КОМПЬЮТЕРНАЯ МАТЕМАТИКА

Теория и методы оптимизации

УДК 519.8

Н.Г. ЖУРБЕНКО, А.П. ЛИХОВИД

К ЧИСЛЕННОЙ ЭФФЕКТИВНОСТИ ОДНОЙ МОДИФИКАЦИИ *R*-АЛГОРИТМА[1]

Рассматривается одна модификация r-алгоритма – алгоритма минимизации с использованием операции растяжения пространства в направлении разности двух последовательных субградиентов. В отличие от r-алгоритма, значения коэффициентов растяжения в предложенной модификации рассчитываются в процессе работы алгоритма. Алгоритм может использоваться с постоянным шагом. Приводятся результаты исследования численной эффективности алгоритма.

Рассматривается задача безусловной минимизации субдифференцируемой функции $f(x)$ в R^n. Предполагается, что в произвольной точке $x \in R^n$ для функции $f(x)$ определен субградиент – вектор $g(x) \in R^n$. Обозначим $\partial f(x)$ – множество субградиентов функции $f(x)$ в точке x. Эффективным алгоритмом минимизации субдифференцируемой функции является *r*-алгоритм – алгоритм, использующий операцию растяжения пространства в направлении разности двух последовательных субградиентов [1], [2]. *r*-алгоритм относится к классу субградиентных алгоритмов с преобразованием пространства.

Численная схема r-алгоритма. В r-алгоритме в качестве оператора преобразования пространства используется оператор растяжения пространства по направлению [2]: $R_\alpha(\eta) = (\alpha - 1)\eta\eta^T + I$. $\eta \in R^n$, $\alpha -$ направление и коэффициент растяжения пространства, $\|\eta\| = 1$, $\alpha \ge 1$. Заметим, что $R_\alpha^{-1}(\eta) = R_\beta(\eta)$, $\beta = 1/\alpha$. $R_\alpha^*(\eta) = R_\alpha(\eta)$ ($R_\alpha(\eta)$ – самосопряженный оператор).

Вычислительная схема r-алгоритма применительно к задаче отыскания безусловного минимума функции $f(x)$ состоит в следующем.

[1]Работа выполнена при частичной поддержке Volkswagen Foundation (грант No 90 306 – Н. Г. Журбенко)).

0-ый шаг алгоритма.

Выбираем начальное приближение $x_0 \in X_0$ (X_0 – исходное пространство переменных) и невырожденное линейное преобразование A_0. пространства X_0. $Y_0 = A_0 X_0$. – преобразованное пространство, $X_0 = B_0 Y_0$, где $B_0 = A^{-1}$. Вычисляем:

1) $g(x_0) \in \partial f(x_0)$ (субградиент в точке x_0);

2) $g_0^* = B_0^* g_0$ – субградиент в преобразованном пространстве $Y_0 = A_0 X_0$.

Переходим к первому шагу алгоритма. Пусть на шаге k алгоритма $(k = 0,1,2,\ldots)$ получены определенные значения векторов x_k, g_k^* (субградиент в преобразованном пространстве) и матрицы B_k (обратная матрица преобразования пространства).

$(k+1)$ **-вый шаг алгоритма** $(k = 0,1,2,\ldots)$.

Вычисляем:

1) h_{k+1} – шаговый множитель, $h_{k+1} \ge 0$;

2) $x_{k+1} = x_k - h_{k+1} B_k g_k^* / \| g_k^* \|$; вычисление x_{k+1} соответствует субградиентный шагу в преобразованном пространстве Y_k по нормированному субградиенту: $y_{k+1} = y_k - h_{k+1} g_k^* / \| g_k^* \|$;

3) $g(x_{k+1}) \in \partial f(x_k)$; (субградиент в точке x_{k+1} исходного пространства X_0);

4) $\tilde{g}_{k+1}^* = B_k^* g(x_{k+1})$ (субградиент в преобразованном пространстве $Y_k = A_k X_0$);

5) $\text{¦}_{k+1} = (\tilde{g}_{k+1}^* - g_k^*) / \| \tilde{g}_{k+1}^* - g_k^* \|$ (направление растяжения пространства Y_k);

6) $\alpha_{k+1} \ge 1$, $\beta_{k+1} = 1/\alpha_{k+1}$ (α_k коэффициент растяжения пространства Y_k);

7) $B_{k+1} = B_k R_{{}_{k+1}}(\text{¦}_k)$, (обратный оператор преобразования пространства $Y_{k+1} = A_{k+1} X_0$);

8) $g_{k+1}^* = R_{{}_{k+1}}(\text{¦}_{k+1}) \tilde{g}_{k+1}$ $(g_{k+1}^* = B_{k+1}^* g(x_k)$. (субградиент в преобразованном пространстве $Y_{k+1} = A_{k+1} X$).

Переходим к $(k+2)$-му шагу алгоритма, или прекращаем работу при выполнении критерия останова.

В известных вариантах r-алгоритма выбор коэффициентов растяжения пространства и шаговых множителей определяется следующим образом. Значения коэффициентов растяжения пространства α_k выбираются одинаковыми на всех итерациях: $\alpha_k = \alpha > 1$. α – параметр r-алгоритма. На практике рекомендуется значение α выбирать порядка 2.0. Величина шагового множителя определяется выбранной процедурой минимизации (процедурой «спуска») по направлению $p_k = -B_k g_k^* / \| g_k^* \|$. Применяемые процедуры являются достаточно грубой реализацией алгоритма локализации минимума по направлению p_k. Основное требование при этом – это выполнение условия: $(p_k, g(x_{k+1})) \ge 0$ (это условие обеспечивает, что $\| \tilde{g}_{k+1}^* - g_k^* \| > 0$).

Наиболее часто используется следующая процедура регулировки шаговых множителей. Пусть на $(k+1)$-ой итерации помимо точки x_k, вектора спуска p_k определено значение «пробного шага» $\tilde{h}_k > 0.$ Значение $\tilde{h}_0$ – параметр r-алгоритма – «величина начального шага».

Пусть заданы числа $0 < q_1 \le 1,$ $q_2 \ge 1$ и целое число $L \ge 2.$ Эти величины являются параметрами (константами) алгоритма. Они будут определять регулировку величин пробного шага. Параметр q_1 используется для уменьшения пробного шага, а q_2 – для его увеличения.

Положим $z_0 = x_k$. Вычисляем: $z_i = z_{i-1} + \tilde{h}_k p_k;$ $g_i = g(z_i) \in \partial f(z_i),$ $i = 1, 2, \ldots,$ до тех пор, пока при некотором $i = l$ выполнится неравенство $(g_i, p_k) \ge 0.$ Тогда $x_{k+1} = z_l$. Если $l = 1,$ то $\tilde{h}_{k+1} := q_1 \tilde{h}_k$.

Наиболее часто используются следующие значения параметров регулировки пробного шага $q_1 = 0.9,$ $q_2 = 1.1,$ $L = 3.$

Вычислительная схема $r(\pm)$-алгоритмов. Вычислительная схема $r(\sigma)$-алгоритмов соответствует приведенной схеме r-алгоритма. Отличие состоит лишь в следующем. Вместо оператора растяжения $R_\alpha(\eta)$ используется следующий оператор:

$$\tilde{R}_\sigma(\tilde{\eta}) = \sigma \tilde{\eta} \tilde{\eta}^T + I. \tag{1}$$

Здесь $\tilde{\eta}$ – вектор в R^n, σ – нормирующий множитель, $\sigma \in R^1$, $\sigma > 0.$ В отличие от оператора $R_\alpha(\eta),$ вектор $\tilde{\eta}$ не нормирован, т. е. выполнение условия $\| \tilde{\eta} \| = 1$ не требуется. Различные варианты алгоритма будут определяться выбором нормирующего множителя $\sigma.$ Далее будут рассмотрены несколько вариантов такого выбора.

Остановимся на простейших свойствах оператора $\tilde{R}_\sigma(\tilde{\eta})$ и его связи с оператором $R_\alpha(\eta).$

Обратный к $R_\alpha(\eta)$ оператор $\tilde{R}_\sigma^{-1}(\tilde{\eta})$ определяется следующим образом:

$$\tilde{R}_\sigma^{-1}(\tilde{\eta}) = -\left(\sigma / (1 + \sigma \tilde{\eta}^2)\right) \tilde{\eta} \tilde{\eta}^T + I.$$

Очевидно, что $\tilde{R}_\sigma(0) = I$ (при этом значение нормирующего множителя не имеет значения). Пусть $\| \tilde{\eta} \| \ne 0$ и $\eta = \tilde{\eta} / \| \tilde{\eta} \|.$ Тогда $\tilde{R}_\sigma(\tilde{\eta}) = R_\alpha(\eta),$ где

$$\alpha = 1 + \sigma \| \tilde{\eta} \|^2 . \tag{2}$$

Таким образом, если $\tilde{\eta} \neq 0,$ то оператор $\tilde{R}_{\sigma}(\tilde{\eta})$ – оператор растяжения по направлению $\tilde{\eta} / \| \tilde{\eta} \|$ с указанным значением коэффициента растяжения (2). Формально, для $\tilde{\eta} = 0$ можно считать, что коэффициент растяжения равен 1, так как в этом случае оператор $\tilde{R}_{\sigma}(\tilde{\eta})$ равен I.

Значение нормирующего множителя σ будет определяться на основании двух последовательных субградиентов $\tilde{g}^{*}_{k+1}, g^{*}_{k}$ в преобразованном пространстве. Естественным требованием на функцию $\sigma(g_1, g_2)$ будет выполнение следующего условия (условия «однородности»): $\sigma(\mu g_1, \mu g_2) = \sigma(g_1, g_2) / \mu^2,$ где $\mu \in R^1, \mu > 0.$ Это условие обеспечивает независимость работы алгоритма от множителя на целевую функцию.

Легко видеть, что алгоритм $r(\sigma_0):$ $\sigma_0(g_1, g_2) = 1 / \| g_2 - g_1 \|^2$ фактически является r-алгоритмом с коэффициентом растяжения равным 2. Любопытно отметить, что именно это значение рекомендуется на практике использования r-алгоритма. Однако в r-алгоритме (и $r(\sigma_0)$-алгоритме) необходимо предусмотреть выполнение условия $\| g_2 - g_1 \| > 0.$ Это условие обеспечивается вышеприведенной процедурой регулировки шаговых множителей (процедурой приближенного спуска по антисубградиенту).

Выбирая различные нормирующие множители $\sigma,$ мы будем получать различные алгоритмы рассматриваемого класса. Проблема рационального выбора этих множителей требует дальнейшего исследования. Приведем примеры нормирующих множителей (помимо приведенного $r(\sigma_0)$).

1. $r(\pm_1)$-алгоритм: $\sigma_1(g_1, g_2) = 1 / \| Nr\{g_1, g_2\} \|^2,$ где $Nr\{g_1, g_2\}$ – кратчайший вектор выпуклой оболочки векторов g_1, g_2.

2. $r(\sigma_2)$-алгоритм: $\sigma_2(g_1, g_2) = 1 / (\| g_2 \| * \| g_1 \|).$

3. $r(\sigma_3)$-алгоритм: $\pm_3(g_1, g_2) = 1 / \max\{\| g_1 \|^2, \| g_2 \|^2\}.$

Для алгоритмов $r(\sigma_1),$ $r(\sigma_2)$ значения коэффициентов растяжения не ограничены: $1 \leq \alpha_{k+1} \leq \infty.$

Для алгоритма $r(\sigma_3)$ значения коэффициентов ограничены: $1 \leq \alpha_{k+1} \leq 5.$

В данной работе приводятся результаты исследования численной эффективности алгоритма $r(\sigma_3)$ (для $r(\sigma_1),$ $r(\sigma_2)$ такие результаты приведены в работах [3, 4]).

Численная эффективность $r(\sigma_3)$-алгоритма. Приведем результаты численных исследований эффективности алгоритма $r(\sigma_3)$ в сравнении с r-алгоритмом ($r(\sigma_0)$-алгоритмом).

Рассматривались три тестовые задачи.

Первая и вторая тестовые задачи являлись задачами минимизации следующих двух функций: $f_1(x) = \sum_{i=1}^{n} \rho_n^{i-1} x_i^2$, $f_2(x) = \sum_{i=1}^{n} \rho_n^{i-1} | x_i |$, где параметр ρ_n выбирался в зависимости от размерности задачи n по формуле $\rho_n = 10^{6/(n-1)}$. Таким образом, степень вытянутости линий уровня («овражности») функций определяется значением параметра $\rho_n^{n-1} = 10^6$, она одинакова для всех функций независимо от числа переменных. Начальная точка $x_i = 1.0,\ i = 1, 2, ...n.$ Критерий останова: $f_k \le 10^{-6}$, где f_k – значение функции на итерации останова k.

Третья тестовая задача – это задача минимизации функции Лемарешаля [5]. Задача состоит в минимизации существенно овражной выпуклой негладкой функции от 10 переменных:

$$f_3(x) = \max_{1 \le k \le 5} f_k(x) \to \min_x ,$$

где $f_k(x) = x^T A_k(x) - b_k^T(x)$, A_k – симметричные 10×10-матрицы, такие, что $A_{kij} = e^{i/j} \cos(ij) \sin k$, если $i < j$, и $A_{kii} = i \left|\sin k\right| / 10 + \sum_{j \ne i} \left|A_{kij}\right|$, а компоненты векторов b_k определяются $b_{ki} = e^{i/k} \sin(ik)$. Начальное приближение – это точка $x_0 = (1, \ldots, 1)^T \in R^{10}$, в которой реализуется значение функции $f(x_0) = 5337.06643$. Функция имеет единственную точку минимума. Минимальное значение функции $f_3^*(x)$ определяется его достаточно точным приближением $f_{\min}^* = -0.841408334596$ (с точностью 10^{-12}, 12 цифр после точки).

Результаты решения тестовых задач приведены в таблицах 1, 2, 3 соответственно, где приняты следующие обозначения:

n – размерность пространства переменных;

k – номер итерации, на которой алгоритм прекратил работу;

kg – количество вычислений обобщенного градиента;

$\alpha_{\max}$ – максимальное значение коэффициента растяжения;

α_{avrg} – среднее значение коэффициента растяжения;

r – результаты работы r-алгоритма с параметрами $q_1 = q_2 = 1$.

В алгоритмах, отмеченных символом “ * ”, используется постоянная величина шага (в преобразованном пространстве).

В алгоритмах, не отмеченных символом “ * ”, используется вышеописанная регулировка шаговых множителей r-алгоритма.

ТАБЛИЦА 1. Минимизация функции $f_1(x)$

Параметры / Алгоритм	n	k	k_g	$\alpha_{\max}$	α_{avrg}
$r(\sigma_3)$	100	926	1136	4.90179	2.79076
$r^*(\sigma_3)$	100	999	1000	4.80209	2.547
r	100	1197	1382	2.00	2.00
$r(\sigma_3)$	300	2737	3301	4.65948	2.79623
$r^*(\sigma_3)$	300	2961	2962	4.76045	2.59546
r	300	3416	3898	2.00	2.00
$r(\sigma_3)$	1000	8784	9690	4.92057	2.73933
$r^*(\sigma_3)$	1000	9271	9272	4.62152	2.62127
r	1000	10926	11930	2.00	2.00

ТАБЛИЦА 2. Минимизация функции $f_2(x)$

Параметры / Алгоритм	n	k	k_g	$\alpha_{\max}$	α_{avrg}
$r(\sigma_3)$	100	2332	2343	3.47955	2.65213
$r^*(\sigma_3)$	100	2330	2331	3.26849	2.65078
r	100	3258	3267	2.00	2.00
$r(\sigma_3)$	300	7186	7197	3.25877	2.64139
$r^*(\sigma_3)$	300	7198	7199	3.13601	2.6406
r	300	10116	10123	2.00	2.00
$r(\sigma_3)$	1000	24651	24673	3.20645	2.63744
$r^*(\sigma_3)$	1000	28215	28216	2.43678	2.32715
r	1000	35186	35199	2.00	2.00

ТАБЛИЦА 3. Минимизация функции $f_3(x)$

Параметры / Алгоритм	k	k_g	$\alpha_{\max}$	α_{avrg}	$f_3^*(x)$
$r(\sigma_3)$	206	257	4.87506	2.73264	– 0.841408334593403
$r^*(\sigma_3)$	285	286	4.57635	2.46481	– 0.841408334596392
r	343	388	2.00	2.00	– 0.841408334596395

Выводы. Предложенный субградиентный алгоритм с преобразованием пространства можно рассматривать как модификацию r-алгоритма. Величины коэффициентов растяжения пространства на итерациях $r(\sigma_3)$-алгоритма не постоянны, они вычисляются в процессе его работы. Алгоритм не требуют использования процедуры одномерного спуска по направлению. Он может использоваться с постоянным шаговым множителем.

Численная схема $r^*(\sigma_3)$-алгоритма (алгоритм с постоянным шаговым множителем в преобразованном пространстве) существенно проще схемы r-алгоритма. В этом алгоритме единственным существенным параметром является значение шагового множителя. При этом значения управляющих параметров процедуры одномерного спуска (q_1, q_2, L) не используются. В r-алгоритме эти параметры и значение коэффициента растяжения пространства выбираются из эвристических соображений.

Численные эксперименты показали достаточно высокую эффективность $r(\sigma_3)$-алгоритма. Его эффективность не уступает эффективности r-алгоритма (с параметрами $q_1 = q_2 = 1$).

М.Г. Журбенко, О.П. Лиховид

ДО ЧИСЕЛЬНОЇ ЕФЕКТИВНОСТІ ОДНІЄЇ МОДИФІКАЦІЇ *R*-АЛГОРИТМУ

Розглядається одна модифікація *r*-алгоритму – алгоритму мінімізації з використанням операції розтягування простору в напрямку різниці двох послідовних субградієнтів. На відміну від *r*-алгоритму, значення коефіцієнтів розтягування в запропонованій модифікації розраховуються в процесі роботи алгоритму. Алгоритм може використовуватися з постійним кроком. Наводяться результати дослідження чисельної ефективності алгоритму.

M.G. Zhurbenko, O.P. Lykhovyd

ON THE NUMERICAL EFFICIENCY OF A MODIFICATION OF *R*-ALGORITHM

We consider a modification of the r-algorithm, the minimization algorithm using the operation of space dilation in the direction of the difference of two successive subgradients. In contrast to the *r*-algorithm, the proposed modification of the algorithm calculate the values of dilation coefficients . The algorithm can be used with a constant step. The results of the study of the numerical efficiency of the algorithm are given.

Список литературы

1. Шор Н.З., Журбенко Н.Г. Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов. *Кибернетика*. 1971. № 3. С. 51 – 59.
2. Шор Н.З. Методы минимизации недифференцируемых функций и их приложения. К.: Наук. думка, 1979. 199 с.
3. Журбенко Н.Г., Чумаков Б.М. Программное управление коэффициентами растяжения *r*-алгоритма. *Теорія оптимальних рішень.* Київ: Ін-т кібернетики імені В.М. Глушкова НАН України. 2012. С. 113–118.
4. Журбенко Н.Г. Численная эффективность одной модификации r-алгоритма. *Теорія оптимальних рішень.* Київ: Інститут кібернетики імені В.М. Глушкова НАН України. 2017. С. 33 – 38.
5. Nonsmooth optimization. Eds. C. Lemareshal, R. Mifflin. Oxford: Pergamon Press, 1978. 186 p.

Получено 19.04.2019

Об авторах:

Журбенко Николай Георгиевич,
кандидат физико-математических наук, старший научный сотрудник
Института кибернетики имени В.М. Глушкова НАН Украины,
E-mail: zhurbnick@gmail.com

Лиховид Алексей Петрович,
научный сотрудник Института кибернетики имени В.М. Глушкова НАН Украины.
E-mail: o.lykhovyd@gmail.com