УДК 519.8

А.П. ЛИХОВИД, А.В. ИВЛИЧЕВ

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ДВУХ РЕАЛИЗАЦИЙ *R*-АЛГОРИТМА

Приведены результаты тестирования двух реализаций r-алгоритма для функций, обладающих высокой степенью «овражности». Результаты вычислительных экспериментов показали, что обе программные реализации r-алгоритма позволяют эффективно и с достаточной точностью находить решения таких оптимизационных задач.

Введение. В настоящее время имеется много реализаций *r*-алгоритма с адаптивной регулировкой шага на языках программирования Фортран, Си, С++, С# и Octave. В данной работе рассматриваются результаты тестирования двух программных реализаций *r*-алгоритма. Первая программа **amplralg** – одна из реализаций на языке С++ с использованием AMPL-интерфейса программы **ralg** на языке Фортран [1]. Она доступна на сайте Института кибернетики по ссылке [2]. Вторая программа **amplralgb5 –** реализация на языке С++ программы **ralgb5** на языке Octave [3, 4]. Для возможности проведения вычислительных экспериментов в среде AMPL эта программа тоже была реализована с использованием AMPL-интерфейса. Программа **amplralg** в отличие от **amplralgb5** использует вычислительную схему экономной B-формы. Кроме этого, они отличаются наличием у программы **amplralg** процедуры восстановления матрицы *B*, а также разными процедурами увеличения шагового множителя: **amplralg** увеличивает шаг в q_1 на каждом шаге после 3 шагов спуска, а **amplralgb5** – через каждые 3 шага спуска.

Цель данной работы – тестирование двух реализаций *r*-алгоритма для функций, обладающих высокой степенью «овражности». Тестирование проводилось в среде системы моделирования AMPL. Далее приводятся результаты вычислительных экспериментов.

1. Вычислительные эксперименты для примера maxquad. Задача **maxquad** [5, стр. 151] состоит в минимизации существенно овражной выпуклой негладкой функции от 10 переменных:

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

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 $A_{kii} = i|\sin k| / 10 + \sum_{j \ne i} |A_{kij}|$, а компоненты векторов b_k определяются $b_{ki} = e^{i/k} \sin(ik)$. Начальным приближением является точка $x_0 = (1, \dots, 1)^T \in R^{10}$, в которой реализуется значение функции $f(x_0) = 5337.06643$. Функция **maxquad** имеет единственную точку минимума, минимальное значение функции f^* определяется его достаточно точным приближением $f^*_{\min} = -0.841408334596$ (с точностью 10^{-12}, 12 цифр после точки). В табл. 1 – 3 приведены результаты программ **amplralgb5** и **amplralg.** Здесь α – коэффициент растяжения, ε_x – параметр останова по аргументу, *itn* – количество итераций, *nfg* – количество вычислений значения функции и ее субградиента, $f_r - f^*$ – отклонение найденного значения функции от оптимального $f^*_{\min} = -0.841408334596$. Фиксированные значения параметров: $q_2 = 1.1$, максимальное количество итераций 5000, $h_0 = 1$.

ТАБЛИЦА 1. Решение задачи **maxquad** с помощью программы **amplralgb5**

ε_x	$\alpha = 2.0,\ q_1 = 1.0$			$\alpha = 3.0,\ q_1 = 1.0$			$\alpha = 4.0,\ q_1 = 1.0$		
	itn	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$
1.0e-5	148	164	4.8e-07	90	124	1.7e-06	87	132	2.6e-07
1.0e-6	175	195	3.1e-08	107	144	1.0e-07	102	153	2.0e-08
1.0e-7	211	236	5.9e-10	133	179	7.3e-10	114	174	1.2e-09
1.0e-8	240	267	3.9e-11	159	211	2.4e-11	141	218	5.9e-12
1.0e-9	278	309	5.8e-13	185	247	4.6e-13	154	237	6.9e-13
1.0e-10	330	368	2.0e-15	223	294	3.3e-16	180	274	5.1e-15
ε_x	$\alpha = 2.0,\ q_1 = 0.8$			$\alpha = 3.0,\ q_1 = 0.8$			$\alpha = 4.0,\ q_1 = 0.8$		
	itn	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$
1.0e-5	68	114	1.3e-07	73	156	1.0e-07	63	153	3.3e-07
1.0e-6	71	120	3.7e-08	85	180	4.0e-09	75	175	9.2e-09
1.0e-7	80	135	3.6e-09	95	200	3.3e-10	753	175	9.2e-09
1.0e-8	102	167	8.6e-12	104	217	2.7e-11	96	219	3.8e-12
1.0e-9	105	170	2.3e-12	118	241	5.2e-13	106	236	2.6e-13
1.0e-10	110	176	9.1e-14	127	257	5.4e-14	114	253	1.0e-14

ТАБЛИЦА 2. Решение задачи **maxquad** с помощью программы **amplralg** с параметром рестарта 10^{-16}

ε_x	$\alpha = 2.0,\ q_1 = 1.0$			$\alpha = 3.0,\ q_1 = 1.0$			$\alpha = 4.0,\ q_1 = 1.0$		
	itn	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$
1.0e-5	147	166	-4.5e-07	93	122	1.4e-06	82	118	4.7e-07
1.0e-6	175	196	-7.6e-08	105	137	1.0e-07	95	137	3.3e-08
1.0e-7	227	257	-1.9e-08	131	180	1.1e-06	139	236	2.1e-09
1.0e-8	268	304	-5.6e-11	150	207	5.5e-10	178	294	4.0e-08
1.0e-9	297	335	-5.5e-13	197	266	2.5e-13	180	299	4.0e-09
1.0e-10	323	364	-2.6e-09	222	305	4.6e-10	198	331	4.9e-13
ε_x	$\alpha = 2.0,\ q_1 = 0.8$			$\alpha = 3.0,\ q_1 = 0.8$			$\alpha = 4.0,\ q_1 = 0.8$		
	itn	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$
1.0e-5	72	135	-4.6e-07	63	122	3.6e-07	80	215	2.6e-07
1.0e-6	80	147	-1.5e-08	75	137	6.2e-09	86	227	8.5e-08
1.0e-7	86	155	-4.6e-09	75	180	6.2e-09	103	272	9.8e-08
1.0e-8	93	164	-2.0e-10	93	207	3.8e-11	147	412	1.0e-11
1.0e-9	112	195	-1.8e-12	109	266	3.1e-09	147	412	1.0e-11
1.0e-10	118	204	-3.8e-13	254	305	4.3e-12	176	480	1.8e-10

ТАБЛИЦА 3. Решение задачи **maxquad** с помощью программы **amplralg** с параметром рестарта 10^{-50}

ε_x	$\alpha = 2.0,\ q_1 = 1.0$			$\alpha = 3.0,\ q_1 = 1.0$			$\alpha = 4.0,\ q_1 = 1.0$		
	itn	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$
1.0e-5	148	164	9.18553e-07	97	121	3.3685e-07	82	125	3.42152e-07
1.0e-6	175	195	4.97295e-08	109	140	1.00738e-07	89	136	5.81135e-08
1.0e-7	211	236	5.93078e-10	136	171	4.81436e-10	105	157	9.93776e-10
1.0e-8	240	267	3.93642e-11	163	205	8.11506e-12	120	178	9.28166e-11
1.0e-9	278	309	1.04095e-12	188	237	2.31815e-13	147	216	5.00933e-13
1.0e-10	330	368	4.88498e-15	212	271	2.9976e-15	185	218	2.22045e-16
ε_x	$\alpha = 2.0,\ q_1 = 0.8$			$\alpha = 3.0,\ q_1 = 0.8$			$\alpha = 4.0,\ q_1 = 0.8$		
	itn	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$	*itn*	*nfg*	$f_r - f^*$
1.0e-5	65	117	7.21527e-07	70	140	4.5375e-08	83	214	7.75685e-08
1.0e-6	80	144	9.29918e-09	70	140	4.5375e-08	83	214	7.75685e-08
1.0e-7	88	152	3.42361e-10	78	157	6.80858e-09	92	238	1.36016e-08
1.0e-8	88	152	3.42361e-10	87	175	8.42857e-10	102	264	2.30699e-10
1.0e-9	104	177	3.40872e-12	110	223	1.66001e-12	126	311	1.22347e-13
1.0e-10	118	198	3.34177e-14	134	269	8.88178e-16	136	330	8.21565e-15

Здесь для программы **amplralg** использовались два значения параметра восстановления матрицы B (рестарта): 10^{-16} и 10^{-50}. Первое значение используется программой по умолчанию, но как видно из табл. 2 это приводит к ухудшению сходимости из-за процедур восстановления (например, для $\alpha = 2.0$, $q_1 = 1.0$, ε_x = 1.0e-10). При использовании второго значения процедур восстановления не было, что позволило, как видно из табл. 3, достичь более точных решений.

2. Вычислительные эксперименты для примера sabs. Задача **sabs** состоит в минимизации кусочно-линейная функция следующего вида:

$$f(x) = \sum_{i=1}^{100} (1.2)^{i-1} |x_i - 1|, \quad f^* = f(x^*) = 0, \quad x^* = (1,1,\ldots,1)^T, \tag{2}$$

где $|a|$ – абсолютная величина числа a. Функция (2) – овражная, так как коэффициенты при $|x_i - 1|$, $i = 1,\ldots,100$ образуют геометрическую прогрессию с показателем $q = 1.2$, где минимальный коэффициент равен $(1.2)^0 = 1$, а максимальный – $(1.2)^{99} \approx 6.9015\text{e+}07$. Итерационный процесс запускается с начальной стартовой точки $x_0 = (0,0,\ldots,0)^T$. Начальный шаг $r(\alpha)$-алгоритма был выбран $h_0 = 10$ (равен $\|x_0 - x^*\|$ – расстоянию от стартовой точки x_0 до точки минимума x^*). Параметр останова по норме градиента $\varepsilon_g = 10^{-12}$, максимальное количество итераций *maxitn* = 5000. В табл. 4, 5 приведены результаты программ **amplralg** и **amplralgb5.** Здесь для программы **amplralg** использовалось значение параметра восстановления матрицы B (рестарта) 10^{-50}. Для вычисления субградиента используется функция $sign(x)$, которая была доопределена в точке 0 $sign(0) = 0$ – это делается в системах программирования Octave и Matlab.

ТАБЛИЦА 4. Решение задачи **sabs** с помощью программы **amplralg**

ε_x	$\alpha = 2.0$, $q_1 = 1.0$			$\alpha = 4.0$, $q_1 = 1.0$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	3970	3979	5.8e-007	2013	2046	7.1e-007
1.0e-9	4313	4322	6.6e-008	2178	2211	7.0e-008
1.0e-10	4602	4611	6.6e-008	2325	2358	7.3e-009
ε_x	$\alpha = 2.0$, $q_1 = 0.9$			$a = 4.0$, $q_1 = 0.9$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	205	336	4.4e+5	854	1732	3.7e-006
1.0e-9	205	336	4.4e+5	941	1921	7.3e-008
1.0e-10	205	336	4.4e+5	1000	2053	6.9e-009

ТАБЛИЦА 5. Решение задачи **sabs** с помощью программы **amplralgb5**

ε_x	$\alpha = 2.0,\ q_1 = 1.0$			$\alpha = 4.0,\ q_1 = 1.0$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	3983	3992	4.9e-07	2041	2073	6.6e-07
1.0e-9	4275	4284	6.1e-08	2192	2224	6.8e-08
1.0e-10	4607	4616	3.6e-09	2333	2365	7.2e-09
ε_x	$\alpha = 2.0\ ,\ q_1 = 0.9$			$\alpha = 4.0\ ,\ q_1 = 0.9$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	1695	3685	3.0e-07	956	1856	4.8e-07
1.0e-9	1715	3705	3.0e-08	975	1876	9.9e-08
1.0e-10	1737	3727	2.03e-09	1074	2075	1.1e-08

Как видно из табл. 4 программа **amplralg** не смогла найти оптимальные решение для случая $\alpha = 2.0,\ \ q_1 = 0.9$ из-за измельчения шага в процедуре адаптивного поиска шагового множителя. При изменении параметра $q_1 = 0.9$ на $q_1 = 0.95$ программа **amplralg** находит оптимальное решение.

3. Вычислительные эксперименты для примера squadr. Задача **squadr** состоит в минимизации функции, аналогичной (2), следующего вида:

$$f(x) = \sum_{i=1}^{100} (1.2)^{i-1} (x_i - 1)^2, \quad f^* = f(x^*) = 0, \quad x^* = (1,1,\ldots,1)^T . \tag{3}$$

Функция (3) как и (2) – овражная. Стартовая точка и параметры $r(\alpha)$ -алгоритма выбраны такими же как для примера 2. В табл. 6, 7 приведены результаты программ **amplralg** и **amplralgb5.**

ТАБЛИЦА 6. Решение задачи **squadr** с помощью программы **amplralg**

ε_x	$\alpha = 2.0,\ q_1 = 1.0$			$\alpha = 4.0,\ q_1 = 1.0$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	3379	4122	4.4e-017	1952	2644	1.4e-015
1.0e-9	3724	4532	5.7e-019	2091	2848	1.3e-017
1.0e-10	4092	4991	1.9e-019	2368	3252	9.8e-020
ε_x	$\alpha = 2.0\ ,\ q_1 = 0.9$			$\alpha = 4.0\ ,\ q_1 = 0.9$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	560	1046	5.1e-014	374	604	1.9e-015
1.0e-9	583	1071	1.1e-016	393	628	2.0e-017

1.0e-10	620	1116	3.8e-019	419	659	1.5e-019

ТАБЛИЦА 7. Решение задачи **squadr** с помощью программы **amplralgb5**

ε_x	$\alpha = 2.0,\ q_1 = 1.0$			$\alpha = 4.0,\ q_1 = 1.0$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	3379	4110	2.7e-16	4548	6422	4.0e-16
1.0e-9	3740	4550	2.0e-19	5000	7068	3.8e-17
1.0e-10	4101	4997	4.8e-21	5000	7068	3.8e-17
ε_x	$\alpha = 2.0\ ,\ q_1 = 0.9$			$\alpha = 4.0\,,\ q_1 = 0.9$		
	itn	*nfg*	f_r	*itn*	*nfg*	f_r
1.0e-8	563	978	5.8e-15	390	605	8.5e-16
1.0e-9	570	985	4.8e-16	412	630	3.1e-18
1.0e-10	601	1021	4.8e-18	431	653	3.1e-19

Как видно из табл. 7 для программы **amplralgb5** случай $\alpha = 4.0,\ q_1 = 1.0$ вызвал определенные вычислительные трудности: сходимость замедлилась и в двух случаях программа остановилась по критерию превышения максимального количества итераций.

4. Вычислительные эксперименты для примера edp [6, стр. 85]. Задача **edp** является задачей нахождения нагрузок энергетических объектов с фиксированными включенными энергоблоками. Эту задачу, используя метод негладких штрафных функций, можно представить в виде задачи безусловной негладкой минимизации. В данных расчетах функция затрат условного топлива задается выпуклой квадратичной функцией. Параметры задачи приведены в [6]. Параметры *r*-алгоритма выбраны следующими: максимальное количество итераций – 50000, $\varepsilon_x = 10^{-6}$, $\alpha = 4.0$, $q_1 = 0.95$, $q_2 = 1.1$, $h_0 = 100$ стартовая точка $x_0 = (0, 0, \ldots, 0)^T$. Штрафной коэффициент равен 10000. Результаты приведены в табл. 8.

ТАБЛИЦА 8. Решение задачи **edp**

amplralgb5				**amplralg**			
itn	*nfg*	f_r	T (сек)	*itn*	*nfg*	f_r	T(сек)
18323	38691	3021209.49076	152.895	21660	40422	3021209.49068	113.11

Как видно из табл. 8 для программы **amplralgb5** количество итераций и вычислений значений функции меньше, чем для **amplralg,** но при этом время выполнения *T* было больше.

Выводы. Результаты вычислительных экспериментов показали, что в целом обе программные реализации r-алгоритма при правильном выборе параметров позволяют эффективно и с достаточной точностью находить решения как гладких, так и негладких оптимизационных задач с высокой степенью «овражности». Планируется проведение дальнейших сравнительных вычислительных экспериментов на тестовых задачах с использованием системы AMPL. Перспективным является использование для ускорения обеих программ современных версий специализированных библиотек BLAS, LAPACK и т. п.

Работа выполнена при поддержке НАН Украины (проекты В.П.120.17 и ВК.120.21.18 – А.П. Лиховид) и Volkswagen Foundation (грант No 90 306 – А.В. Ивличев).

О.П. Лиховид, А.В. Івлічев

РЕЗУЛЬТАТИ ТЕСТУВАННЯ ДВОХ РЕАЛІЗАЦІЙ *R*-АЛГОРИТМУ

Наведено результати тестування двох реалізацій *r*-алгоритму для функцій, які мають високу ступінь «яружності». Результати обчислювальних експериментів показали, що обидві програмні реалізації *r*-алгоритму дозволяють ефективно і з достатньою точністю знаходити розв'язки таких оптимізаційних задач.

O.P. Lykhovyd, A.V. Ivlichev

RESULTS OF TESTING TWO IMPLEMENTATIONS OF R-ALGORITHM

The results of testing two implementations of r-algorithm for functions with a high degree of "ravine" are presented. The results of the computational experiments showed that both software implementations of r-algorithm allow efficient and accurate solution of such optimization problems.

Список литературы

1. Алгоритм минимизации негладких функционалов / $r(\alpha)$-алгоритм. АН УССР, РФАП. № 22. 1976.
2. http://www.incyb.kiev.ua/file/NonDiffOpt_Nurm/index_r.htm
3. Стецюк П.И. Методы эллипсоидов и r-алгоритмы. Кишинэу. Эврика, 2014. 488 с.
4. Стецюк П.И. Субградиентные методы ralgb5 и ralgb4 для минимизации овражных выпуклых функций. *Вычислительные технологии*. 2017. Т. 22, № 2. С. 127 – 149.
5. Nonsmooth optimization. Eds. C. Lemareshal, R. Mifflin. Oxford: Pergamon Press, 1978. 186 p.
6. Лиховид А.П., Фесюк А.В. Задачи нахождения оптимальных нагрузок энергетических объектов с нелинейными функциями стоимости. *Теорія оптимальних рішень*. Київ: Ін-т кібернетики імені В.М. Глушкова НАН України, 2014. С. 84 – 90.

Получено 15.03.2018