РЕГУЛЯРИЗАЦИЯ МАТРИЦЫ ПРЕОБРАЗОВАНИЯ В МОДИФИКАЦИЯХ R-АЛГОРИТМА

Н.Г. ЖУРБЕНКО, А.П. ЛИХОВИД
Институт кибернетики НАН Украины, Киев, Украина
zhurbnick@gmail.com, o.lykhovyd@gmail.com

Аннотация. *Для различных модификаций r-алгоритма предлагается процедура регуляризации матрицы преобразования пространства, предназначенная для предотвращения ее вырождения. Результаты численных исследований показывают, что использование процедуры регуляризации матрицы улучшает устойчивость алгоритмов с преобразованием пространства по отношению к ошибкам округления.*
Ключевые слова: *метод оптимизации, субградиентный алгоритм, преобразование пространства*.

Более 50 лет назад был разработан субградиентный алгоритм минимизации с растяжением пространства в направлении разности двух последовательных градиентов – r-алгоритм [1], [2]. Практика использования r-алгоритма показывает, что до сих пор он является одним из наиболее эффективных алгоритмов негладкой оптимизации.

r-алгоритм используется с большими значениями коэффициентов растяжения пространства (≈ 2). Поэтому при решении задач небольшой размерности с большой точностью это может привести к вырождению матрицы преобразования. В работе [3] предложена процедура регуляризации этой матрицы применительно к r-алгоритму

Отметим, что к настоящему времени разработано семейство модификаций r-алгоритма с программным управлением значениями коэффициентов растяжения пространства [4], [5]. В это семейство входит и алгоритм с растяжением пространства по направлению разности нормированных субградиентов [6]. Приведенная в данной работе процедура регуляризации может использоваться и для этого семейства модификаций r-алгоритма.

Рассматривается задача безусловной минимизации субдифференцируемой функции $f(x)$ в R^n. Обозначим $\partial f(x)$ множество субградиентов функции $f(x)$ в точке x. Общая схема рассматриваемых (суб)градиентных алгоритмов с преобразованием пространства состоит в следующей итеративной процедуре:

$$x_{k+1} = x_{k+1} - h_k B_k B_k^* g_k / \| B_k^* g_k \|, \tag{1}$$

где h_k – шаговый множитель; $g_k \in \partial f(x_k)$; B_k – матрица обратная к матрице преобразования пространства A_k $(B_k = A_k^{-1})$; $g_k^* = B_k^* g_k$ – субградиент в преобразованном пространстве, соответствующий субградиенту g_k в исходном пространстве. Пусть $g_{k+1} \in \partial f(x_{k+1})$, $g_{k+1}^* = B_k^* g_{k+1}$. На шаге $k+1$ выполняется очередное преобразование пространства: $B_{k+1} = B_k T_{k+1}$, где T_{k+1} – оператор обратный к оператору очередного преобразования пространства. Для рассматриваемого семейства модификаций r-алгоритма оператор T_{k+1} определяется на основе субградиентов g_k^* и g_{k+1}^*.

Процедура регуляризации состоит в дополнительном условии на детерминант оператора T_{k+1}: $\det T_{k+1} = 1$.

Рассмотрим пример модификации r-алгоритма.

В *r*-алгоритме используется оператор растяжения пространства [2]: $R(\eta) = (\alpha - 1)\eta\eta^T + I$, где $\eta \in R^n$, α – направление и коэффициент растяжения пространства, $|\eta| = 1$, $\alpha \geq 0$.

Оператор $T_{k+1} = R_{\beta_{k+1}}(\eta_{k+1})$, $\eta_{k+1} = (g_{k+1}^* - g_k^*) / \| g_{k+1}^* - g_k^* \|$.

Как следует из итеративной схемы (1) в r-алгоритме используется операция деления на $| g_k^* |$ – норму субградиента в преобразованном пространстве. Поэтому при программной реализации *r*-алгоритма необходимо принять меры по устранению возможной ошибки «деление на нуль». Это связано с тем, что $g_k^* = B_k^* g(x_{k-1})$, а матрица B_k равна произведению матриц операторов «сжатия» ($B_k = B_{k-1} R_{\beta_k}(\eta_{k-1})$). Отсюда следует, что с увеличением числа итераций величина $| g_k^* |$ может быть существенно меньше $| g_k |$. Таким образом, возможна ситуация, когда задача с заданной точностью еще не решена, а величина $|g_k^*|$ принимает недопустимо малое значение.

Для учета такой ситуации в некоторых программных реализациях r-алгоритма используется процедура «восстановления» матрицы B_k. Однако использование такой процедуры может привести к увеличению трудоемкости алгоритма при решении задач небольшой размерности с высокой точностью.

Процедура регуляризации применительно к r-алгоритму будет состоять в следующем. Заметим, что $\det(R_{\beta_{k+1}}(\eta_k)) = \beta_{k+1} < 1$. Вместо оператора $R_{\beta_{k+1}}(\eta_k)$ будем использовать оператор $\tilde{R}_{\beta_{k+1}}(\eta_k) = (1/\sqrt[n]{\beta_{k+1}})R_{\beta_{k+1}}(\eta_k)$, для которого $\det(\tilde{R}_{\beta_{k+1}}(\eta_k)) = 1$. Таким образом, соответствующее оператору $\tilde{R}_{\beta_{k+1}}(\eta_k)$ преобразование пространства происходит с сохранением объемов. Оператор $\tilde{R}_{\alpha_{k+1}}(\eta_k)$ фактически можно интерпретировать как «растяжение» пространства оператором $R_{\alpha_{k+1}}(\eta_k)$ и равномерным сжатием по всем направлениям с коэффициентом $\sqrt[n]{\beta_{k+1}} = 1/\sqrt[n]{\alpha_{k+1}}$. Заметим, что это дополнительное сжатие не изменяет структуру поверхностей уровня функции – изменяется лишь их масштаб.

Таким образом модифицированный алгоритм обозначается в дальнейшем как Ř-алгоритм.

Приведем результаты численных исследований эффективности Ř-алгоритма в сравнении с r-алгоритмом на примере задачи минимизации функции Розенброка [7] для двух переменных:

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2.$$

	α	k	k_g	$k_{restart}$
Ř	2	66	224	0
r	2	72	231	3
Ř	3	48	265	0
r	3	61	289	4

Таблица. Минимизация функции Розенброка.

В таблице приняты следующие обозначения: k – номер итерации, на которой алгоритм прекратил работу; k_g – количество вычислений субградиента; $k_{restart}$ – количество процедур восстановления матрицы. Точность решения по функционалу 10^{-18}.

Выводы. Результаты численных исследований показывают, что эффективности Ř-алгоритма и *r*-алгоритма примерно одинаковы. Однако использование процедуры регуляризации матрицы улучшает устойчивость алгоритмов с преобразованием пространства по отношению к ошибкам округления.

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

Литература.

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 применение. Киев: Наук. Думка. 1979. 200 с.
3. Журбенко Н.Г., Лиховид А.П. Регуляризация матрицы преобразования r-алгоритма. Теорія оптимальних рішень. Київ: Ін-т кібернетики ім. В.М.Глушкова НАН України, 2018. С. 145–151.
4. Журбенко Н.Г. Об одной модификации r-алгоритма. Материалы 3-й международной конференции „Математическое моделирование, оптимизация и информационные технологии”. Кишинеу: Эврика, 2012. С. 355–361.
5. Журбенко Н.Г., Чумаков Б.М. Программное управление коэффициентами растяжения r-алгоритма. Теорія оптимальних рішень. Київ: Ін-т кібернетики ім. В.М. Глушкова НАН України, 2012. С. 113–118.
6. Журбенко Н.Г. r-алгоритм на основе разности нормированных субградиентов. Материалы 4-й Межд. науч. конф. "Математическое моделирование, оптимизация и информационные технологии", г. Кишинэу, Республика Молдова, 25-28 марта 2014 г. Кишинэу: Эврика, 2014. Т. II. С. 197–201.
7. Rosenbrock, H.H. (1960). An automatic method for finding the greatest or least value of a function. The Computer Journal. 3 (3): 175–184.