k выбирается по формуле

а) $h_k = \arg\min_{h \geq 0} f\left(x_k - hB_k g_{\varphi_k}(y_k)\right)$.

Если h_k определяется неоднозначно, то выбираем его наименьшее значение;

2) вычислим $G_f(x_{k+1}) = \{g_i\}_{i=1}^{m_k}$ – множество почти-градиентов функции $f(x)$ в точке x_{k+1}. Если $m_k = 1$ и $g_1 = 0$, то $x^* = x_{k+1}$ и останов алгоритма. Иначе выбираем $g = g_j$, где индекс $j : 1 \leq j \leq m_k$ такой, на котором достигается минимум скалярного произведения $\left(B_k^T g_j, g_{\varphi_k}(y_k)\right)$. Если таких векторов больше чем один, то выбираем произвольный из них;

3) пересчитаем матрицу (преобразования пространства)

$$B_{k+1} = B_k + \left(\frac{1}{\alpha} - 1\right)\left(B_k \xi_k\right)\xi_k^T, \tag{6}$$

где

$$\xi_k = \frac{B_k^T r_k}{\| B_k^T r_k \|}, \quad r_k = g - g_f(x_k).$$

4) переходим к очередной итерации с x_{k+1}, B_{k+1} и $g_f(x_{k+1}) = g$.

Здесь h_k – величина шага, такая что $h_k \geq 0$, $g_f(x_k)$ и $g_f(x_{k+1})$ – почти-градиенты функции $f(x)$ в точках x_k и x_{k+1}.

Теорема 5. Пусть $f(x)$ – выпуклая функция, такая, что $\lim_{\|x\|\to\infty} f(x) = +\infty$, и последовательность $\{x_k\}_{k=0}^{\infty}$, генерируемая $r_0(\alpha)$-алгоритмом, удовлетворяет условию

$$\lim_{k\to+\infty} \|x_{k+1} - x_k\| = 0. \tag{7}$$

Если x^* – изолированная точка глобального минимума, а точка x_0 – такая, что выпуклое множество $\{x : f(x^*) \leq f(x) \leq f(x_0)\}$, содержащее x_0 и x^*, не содержит, кроме x^*, других точек z, в которых семейство $G_f(z)$ линейно зависимо, то последовательность $\{x_k\}_{k=0}^{\infty}$ сходится к точке x^*.

Доказательство. Для доказательства достаточно показать, что: (i) теорема 5 – это частный случай теоремы 4; (ii) $r_0(\alpha)$-алгоритм – это частный случай $r_\mu(\alpha)$-алгоритма.

Справедливость (i) следует из того, что выпуклая функция $f(x)$ – это почти-дифференцируемая (см. теорему 1). Поэтому для нее изолированную точку локального минимума (теорема 4) можно заменить на изолированную точку глобального минимума, а связную компоненту множества, где предполагается наличие точки локального минимума почти-дифференцируемой функции (теорема 4), можно заменить на выпуклое множество, в котором содержится единственная точка глобального минимума функции $f(x)$.

Справедливость (ii) следует из того, что условие (п. 1 и условия п. 3) в $r_0(\alpha)$-алгоритме являются одним из вариантов условий раздела 2 п. a) и b) в $r_\mu(\alpha)$-алгоритме и выполняются при всех значениях $\mu \geq 0$. На самом деле, из раздела 4, п. 1 условие a) означает, что неотрицательную величину шага h_k мы выбираем на отрезке $[0, h_k]$, где функция $\varphi_k(h) = f(x_{k+1}(h))$ не возрастает. Оно же в сочетании с условиями пункта 3) гарантирует выбор шага из условия минимума функции по направлению антисубградиента, а $\lim_{\|x\|\to\infty} f(x) = +\infty$ гарантирует, что такой минимум всегда достижим и для него выполняется условие $\left(B_k^T g_j, g_{\varphi_k}(y_k)\right) \le 0.$ Последнее можно заменить условием $\left(B_k^T g_j, g_{\varphi_k}(y_k)\right) \le \mu,$ справедливым для любого параметра $\mu \ge 0.$ Доказательство завершено.

Теорему 5 можно усилить, разрешив множеству $\left\{x : f(x^*) \le f(x) \le f(x_0)\right\}$ содержать определенного вида точки с линейно зависимыми почти-градиентами. Это подтверждает

Теорема 6. Если $\alpha = 3,$ то $r_0(\alpha)$-алгоритм не будет зацикливаться в точках $x_0 = z_1$ и $x_0 = z_2$ для функции (4).

Доказательство. Пусть стартовая точка $x_0 = z_2.$ Множество почти-градиентов функции (4) в точке x_0 состоит из четырех векторов: $g_1 = (10,1)^T$, $g_2 = (-6,9)^T$, $g_3 = (-10,1)^T$, $g_4 = (6,9)^T$ (см. рис. 2). Если в качестве $g_f(x_0)$ выбран почти-градиент $g_1,$ то $h_0 = 0,$ так как $(g_1, g_3) = -99 < 0$ (т. е. в точке x_0 существует такой почти-градиент с которым g_1 составляет тупой угол). Следовательно, очередная точка $x_1 = x_0$ и множество почти-градиентов в ней состоит из тех же четырех векторов. Согласно п. 3) в качестве почти-градиента в точке x_1 будет выбран вектор $g = g_3,$ так как $(g_1, g_3) = -99$ меньше, чем $(g_1, g_2) = -51$ и $(g_1, g_4) = 69.$ Следовательно, первое растяжение пространства будет производиться в направлении разности почти-градиентов g_3 и $g_1,$ в результате чего на первой итерации $r_0(3)$-алгоритма получаем направление растяжения $\xi_1 = (-1, 0)^T$ и матрицу $B_1 = \begin{vmatrix} 1/3 & 0 \\ 0 & 1 \end{vmatrix}.$

Почти-градиент функции $\varphi_1(y) = f(B_1 y)$ в точке $y_1 = B_1^{-1} x_1$ вычисляется по формуле $g_{\varphi_1}(y_1) = B_1^T g_f(x_1),$ где $g_f(x_1)$ – почти-градиент функции $f(x)$ в точке x_1. Следовательно, образами почти-градиентов g_1, g_2, g_3 и g_4 в преобразованном пространстве переменных $Y_1 = B_1^{-1} X$ будут такие почти-градиенты: $g_1^{'} = (10/3, 1)^T$, $g_2^{'} = (-2, 9)^T$, $g_3^{'} = (-10/3, 1)^T$ и $g_4^{'} = (2, 9)^T$. Для функции $\varphi_1(y)$ шаг градиентного спуска определяется почти-градиентом $g_3^{'}$ и, учитывая, что $(g_3^{'}, g_1^{'}) = -91/9 < 0,$ для второй итерации $r_0(3)$-алгоритма величина шага также будет нулевой, т. е. $h_1 = 0.$ Следовательно, точка $x_2 = x_0$ и согласно п. 3) в качестве почти-градиента в точке x_2 будет выбран $g = g_1,$ так как $(g_3^{'}, g_1^{'}) = -91/9$ меньше, чем $(g_3^{'}, g_2^{'}) = 47/3$ и $(g_3^{'}, g_4^{'}) = 7/3.$ Следовательно, на второй итерации растяжение пространства будет производиться в направлении разности почти-градиентов $g_1^{'}$ и $g_3^{'}$, в результате чего получаем $\xi_2 = (1,0)^T$ и матрицу $B_2 = \begin{vmatrix} 1/9 & 0 \\ 0 & 1 \end{vmatrix}$.

Образами почти-градиентов g_1, g_2, g_3 и g_4 в преобразованном пространстве $Y_2 = B_2^{-1}X$ будут почти-градиенты: $g_1^{''} = (10/9,1)^T$, $g_2^{''} = (-2/3,9)^T$, $g_3^{''} = (-10/9,1)^T$ и $g_4^{''} = (2/3,9)^T$. Для функции $\varphi_2(y)$ шаг градиентного спуска определяется почти-градиентом $g_1^{''}$. Учитывая, что $(g_1^{''}, g_3^{''}) = -19/81 < 0,$ для третьей итерации $r_0(3)$-алгоритма величина шага $h_2 = 0$. Следовательно, точка $x_3 = x_0$ и в ней будет выбран $g = g_3$, так как $(g_1^{''}, g_3^{''}) = -19/81$, а $(g_1^{''}, g_2^{'}) = 223/27$ и $(g_1^{''}, g_4^{''}) = 263/27$. Следовательно, на третьей итерации растяжение пространства будет производиться в направлении разности почти-градиентов $g_3^{''}$ и $g_1^{''}$, в результате чего $\xi_3 = (-1,0)^T$ и $B_3 = \begin{vmatrix} 1/27 & 0 \\ 0 & 1 \end{vmatrix}$.

Образами почти-градиентов g_1, g_2, g_3 и g_4 в преобразованном пространстве $Y_3 = B_3^{-1}X$ будут почти-градиенты $g_1^{'''} = (10/27,1)^T$, $g_2^{'''} = (-2/9,9)^T$, $g_3^{'''} = (-10/27,1)^T$ и $g_4^{'''} = (2/9,9)^T$. Для функции $\varphi_3(y)$ шаг градиентного спуска определяет почти-градиент $g_3^{'''}$ и, учитывая, что, $(g_3^{'''}, g_1^{'''}) = 629/729 > 0,$ $(g_3^{'''}, g_2^{'''}) = 2261/243 > 0$ и $(g_3^{'''}, g_2^{'''}) = 2221/243 > 0$, то для четвертой итерации $r_0(3)$-алгоритма величина шага станет ненулевой, т. е. $h_3 > 0$. Это означает, что алгоритм перейдет в точку x_4 с меньшим значением функции (4).

По аналогичной схеме можно показать, что за три итерации $r_0(3)$-алгоритм найдет направления убывания функции из стартовой точки $x_0 = z_2$ (симметричный случай). Теорема доказана.

Заключение. В $r_{\mu}(\alpha)$-алгоритме за приближенный поиск минимума функции по направлению антисубградиента в преобразованном пространстве переменных отвечает параметр μ. Он обеспечивает выполнение неравенства $h_k \le h_k^*$, где h_k^* соответствует минимуму функции. В практических вариантах r-алгоритмов шаг h_k выбирается так, чтобы выполнялось неравенство $h_k > h_k^*$. Для них «приближенный» поиск минимума функции направлен на уменьшение общего количества вычислений значений функции и ее субградиента и делается так, чтобы на одну итерацию алгоритма приходилось в среднем два-три таких вычисления. Одной из эффективных реализаций такой стратегии является $r(\alpha)$-алгоритм с адаптивной регулировкой шага, который преодолевает «ловушки» для минимизирующей последовательности $r_{\mu}(\alpha)$-алгоритма.

Работа выполнена при поддержке НАНУ, проект № 0114U001055.

П.І. Стецюк, А.В. Івлічев, А.О. Іщенко

ПРО ЗБІЖНІСТЬ $r_\mu(\alpha)$-АЛГОРИТМА

Наводиться опис $r_\mu(\alpha)$-алгоритма для мінімізації майже-диференційовної функції. Розглядається кусочно-лінійна опукла функція, для якої дві точки з лінійно залежними майже-градієнтами можуть служити «пастками» для $r_\mu(\alpha)$-алгоритма. Для мінімізації опуклої функції запропоновано $r_\mu(\alpha)$-алгоритм і показано, що його не можна «зациклити» в точках-пастках розглянутої кусочно-лінійної опуклої функції.

P.I. Stetsyuk, A.V. Ivlichev, A.A. Ischenko

ON THE CONVERGENCE OF THE $r_\mu(\alpha)$-ALGORITHM

A description of the $r_\mu(\alpha)$-algorithm for minimizing the near-differentiable function is given. We consider piecewise-linear convex function, for which two points with linearly dependent almost-gradients can serve as the “traps” for the $r_\mu(\alpha)$-algorithm. The $r_\mu(\alpha)$-algorithm for minimizing a convex function is proposed. It is shown that $r_\mu(\alpha)$-algorithm can not be “looped” at point-traps for the piecewise-linear convex function considered.

1. *Шор Н.З.* Методы минимизации недифференцируемых функций и их приложения. – Киев: Наук. думка, 1979. – 199 с.
2. *Шор Н.З.* Исследования сходимости метода градиентного типа с растяжением пространства в направлении разности двух последовательных субградиентов // Кибернетика. – 1975. – № 4. – С. 48 – 53.
3. *Стецюк П.И.* К вопросу сходимости r-алгоритмов // Кибернетика и системный анализ – 1995. – № 6. – С. 173 – 177.
4. *Шор Н.З.* О классе почти-дифференцируемых функций и одном методе минимизации функций этого класса // Кибернетика. – 1972. – № 4. – С. 65 – 70.

Получено 20.01.2015

Об авторах:

Стецюк Петр Иванович,
доктор физико-математических наук,
заведующий отделом Института кибернетики имени В.М. Глушкова НАН Украины,
E-mail: stetsyukp@gmail.com

Ивличев Андрей Владимирович,
ведущий инженер-программист
Института кибернетики имени В.М. Глушкова НАН Украины,
E-mail: ivlichev1990@gmail.com

Ищенко Андрей Александрович,
студент 5 курса Киевского отделения Московского физико-технического института.
E-mail: andrey.ischenko@gmail.com