УДК 519.8

А.П. ЛИХОВИД

# ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ ПО ПРИМЕНЕНИЮ *R*-АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ОДНОГО КЛАССА ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

*Приведены результаты сравнительных вычислительных экспериментов по применению r-алгоритмов для нахождения локальных решений задач с равновесными ограничениями (MPEC) с использованием тестовой библиотеки MacMPEC.*



**Введение**. В настоящее время опубликовано много работ по нахождению локальных решению задач с равновесными ограничениями (mathematical programs with equilibrium constraints – MPEC), в которых они рассматриваются как задачи невыпуклого нелинейного программирования, а для нахождения локальных решений используются современные пакеты нелинейного программирования [1–2]. Показывается, что этот подход – перспективный, несмотря на вычислительные трудности, определяемые сложностью данного класса задач. С этой точки зрения вызывает интерес проведение сравнительного анализа эффективности нахождение локальных решений этого класса задач различными известными алгоритмами нелинейной оптимизации, в том числе и *r*-алгоритмом.

**Постановка задачи.** Рассмотрим постановку задач математического программирования с равновесными ограничениями (MPEC) в следующем виде [2]:

$$f(x) \to \min$$
$$0 \le g(x) \perp x \ge 0, \qquad (1)$$

где $x \in \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^n$, а обозначение $0 \le g(x) \perp x \ge 0$ – краткая запись для $x^T g(x) = 0$.

Задачу (1) можно рассматривать как задачу невыпуклого нелинейного программирования, например, в следующем виде:

$$\begin{gathered} f(x) \to \min \\ g(x) \geq 0, x \geq 0, \\ g_j(x) x_j \leq 0, \quad \forall j \in j. \end{gathered} \tag{2}$$

Попытаемся применить для нахождения ее локальных минимумов существующие алгоритмы нелинейного программирования. В этой работе будет рассмотрен подход к решению задач вида (2) с использованием метода точных штрафов, а для решения получившейся задачи недифференцируемой оптимизации предлагаетя использовать *r*-алгоримы [3]. Приведены сравнительные результаты численных экспериментов для одной реализации *r*-алгоритма и современных программ нелинейной оптимизации LOQO [4] и SNOPT [5] на известном наборе тестовых задач MacMPEC [6].

**Метод решения.** Рассмотрим задачу (2). Следуя методу точных штрафных функций [3], заменим исходную задачу следующей задачей безусловной оптимизации:

$$\min f(x) + \rho \max\{0, -g(x), -x, g_j(x) x_j\}. \tag{3}$$

Здесь $\rho > 0$ – штрафной множитель. Теория метода точных штрафных функций гарантирует, что при достаточно большом значении штрафного множителя решение задачи (3) эквивалентно решению задачи (2), т.е. каждое решение задачи (2) будет и решением задачи (3). Задача (3) – задача недифференцируемой оптимизации. Для ее решения можно использовать алгоритмы оптимизации субградиентного типа, например *r*-алгоритмы. Детальное описание применяемого *r*-алгоритма приведено в [3]. Далее будут представлены результаты численных экспериментов и оценки эффективности применения *r*-алгоритма для тестового набора задач.

**Численные экперименты.** Для выполнения численных экспериментов разработана программа на языке программирования C++, реализующая *r*-алгоритм с адаптивной регулировкой шага. Данная программа, которую обозначим как Ralg, была подключена к интерфейсу известной системы моделирования оптимизационных задач AMPL [7]. При этом задачи условной оптимизации автоматически приводятся к виду (3). Преимущества такого подхода состоят в следующем:

к системе AMPL подключены почти все известные оптимизационные программы, что позволяет легко проводить сравнительные эксперименты;

на языке моделирования этой системы реализованы многие известные наборы тестовых задач, предлагаемые на различных сайтах сети Интернет, что позволяет провести численные расчеты на большом количестве различных задач;

средства системы моделирования позволяют автоматизировать вычисления значений функций и производных.

Характеристики компьютера, на котором проводились исследования:

компьютер IBM PC Pentium;

процессор CPU Pentium 750MHz;

оперативная память 256MB;

операционная система Windows 2000.

Для проведения численных экспериментов была выбрана известная библиотека тестовых задач MacMPEC. Все задачи в этой библиотеке написаны на языке моделирования AMPL и доступны на сайте [6].

Для сравнительного анализа, вместе с $r$-алгоритмом, были выбраны известные программы LOQO и SNOPT, одни из лучших для решения задач нелинейной оптимизации. LOQO – одна из реализаций метода внутренних точек, а SNOPT – одна из реализаций квазиньютоновских методов. Все они подключены к интерфейсу системы моделирования AMPL. Использовались стандартные значения управляющих параметров для этих программ. Для $r$-алгоритма стандартные значения параметров для решения тестовых задач были выбраны следующими: $\alpha = 2$ – коэффициент растяжения пространства; $\|x_{k+1} - x_k\| \le 10^{-8}$ – точность при завершении по аргументу ($k$ – номер итерации); $h = 1$ – величина начального шага; значения параметров $q_1$, $q_2$ равнялись соответственно 1.1 и 0.95.

Для проведения вычислительных экспериментов из библиотеки MacMPEC была взята 61 тестовая задача. Максимальное число переменных – 278, а максимальное количество ограничений – 250. О размерности тестовых задач можно судить по данным табл. 1, где приведено распределение размерностей задач по заданным интервалам.

Важная характеристика – количество отказов, т.е. когда программа останавливается точке или по превышению заданного количества итераций, или из-за невозможности дальнейшего продолжения вычислений вследствие возникших численных затруднений. В табл. 2 приведено количество отказов для трех программ на исследуемом наборе тестовых задач.

ТАБЛИЦА 1

| Интервал | Распределение задач по количеству переменных | Распределение задач по количеству ограничений |
|---|---|---|
| 1 – 20 | 45 | 47 |
| 21 – 40 | 7 | 7 |
| 41 – 60 | 2 | 0 |
| 61 – 120 | 2 | 2 |
| 121 – 160 | 1 | 1 |
| 161 – 180 | 1 | 2 |
| 181 – 220 | 2 | 1 |
| 221 – 280 | 1 | 1 |

ТАБЛИЦА 2

| Программа | Количество отказов |
|---|---|
| SNOPT | 3 |
| LOQO | 12 |
| Ralg | 1 |

Из табл. 2 видно, что программа Ralg только в одном случае выдала сообщение об отказе, а наибольшее количество отказов оказалось у программы LOQO.

Далее представлено сравнение эффективности трех программ, использующее подход, описанный в [8]. Для сравнения использовалось время центрального процессора, требуемое для решения тестовых задач. Предположим, что мы сравниваем эффективность $n_s$ программ на наборе из $n_p$ тестовых задач. Пусть $t_{p,s}$ – время решения задачи $p$ программой $s$. Тогда вычислим следующие величины:

$$\rho_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : 1 \le s \le n_s\}}, \forall (p,s). \tag{4}$$

Если программа $s$ выдала отказ для задачи $p$, тогда $\rho_{p,s} = \infty$.

Для каждой программы $s$ строится график функции

$$\phi_s(\tau) = \frac{m_p}{n_p}, \tag{5}$$

где $m_p$ $(1 \le m_p \le n_p)$ – количество тестовых задач, удовлетворяющих $\rho_{p,s} \le \tau$.

Функция $\phi_s(\tau) : \mathbb{R} \to [0,1]$ представляет собой функцию распределения для величин (4) и служит как для оценки эффективности, так и надежности решеня тестовых задач программой $s$.

На рисунке показаны графики функций (5) для программ SNOPT, LOQO и Ralg, из которых видно, что наилучший результат показала программа SNOPT, а результаты программ LOQO и Ralg близки, при этом Ralg, немного уступая LOQO по быстродействию, имеет намного меньшее количество отказов.

Необходимо отметить, что все три программы для большинства тестовых задач нашли решения, совпадающие с представленными на сайте [6]. Для нескольких задач найдены допустимые точки, имеющие лучшие значения целевой функции, чем представленные на этом сайте. Более того, было найдено допустимое решение для задачи, объявленной на сайте как не имеющей допустимых решений.

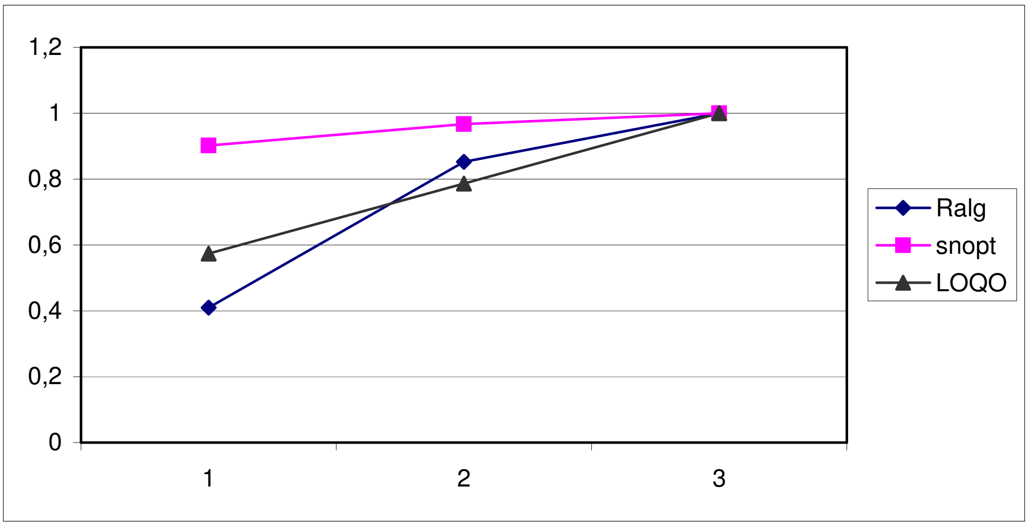


РИСУНОК. Графики функций (5) для программ SNOPT, LOQO и Ralg

**Заключение**. Из результатов вычислительных экспериментов можно сделать вывод, что лучшую производительность показала програма SNOPT, а программа Ralg, использующая метод точных штрафных функций и *r*-алгоритм, несколько уступая на данном тестовом наборе по быстродействию программе SNOPT, сравнима с ней по надежности нахождения решений и вполне работоспособна для нахождения локальных минимумов для задач данного класса средней размерности.

*О.П. Лиховид*

ОБЧИСЛЮВАЛЬНІ ЕКСПЕРИМЕНТИ ПО ВИКОРИСТАННЮ *R*-АЛГОРИТМІВ ДЛЯ РІШЕННЯ ОДНОГО КЛАСУ ЗАДАЧ НЕЛІНІЙНОГО ПРОГРАМУВАННЯ

Розглядається дослідження ефективності використання *r*-алгоритмів для рішення задач нелінійного програмування з рівноважними обмеженнями. Наведені результати порівняльних обчислювальних експериментів з використанням пакета AMPL та задач з бібліотеки MacPEC для програм SNOPT, LOQO та Ralg.

*O.P. Lykhovyd*

ON COMPUTATIONAL EXPERIMENTS WITH *R*-ALGORITHM TO SOLVE ONE CLASS OF NONLINEAR PROGRAMMING PROBLEMS

The paper investigates the efficiency of *r*-algorithms used to mathematical programs with equilibrium constraints. The comparative computational results for SNOPT, LOQO, Ralg solvers with AMPL package on test problems from MacMPEC library are given.

1. *Fletcher R., Leyffer S.* Numerical experience with solving MPECs as NLP. University of Dundee Technical Report NA-210, August 2002.
2. *Benson H.Y., Shanno D.F., Vanderbei R.J.* Interior-point methods for nonconvex nonlinear programming: complementarity constraints. Princeton University, 2002.
3. *Shor N.Z.* Nondifferentiable Optimization and Polynomial Problems. – Boston Dordrecht. – London: Kluwer Academic Publishers, 1998. – 412 p.
4. *Vanderbei R.J.* LOQO: An interior point code for quadratic programming // Technical Report SOR-94-15, School of Engineering and Applied Science, Department of Civil Engineering and Operations Research, Princeton University. – 1994.
5. *Gill P.E., Murray W, and Saunders M.A.* SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM J. on Optimization 12 (2002), 979-1006.
6. *Leyffer S.* MacMPEC Test Suite. www.maths.dandee.ac.uk/~sleyffer/MacMPEC/.
7. *Fourer R., Gay D., Kernighan B.* AMPL: A Modeling Language for Mathematical Programming. Duxbury Press-Brooks-Cole Publishing Company. – 1993. – 351 p.
8. *Dolan E.D., More J.J.* Benchmarking optimization software with performance profiles. Math. Programming, 91:201-214, 2002.

Получено 05.06.2006