Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов

Н. З. Шор, Н. Г. Журбенко
Кибернетика. – 1971. – № 3. – С. 51–59.

В работах [1], [2] описан класс алгоритмов оптимизации градиентного типа с использованием операции растяжения пространства в направлении градиента (или его аналога) (ОГСРП–алгоритмы). Серьезной проблемой при построении такого рода алгоритмов был вопрос выбора на каждой итерации величины шага спуска, который сравнительно просто разрешался в том случае, когда было заранее известно значение функции в точке минимума. В более общем случае приходилось усложнять алгоритмы [2]. В данной статье предлагается новый класс алгоритмов, основанный на использовании операции растяжения пространства в направлении разности двух последовательных градиентов. В отличие от вышеупомянутых алгоритмов типа ОГСРП этот класс позволяет применить способ выбора шага, близкий к применяемому в методе наискорейшего спуска. При этом получаются алгоритмы, обеспечивающие монотонность или, в некотором смысле, "почти" монотонность спуска.

Дадим описание общей структуры алгоритмов с растяжением пространства в направлении разности двух последовательных градиентов. Алгоритмы этого класса будем сокращенно называть r–алгоритмами. Затем для одного предельного варианта r-алгоритмов докажем, при определенных условиях, квадратичную скорость сходимости. Будут рассмотрены вопросы сходимости других вариантов r-алгоритмов и дан анализ проведенных численных экспериментов.

1. Общая структура r-алгоритмов

Пусть $f(x)$ – непрерывно дифференцируемая функция, определенная в n–мерном эвклидовом пространстве E_n, и выполняется условие

$$\lim_{\|x\|\to\infty} f(x) = +\infty, \tag{1}$$

$g_f(x)$ – градиент функции $f(x)$ в точке x. Рассмотрим следующий итеративный процесс минимизации $f(x)$.

Первый шаг задается начальным значением $x_0 \in E_n$; вычисляем $g_f(x_0)$, выбираем $h_1 > 0$ и находим

$$x_1 = x_0 - h_1\, g_f(x_0)\,.$$

Принимаем $g_f(x_0) = \widetilde{g}_1$; $B = E$ (единичная матрица). Переходим к второму шагу.

Пусть в результате вычислений после k шагов процесса $(k = 1, 2, \ldots)$ получены определенные значения векторов x_k, $\widetilde{g}_k \in E_n$ и матрицы B_k. Опишем $(k+1)$-й шаг процесса.

Вычисляем следующие величины:

1. $g_f(x_k)$ – градиент функции $f(x)$ в точке x_k.

2. $B_k^*\, g_f(x_k) = g_k^*$.

g_k^* – градиент функции $\varphi_k(y) = f(B_k\, y)$ в точке $y = A_k\, x_k$, где $A_k = B_k^{-1}$.

$$\textbf{3.}\quad r_k = g_k^* - \widetilde{g}_k. \tag{2}$$

r_k – разность двух градиентов от функции $\varphi_k(y)$, вычисленных в точках $y_k = A_k\, x_k$ и $\widetilde{y}_k = A_k\, x_{k-1}$.
r_k можно вычислить и другим способом при $k = 1, 2, \ldots$:

$$r_k = B_k^*\, \left[g_f(x_k) - g_f(x_{k-1})\right], \tag{3}$$

однако использование формулы (2) дает некоторую экономию вычислений в целом.

4. $\xi_{k+1} = \dfrac{r_k}{\|r_k\|}$.
Нормировка вектора r_k нужна для подготовки очередной операции растяжения пространства в направлении, определяемом вектором r_k.

5. β_{k+1} – величина, обратная коэффициенту растяжения пространства на $(k+1)$-ом шаге.

6. $B_{k+1} = B_k \cdot R_{\beta_{k+1}}(\xi_{k+1})$.

$B_{k+1} = A_{k+1}$, где $A_{k+1} = R_{\frac{1}{\beta_{k+1}}}(\xi_{k+1}) A_k$ – матрица преобразования пространства после $(k+1)$-го шага; здесь $R_\alpha(\xi)$ – оператор растяжения пространства в направлении ξ с коэффициентом α $(\alpha \geq 0,\ \|\xi\| = 1)$.

Подробно свойства этого оператора описаны в [1]. Отметим основную формулу:

$$R_\alpha(\xi)x = x + (\alpha - 1)(x,\xi)\xi.$$

7. h_{k+1} – шаговый множитель в формуле спуска в заданном направлении.

8. $\widetilde{g}_{k+1} = R_{\beta_{k+1}}\left(\xi_{k+1}\right)\, g_k^*$

$$\begin{aligned}\widetilde{g}_{k+1} = R_{\beta_{k+1}}\left(\xi_{k+1}\right)\, g_k^* &= R_{\beta_{k+1}}\left(\xi_{k+1}\right)\, B_k^*\, g_f\left(x_k\right) = \\ &= \Big(B_k\, R_{\beta_{k+1}}\left(\xi_{k+1}\right)\Big)^*\, g_f\left(x_k\right) = B_{k+1}^* g_f\left(x_k\right).\end{aligned}$$

Отсюда $\widetilde{g}_{k+1}$ – значение градиента функции $\varphi_{k+1}(y) = f\left(B_{k+1}y\right)$ в точке $\widetilde{y}_{k+1} = A_{k+1}\, x_k$.

9. $x_{k+1} = x_k - h_{k+1}\, B_{k+1}\, \widetilde{g}_{k+1}$.

$$\tag{4}$$

Применим к обеим частям формулы (4) оператор A_{k+1}. Тогда

$$y_{k+1} = A_{k+1}\, x_{k+1} = A_{k+1}\, x_k - h_{k+1}\, \widetilde{g}_{k+1} = \widetilde{y}_{k+1} - h_{k+1}\, \widetilde{g}_{k+1}.$$

Таким образом, формула (9) фактически реализует шаг градиентного спуска для функции $\varphi_{k+1}(y)$.

10. Переход к $(k+2)$-му шагу с запоминанием x_{k+1}, $\widetilde{g}_{k+1}$ и B_{k+1} или окончание работы алгоритма при выполнении некоторого критерия остановки.

Приведенное описание класса алгоритмов градиентного типа с изменяемой метрикой порождает конкретные алгоритмы при уточнении способа выбора последовательностей $\{h_{k+1}\}$ и $\{\beta_{k+1}\}$ и критерия остановки. Такие алгоритмы мы будем называть r-алгоритмами. Имеется еще одна возможность модификации r-алгоритмов – использование так называемой операции восстановления, когда периодически после заданного числа итераций происходит "восстановление" матрицы B_k, т. е. она заменяется единичной матрицей. Вопрос о сходимости алгоритмов с монотонным изменением значения минимизирующей функции и с восстановлением решается сравнительно просто, так как он фактически сводится к вопросу о сходимости обычного градиентного спуска без изменения метрики, который подробно исследован в некоторых работах [3]. В частности, если $f(x)$ непрерывно дифференцируема и h_{k+1} в формуле

(4) выбирается из условия минимума по направлению, то при использовании r-алгоритмов с восстановлением множество предельных точек последовательности $\{x\}$ состоит из стационарных точек функции $f(x)$ с одинаковым значением $f(x)$. Доказательство этого факта практически ничем не отличается от доказательства аналогичного результата для метода наискорейшего спуска.

В случае, когда восстановление не предполагается, вопрос о сходимости r-алгоритма требует специальных исследований.

Мы рассмотрим более подробно один предельный вариант r-алгоритма с восстановлением, в котором $\beta_k,\ k = 1, 2, \ldots$, выбирается равным нулю, а h_{k+1} выбирается из условия минимума выражения

$$f\left(x_k - h_{k+1}\, B_{k+1}\, \widetilde{g}_{k+1}\right).$$

При этом, как легко заметить, либо после некоторого числа $k^* < n$ шагов g_{k^*+1} станет равным 0, либо $B_{n+1} = 0$. В самом деле, оператор $R_0(\eta)$ означает проектирование на подпространство, ортогональное вектору η. Произведение операторов $\prod\limits_{i=1}^{k} R_0(\eta_i)$ не зависит от порядка сомножителей, является самосопряженным оператором и осуществляет проекцию на подпространство, являющееся ортогональным дополнением к линейной оболочке векторов $\eta_i,\ i = 1, 2, \ldots$.

Запишем цепочку равенств

$$\begin{aligned} r_1 &= g_f\left(x_1\right) - g_f\left(x_0\right); \ldots \\ r_k &= R_0\left(\frac{r_{k-1}}{\|r_{k-1}\|}\right) \cdot R_0\left(\frac{r_{k-2}}{\|r_{k-2}\|}\right) \cdots R_0\left(\frac{r_1}{\|r_1\|}\right) \cdot \Big[g_f\left(x_k\right) - g_f\left(x_{k-1}\right)\Big]; \\ &\quad k > 1. \end{aligned}$$

Мы видим, что векторы $r_1, \ldots, r_k$ получаются в результате процесса последовательной ортогонализации, примененной к векторам

$$g_f\left(x_1\right) - g_f\left(x_0\right), \quad g_f\left(x_2\right) - g_f\left(x_1\right), \ldots, g_f\left(x_k\right) - g_f\left(x_{k-1}\right),$$

$k = 1, 2, \ldots$ Если $r_1, \ldots, r_n$ отличны от нуля, то оператор $B_{n+1} = 0$, так как он означает проекцию на подпространство, ортогонально дополняющее всё пространство. Если для некоторого $k \leq n$, $r_k = 0$, то это означает, что

$$B^*_{k-1}\Big[g_f\left(x_k\right) - g_f\left(x_{k-1}\right)\Big] = 0,$$

а это возможно лишь при условии

$$B_k^* \, g_f\,(x_{k-1}) = \widetilde{g}_k = 0 \text{ и } x_k = x_{k-1}.$$

Таким образом, в предлагаемом варианте r-алгоритма с необходимостью нужно применять восстановление при условии $B_k^* \, g_f\,(x_{k-1}) = 0$. Этот вариант можно рассматривать как модификацию метода сопряженных градиентов. Легко показать, что для неотрицательно определенной квадратичной формы решение получается за число шагов, не превышающее n. Для произвольной достаточно гладкой функции можно доказать теорему о квадратичной скорости сходимости. Отметим, что доказательство аналогичной теоремы для других модификаций метода сопряженных градиентов встречает серьезные трудности [4].

Теорема. *Пусть функция $f(x)$, определенная в E_n, дважды непрерывно дифференцируема в некоторой окрестности S точки минимума x^*, причем в этой окрестности матрица вторых производных (гессиан) $H(x)$ удовлетворяет условию Липшица*

$$\|H(x) - H(x')\| \le L\,\|x - x'\|\,; \quad x,\, x' \in S. \tag{5}$$

Кроме того, $H(x^*)$ – положительно определенная матрица. Тогда найдется такая окрестность точки $x^* S' \subseteq S$, что если $x_0 \in S'$, то найдется такое число $c > 0$, что

$$\|x_n - x^*\| \le c\,\|x_0 - x^*\|^2\,,$$

где x_n – точка, получаемая после n шагов работы приведенного выше алгоритма (если для некоторого $\overline{k} < n\,\widetilde{g}_{\overline{k}}$, то примем $x_n = x_{\overline{k}}$).

Доказательство. Без ограничения общности будем считать $x^* = 0$ и $f(0) = 0$. Доказательство проведем методом индукции по размерности пространства E_n. При $n = 1$ теорема доказывается тривиально. Пусть теорема справедлива для $n = p$. Докажем ее справедливость для $n = p + 1\,(p \ge 1)$. Обозначим максимальное собственное число оператора $H(0)$ символом M_0 и минимальное – символом m_0. В силу положительной определенности $H(0)$ $m_0 > 0$. Исходя из непрерывности гессиана $H(x)$, получаем, что найдется такая окрестность нуля $S_\delta = \Big\{x \,:\, \|x\| < \delta\Big\}$ и такие положительные числа M и m, что при $x \in S_\delta$ максимальное собственное число оператора $H(x)$ не будет превышать M, а минимальное будет не меньше m.

Пусть e – вектор, $\|e\| = 1$. Примем $x_0 = \varepsilon\, e,\ 0 < \varepsilon < \delta$.

$$g_f\,(x_0) = \int_0^\varepsilon H\,(\mu\, e)\, e\, d\mu = \int_0^\varepsilon \Big[H(0) + G\,(\mu\, e)\Big] e\, d\mu.$$

Используя условие (5), получаем

$$\|G\,(\mu\, e)\| \le \mu\, L,$$

откуда

$$g_f\,(x_0) = \varepsilon\,\Big[H(0)\,e + r\,(x_0)\Big], \tag{6}$$

причем

$$\|\; r\,(x_0)\; \| \le \frac{1}{\varepsilon}\int\limits_0^{\varepsilon} \mu\, L\, d\mu = \frac{L\,\varepsilon}{2}. \tag{7}$$

Определим x_1 в соответствии с исследуемым алгоритмом:

$$x_1 = x_0 - h^*\frac{g_f\,(x_0)}{\|g_f\,(x_0)\,\|},$$

где h^* – минимальный положительный корень уравнения

$$\left(g_f\,(x_0)\,,\, g_f\left(x_0 - h^*\frac{g_f\,(x_0)}{\|g_f\,(x_0)\,\|}\right)\right) = 0.$$

Легко видеть, что при достаточно малом ε

$$h^* \le \frac{\|g_f\,(x_0)\,\|}{m} \le \frac{M\,\varepsilon}{m}.$$

В самом деле, пусть

$$x(h) = x_0 - h\frac{g_f\,(x_0)}{\|g_f\,(x_0)\|}, \quad \varepsilon < \frac{\delta}{\dfrac{M}{m} + 1}\,.$$

Рассмотрим функцию $\varphi(h) = f\big(x(h)\big)$. При $0 \le h \le \dfrac{M\varepsilon}{m}$ $x(h) \in S_\delta$. Тогда в силу свойств гессиана $H(x)$ в области S_δ

$$\frac{d^2\varphi(h)}{dh^2} \ge m,$$

откуда

$$\frac{d\varphi}{dh} \le \|g_f\,(x_0)\| - mh \le M\varepsilon - mh.$$

Значит, $h^* \le \dfrac{M\,\varepsilon}{m}$, что и требовалось доказать.

Далее,

$$\begin{aligned} g_f(x_1) &= g_f\left(x_0 - h^*\frac{g_f(x_0)}{\|g_f(x_0)\|}\right) = \\ &= g_f(x_0) - \int_0^{h^*} H\left(x_0 - \mu\frac{g_f(x_0)}{\|g_f(x_0)\|}\right)\frac{g_f(x_0)}{\|g_f(x_0)\|}d\mu = \\ &= g_f(x_0) - \int_0^{h^*}\left[H(x_0) + G_1\left(\mu\frac{g_f(x_0)}{\|g_f(x_0)\|}\right)\right]\frac{g_f(x_0)}{\|g_f(x_0)\|}d\mu, \end{aligned}$$

причем $\left\|G_1\left(\mu\frac{g_f(x_0)}{\|g_f(x_0)\|}\right)\right\| \leq L\mu$. Отсюда

$$g_f(x_1) = g_f\left(x_0 - h^*\frac{g_f(x_0)}{\|g_f(x_0)\|}\right) = \\ = g_f(x_0) - h^* H(x_0)\frac{g_f(x_0)}{\|g_f(x_0)\|} + h^* r_1(x_0), \quad (8)$$

причем

$$\|r_1(x_0)\| \leq \frac{L\,h^*}{2} < \frac{L\,M_\varepsilon}{2m}. \quad (9)$$

Используя (6)–(9), получаем

$$\begin{aligned} &\Big(g_f(x_0), g_f(x_1)\Big) = \\ = \;& \varepsilon^2\left[\Big(H(0)\,e + r(x_0), H(0)\,e + r(x_0)\Big) - \frac{h^*}{\|g_f(x_0)\|}\times\right. \\ & \times\left.\left(\Big(H(0) + G(x_0)\Big)\Big(H(0)\,e + r(x_0)\Big), H(0)\,e + r(x_0)\right)\right] + \\ & + \;\varepsilon\,h^*\Big(H(0)\,e + r(x_0), r_1(x_0)\Big) = \\ = \;& \varepsilon^2\left[\Big(H(0)\,e, H(0)\,e\Big) - \frac{h^*}{\|g_f(x_0)\|}\Big(H^2(0)\,e, H(0)\,e\Big)\right] + r_2(x_0), \end{aligned}$$

причем при достаточно малом ε, как легко видеть, $|r_2(x_0)| \leq d\varepsilon^3$, где d – некоторое положительное число. Приравняв $\Big(g_f(x_0), g_f(x_1)\Big)$ нулю, получаем

$$1-\frac{d\varepsilon}{\|H(0)\,e\|^2}q \le h^* \le 1+\frac{d\varepsilon}{\|H(0)\,e\|^2}q,$$

где

$$q=\frac{\|H(0)\,e\|^2\;\|g_f\,(x_0)\|}{(H^2(0)\,e,H(0)\,e)}.$$

Теперь покажем, что расстояние от начала координат до гиперплоскости, проходящей через точку x_1 и с нормалью, задаваемой вектором

$$\Delta = g_f\,(x_0)-g_f\,(x_1)\,,$$

имеет порядок ε^2.

Запишем уравнение этой гиперплоскости:

$$\Big(x-x_1,g_f\,(x_0)-g_f\,(x_1)\Big)=0.$$

Расстояние от начала координат до гиперплоскости вычисляется по формуле

$$s=\frac{\big|\big(g_f\,(x_0)-g_f\,(x_1)\,,x_1\big)\big|}{\big\|g_f\,(x_0)-g_f\,(x_1)\big\|},$$

$$x_1=x_0-h^*\frac{g_f\,(x_0)}{\|g_f\,(x_0)\,\|}=x_0-\frac{\|H(0)\,e\|^2\;g_f\,(x_0)}{(H^2(0)\,e,H(0)\,e)}-\frac{\nu q g_f\,(x_0)}{\big\|g_f\,(x_0)\big\|}, \qquad (10)$$

где

$$-\frac{d\varepsilon}{\|H(0)\,e\|^2}\le\nu\le\frac{d\varepsilon}{\|H(0)\,e\|^2}. \qquad (11)$$

Далее, используя (8)–(11), получаем

$$\Big(g_f\,(x_0)-g_f\,(x_1)\,,x_1\Big)=$$

$$=\frac{h^*\varepsilon^2}{\big\|g_f\,(x_0)\big\|}\Big(\Big(H(0)+G\,(x_0)\Big)\Big(H(0)\,e+r\,(x_0)\Big),$$

$$e-\frac{\|H(0)\,e\|^2}{(H^2(0)\,e,H(0)\,e)}\Big(H(0)\,e+r\,(x_0)\Big)\Big)-$$

$$- \quad \frac{1}{\left\|g_f(x_0)\right\|}\Big(\nu q g_f(x_0), g_f(x_0) - g_f(x_1)\Big) - \Big(h^* r_1(x_0), x_1\Big) =$$
$$= \quad \varepsilon^2 \frac{h^*}{\left\|g_f(x_0)\right\|}\Big(H^2(0)e, e\Big) - \Big(H(0)e, H(0)e\Big) + r_3(x_0) = r_3(x_0),$$

где $|r_3(x_0)| \le c_1\varepsilon^3$, а c_1 – положительное число, которое легко вычисляется с использованием оценок (7), (9), (11). Так как

$$\Big(g_f(x_0), g_f(x_1)\Big) = 0$$

то

$$\|g_f(x_1) - g_f(x_0)\| \ge \|g_f(x_0)\| \ge m\varepsilon.$$

Отсюда

$$s \le \frac{c_1}{m}\varepsilon^2 = c_2\varepsilon^2, \quad \text{где} \quad c_2 = \frac{c_1}{m}. \tag{12}$$

Рассмотрим функцию $\psi(x)$, определенную в p-мерной гиперплоскости

$$L(x_0) = \Big\{x \,:\, \Big(g_f(x_1) - g_f(x_0), x - x_1\Big) = 0\Big\}$$

и равную $f(x)$ для точек этой гиперплоскости.

Покажем, что $\min \psi(x) = \min\limits_{x \in L(x_0)} f(x)$ достигается в точке такой, что

$$\left\|x_1^*\right\| \le c_3\,\varepsilon^2.$$

где c_3 – положительная величина, не зависящая от ε при достаточно малом ε.

Рассмотрим точку $\overline{x}_1$, лежащую в $L(x_0)$ на кратчайшем расстоянии от начала координат. Из (12) получаем

$$s = \left\|\overline{x}_1\right\| \le c_2\,\varepsilon^2,$$

откуда

$$\min_{x \in L(x_0)} f(x) \le f(\overline{x}_1) \le \frac{M\varepsilon^4 c_2^2}{2} = c_4\,\varepsilon^4,$$

С другой стороны,

$$f(x_1^*) \ge \frac{m\left\|x_1^*\right\|}{2}.$$

Получаем

$$\frac{m\left\|x_1^*\right\|^2}{2} \leq c_4\,\varepsilon^4, \qquad \left\|x_1^*\right\| \leq \sqrt{\frac{2c_4}{m}}\varepsilon^2 = c_3\,\varepsilon^2,$$

Заметим, что c_3 определяется константами L, M и m. Так как дальнейший спуск будет проходить в гиперплоскости $L\left(x_0\right)$ размерности p, а по предположению индукции рассматриваемая теорема справедлива для функции, определенной в p-мерном пространстве, то справедливо соотношение

$$\left\|x_n\right\| = \left\|x_{p+1}\right\| \leq \left\|x_1^*\right\| + \left\|x_{p+1} - x_1^*\right\| \leq c_3\,\varepsilon^2 + c_p\left\|x_1 - x_1^*\right\|, \qquad (13)$$

где c_p – число, определяемое по предположению индукции только константами L, M, m и p при достаточно малом значении $\left\|x_1 - x_1^*\right\|$. Оценим $\left\|x_1 - x_1^*\right\|$.

Так как

$$f\left(x_0\right) > f\left(x_1\right) \geq f\left(x_1^*\right) \geq f\left(0\right) = 0$$

то

$$f\left(x_1\right) - f\left(x_1^*\right) < f\left(x_0\right) \leq \frac{M\varepsilon^2}{2}.$$

С другой стороны,

$$\frac{m\left\|x_1 - x_1^*\right\|^2}{2} \leq f\left(x_1\right) - f\left(x_1^*\right).$$

Получаем

$$\left\|x_1 - x_1^*\right\|^2 \leq \frac{M}{m}\varepsilon^2.$$

Окончательно имеем

$$\left\|x_n\right\| \leq \left(c_3 + c_p\frac{M}{m}\right)\varepsilon^2 = c\left\|x_0\right\|^2,$$

где c – величина, зависящая только от L, M, m и n. Доказательство теоремы закончено.

2. Варианты алгоритмов без восстановления. Численные эксперименты

Ниже приводится вариант r-алгоритма без восстановления. Хотя строгие результаты о сходимости этих алгоритмов и скорости сходимости пока не получены, численные эксперименты показывают высокую скорость сходимости по сравнению с достигаемой в различных модификациях алгоритмов с изменяемой метрикой или метода сопряженных градиентов [4], [6]. Существенная особенность, однако, состоит в том, что определенные модификации r–алгоритмов применимы для минимизации негладких функций, в то время как другие методы, обладающие ускоренной сходимостью, требуют во всяком случае непрерывности градиента.

r–алгоритм с постоянным коэффициентом растяжения был реализован программой для вычислительной машины БЭСМ-6. Опишем кратко программу. Так как общая схема алгоритма описана, то остановимся лишь на алгоритме одномерного поиска минимума функции $f(x)$ в направлении η, с помощью которого осуществляется выбор шага h_{k+1}. Для реализации наискорейшего спуска с высокой точностью приходится производить большое количество вычислений значений функции. В разработанном алгоритме поиск минимума по направлению производится таким образом, что точность поиска минимума увеличивается в процессе счета.

Во всех приведенных числовых примерах одномерный поиск минимума производится следующим образом.

Выбираем числа $\mu \geq 1,\ 0 < \gamma < 1$ и целое число L. Выбираем число h – начальный пробный шаг.

На $(k+1)$–ой итерации определены:

x_k –точка, полученная на k–ой итерации;

η – вектор, в направлении которого необходимо осуществить спуск из точки x_k;

h – пробный шаг.

Из точки x_k делаем шаг длины h в направлении $\eta\ :\ z_1 = x_k + h\,\eta$. Вычисляем $f\,(z_1)$. Если $f\,(z_1) < f\,(x_k)$, то из точки z_1 снова делаем шаг длины h в направлении $\eta\ :\ z_2 = z_1 + h\,\eta$. Вычисляем $f\,(z_2)$. Если $f\,(z_2) < f\,(z_1)$, то из точки z_2 опять делаем шаг по η и т. д. до тех пор, пока на некотором шаге значение функции не возрастает. Эту последнюю точку и принимаем за x_{k+1}. Пусть l – число шагов при этой процедуре. Тогда, если оказалось, что $l > L$, то за новый пробный шаг принимаем число

$\left(\mu \dfrac{l}{L} h\right)$. Если оказалось, что функция возрастает на первом шаге, то за новый шаг принимаем число $(\gamma\, h)$, во всех остальных случаях пробный шаг h не изменяем.

При счете всех приведенных задач γ, μ и начальный шаг h принимались равными соответственно 0.1, 5/4, 0.1. Вышеописанная процедура не гарантирует монотонного убывания функции, однако, как показали эксперименты, возрастание функции происходит очень редко.

В программе производится контроль малости элементов b_{ij} матрицы $B\,(B = A^{-1}$, где A – матрица преобразования пространства переменных). Через каждые k итераций вычисляется величина $d = \max\limits_i |b_{ii}|$ и если $d < \varepsilon$, то матрицу B умножаем на число ϱ. При счете $k = 10$, $\varepsilon = 1$, $\varrho = 10^3$.

Остановимся на вопросе трудоемкости программы. Основными дополнительными операциями в программе являются операции умножения матрицы на вектор, требующий $2n^2$ операций умножения и сложения (n – размерность пространства переменных), и операция умножения матрицы B на матрицу оператора растяжения $R_\alpha(\eta)$.

Благодаря тому, что $R_\alpha(\eta)$ – матрица специального вида, операция $B \cdot R_\alpha(\eta)$ требует $5n^2$ операций умножения и сложения.

Основной массив необходимой дополнительной памяти занимает матрица B, для хранения которой необходимо n^2 ячеек. Поэтому, используя лишь оперативную память машины БЭСМ-6, мы можем решать задачи с числом переменных до 140.

При описании результатов счета приняты следующие обозначения:

$f = f\,(x_1, \ldots, x_n)$ – минимизируемая функция;

x^*–истинная точка минимума $f(x)$;

f^* – истинный минимум $f(x)$;

x_0 – начальная точка – точка на нулевой итерации;

$\widetilde{x}_N$ – точка, полученная в результате счета;

k – счетчик итерации;

f – значение функции;

α – коэффициент растяжения;

d – величина градиента в точке $\widetilde{x}$: $d = \left\|\nabla f\,(\widetilde{x})\right\|$;

$\delta = \max\limits_i |\widetilde{x}_i - x_i^*|$;

l – среднее количество вычислений значений функции на одной итерации.

Условием прекращения счета во всех примерах являлось выполнение одного из следующих неравенств:

$$\left\|\Delta f\left(x_k\right) < 10^{-6}\right\|, \quad \left\|x_k - x_{k-1}\right\| < 10^{-7}, \quad \left\|B^* \nabla f\left(x_k\right)\right\| < 10^{-18}.$$

N – номер итерации, при которой произошел останов.

Пример 1 [5], [6].

$f(x) = \left(x_1^2 - x_2\right)^2 + (x_1 - 1)^2;$
$x^* = (1; 1); \quad f^* = 0; \quad x_0 = (-1,2; 1); \qquad$ при α, равном 2 и 3,
$\widetilde{x}_N = (1,000000; 1,000000).$

Пример 2 [4].

$f(x) = \sum\limits_{i=1}^{10} \left(e^{-0,2i} + 2e^{-0,4i} - x_1 e^{-0,2x_2 i} - x_3 e^{-0,4x_4 i}\right)^2;$
$x^* = (1; 1; 2; 1); \quad f^* = 0; \quad x_0 = (0; 0; 0; 0);$
$\alpha = 2; \quad \widetilde{x}_{83} = (0,999999; 1,000000; 2,000000; 1.000000);$
$\alpha = 3; \quad \widetilde{x}_{90} = (0,999999; 0,999999; 2,000001; 1,000000).$

Пример 3 [4].

$f(x) = 10^{-3} \sum\limits_{i=1}^{10} \left(10^3 e^{-0,2i} + 210^3 e^{-0,4i} - x_1 e^{-0,2x_2 i} - x_3 e^{-0,2x_4 i}\right)^2;$
$x^* = (1000; 1; 2000; 2); \quad f^* = 0; \quad x_0 = (500; 0; 2500; 3);$
$\alpha = 2; \quad \widetilde{x}_{100} = (1000,000; 1,000000; 2000,000; 2,000000);$
$\alpha = 3; \quad \widetilde{x}_{72} = (1000,001; 1.000000; 1999,999; 2,000000).$

Пример 4 [6].

$$\begin{aligned} f(x) = {} & 100\left(x_1^2 - x_2\right) + (x_1 - 1)^2 + 90\left(x_3^2 - x_4\right)^2 + (x_3 - 1)^2 + \\ & + 10,1\Big((x_2 - 1)^2 + (x_4 - 1)^2\Big) + 19,8\,(x_2 - 1)(x_4 - 1); \end{aligned}$$
$x^* = (1; 1; 1; 1); \quad f^* = 0; \quad x_0 = (-3; -1; -3; -1);$
$\alpha = 2; \quad \widetilde{x}_{99} = (0,999999; 0,999999; 1,000000; 1,000000);$
$\alpha = 3; \quad \widetilde{x}_{76} = (1.000000; 1,000000; 1,000000; 1,000000).$

Пример 5 [6].

$f(x) = \left(e^{x_1} - x_2\right)^4 + 100\,(x_2 - x_3)^6 + th^4\,(x_3 - x_4) + x_1^8 + (x_4 - 1)^2;$
$x^* = (0; 1; 1; 1); \quad f^* = 0; \quad x_0 = (1; 2; 2; 2);$

$$\alpha = 2; \quad \widetilde{x}_{36} = (0,0006; 1,0005; 1,0000; 1,0000);$$
$$\alpha = 3; \quad \widetilde{x}_{32} = (-0,0004; 0,9996; 1,0000; 1,0000).$$

Пример 6 [6].

$$f(x) = (x_1 + 10x_2)^2 + 5\,(x_3 - x_4)^2 + (x_2 - 2x_3)^2 + 10\,(x_1 - x_4)^4;$$
$$x^* = (0; 0; 0; 0); \quad f^* = 0; \quad x_0 = (10; 10; 10; -10);$$
$$\alpha = 2; \quad \widetilde{x}_{50} = \Big(4_{10^{-7}}; -2_{10^{-7}}; 6_{10^{-7}}; 7_{10^{-7}}\Big);$$
$$\alpha = 3; \quad \widetilde{x}_{46} = \Big(3,8_{10^{-5}}; 0,4_{10^{-5}}; -4,6_{10^{-5}}; -4,6_{10^{-5}}\Big).$$

Более подробная информация о результатах численных экспериментов содержится в таблице.

Приведенные выше данные о решении с помощью r–алгоритма ряда задач минимизации с ярко выраженными овражными особенностями говорят о том, что этот алгоритм является эффективным средством решения такого рода задач. По числу итераций, требуемых для достижения определенной точности, он близок к наиболее удачным модификациям алгоритмов типа Давидона–Флетчера–Пауэлла и метода сопряженных градиентов [6], [4]. Алгоритм устойчив по отношению к неточностям определения минимума по направлению (в алгоритме был использован заведомо грубый способ локализации минимума по направлению). Как показал опыт вычислений, коэффициент растяжения пространства α целесообразно выбирать между 2 и 3 (мы приводим лишь результаты экспериментов для α, равных 2 и 3, но опыты проводились и для множества других значений коэффициента растяжения), так как в большинстве случаев при дальнейшем увеличении α количество итераций если и уменьшается, то незначительно, но зато заметно увеличивается среднее число вычислений функции при поиске минимума по направлению (конечно, эти рекомендации относятся лишь к той конкретной модификации алгоритма, которую мы применяли).

Были проведены также эксперименты, в которых счет производился по несколько измененному алгоритму. Изменение состояло в том, что минимум по направлению находился с высокой точностью. При этом наблюдалось уменьшение числа итераций на 20–30% при сохранении прежних критериев остановки.

	I		II		III		IV		V		VI	
α	2	3	2	3	2	3	2	3	2	3	2	3
k	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$	$f(x_k)$
0	2.4_{10+1}	2.4_{10+1}	1.0_{10+1}	1.0_{10+1}	5.4_{10+2}	5.4_{10+2}	1.9_{10+4}	1.9_{10+4}	2.3_{10+0}	2.3_{10+0}	1.6_{10+6}	1.6_{10+6}
10	1.5_{10+0}	9.2_{10-1}	2.7_{10-4}	1.5_{10-3}	6.7_{10-2}	4.8_{10-2}	4.0_{10+0}	3.8_{10+0}	1.5_{10-4}	1.5_{10-4}	6.3_{10+0}	1.5_{10+1}
20	2.5_{10-1}	8.8_{10-3}	7.7_{10-5}	2.7_{10-4}	1.9_{10-2}	1.3_{10-2}	2.4_{10+0}	2.2_{10+0}	1.3_{10-9}	4.2_{10-9}	2.2_{10-4}	4.8_{10-3}
30	1.6_{10-3}	5.5_{10-9}	3.5_{10-5}	1.0_{10-4}	1.5_{10-2}	3.6_{10-3}	1.3_{10+0}	5.5_{10-1}	4.0_{10-14}	2.2_{10-13}	2.3_{10-8}	1.6_{10-7}
40	2.8_{10-7}	-	8.2_{10-6}	4.2_{10-5}	5.6_{10-3}	1.6_{10-4}	4.6_{10-1}	1.8_{10-2}	-	-	5.1_{10-11}	2.4_{10-11}
50	1.9_{10-10}	-	2.1_{10-7}	1.5_{10-5}	1.1_{10-3}	1.3_{10-7}	4.3_{10-3}	4.5_{10-5}	-	-	-	4.6_{10-13}
60	3.4_{10-13}	-	5.79_{10-11}	2.0_{10-7}	1.9_{10-5}	2.7_{10-10}	2.2_{10-4}	4.0_{10-8}	-	-	-	-
70	-	-	1.2_{10-12}	7.1_{10-10}	2.6_{10-8}	5.3_{10-14}	3.3_{10-6}	3.2_{10-11}	-	-	-	-
80	-	-	6.0_{10-15}	3.9_{10-12}	1.1_{10-10}	-	3.0_{10-7}	-	-	-	-	-
90	-	-	-	2.4_{10-17}	1.9_{10-13}	-	3.2_{10-10}	-	-	-	-	-
N	63	39	83	90	100	72	99	76	36	32	46	50
$f(x_N)$	1.5_{10-14}	2.3_{10-15}	8.5_{10-17}	2.4_{10-17}	9.0_{10-16}	3.7_{10-14}	5.7_{10-12}	3.7_{10-12}	5.4_{10-16}	1.2_{10-14}	3.4_{10-14}	4.6_{10-13}
δ	10^{-7}	10^{-7}	10^{-7}	10^{-6}	10^{-4}	10^{-3}	10^{-6}	10^{-6}	10^{-3}	10^{-3}	10^{-6}	10^{-4}
d	5.4_{10-6}	7.7_{10-7}	2.8_{10-8}	3.5_{10-8}	8.6_{10-7}	4.5_{10-7}	2.1_{10-5}	4.0_{10-5}	4.8_{10-8}	2.5_{10-7}	3.0_{10-6}	1.1_{10-5}
l	4.3	7.6	4.8	4.7	4.3	5.7	4.4	6.9	4.1	3.8	10	13.9

[a]

[a]Римскими цифрами в первой строке указаны номера примеров

3. Возможность применения r-алгоритмов к задачам минимизации негладких функций

Пусть мы имеем кусочно-гладкую выпуклую функцию вида

$$f(x) = \max_{1 \le i \le l} \varphi_i(x),$$

где $\varphi_i(x)$ – выпуклые гладкие функции, и пусть мы находимся в точке x_0 такой, что $\varphi_i(x_0) = f(x_0)$ при $i \in I^*$, где I^* – некоторое подмножество целочисленного интервала $[1, l]$, и в некоторой окрестности этой точки существует гладкая гиперповерхность

$$S^* = \Big\{ x \,:\, f(x) = \varphi_i(x), \quad i \in I^* \Big\}.$$

Эту гиперповерхность можно рассматривать, как "русло оврага" в том случае, когда

$$\big(g_{\varphi_i}(x), g_{\varphi_j}(x)\big) < 0, \quad i, j \in I^*, \quad i \neq j.$$

Из определения гиперповерхности S^* следует, что для любого касательного к этой гиперплоскости в точке x вектора $r(x)$ справедливо соотношение

$$\Big(r(x), g_{\varphi_i}(x) - g_{\varphi_j}(x) \Big) = 0, \quad i, j \in I^*.$$

Таким образом, растяжение пространства в направлении $g_{\varphi_i}(x) - g_{\varphi_j}(x)$ уменьшает составляющие обобщенных градиентов, направленные ортогонально "руслу оврага", и оставляет без изменения составляющие, направленные вдоль "русла оврага".

Эти эвристические соображения показывают, что последовательное применение операций растяжения пространства в направлении разности двух последовательных обобщенных градиентов при определенном способе выбора шага (когда мы идем до минимума в выбранном направлении и переходим его с пробным шагом, чтобы вычислить градиент по другую сторону от "русла оврага") приближенно соответствует операции проектирования градиента на "русло оврага".

Были проведены численные эксперименты, которые показывают, что применение операции растяжения пространства в направлении разности двух последовательных градиентов значительно ускоряет сходимость обобщенного градиентного спуска. По материалам исследования применимости r–алгоритмов к задачам минимизации негладких функций готовится отдельная статья.

Литература

1. Шор Н.З. Использование операций растяжения пространства в задачах минимизации выпуклых функций // Кибернетика. – 1970. – № 1. – С. 6–12.

2. Н. З. Шор, О скорости сходимости метода обобщенного градиентного спуска с растяжением пространства // Кибернетика. – 1970. – № 2. – С. 80–85.

3. Любич Ю.И. Общая теория релаксационных процессов для выпуклых функционалов // Успехи математических наук. – 1970. – № 1.

4. Поляк Б.Т. Метод сопряженных градиентов, Труды II Зимней школы по математическому программированию и смежным вопросам (Дрогобыч), вып. 1, М., 1969.

5. Розенброк Х., Стори С. Вычислительные методы для инженеров–химиков. – М.: "Мир", 1968.

6. Huang H.I., Levy A.V. Experiments on Quadratically Convergent Algorithms for Function Minimization // Journal of Optimization Theory and Applications. – 1970. – 3, v. 6.