УДК 519.8

А.П. ЛИХОВИД

О ПРИМЕНЕНИИ R-АЛГОРИТМА ДЛЯ РЕШЕНИЯ ОДНОГО КЛАССА ЗАДАЧ СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Рассматривается исследование эффективности применения r-алгоритмов для решения стохастических задач с простой рекурсией и случайной технологической матрицей ограничений второго этапа. Приведены результаты сравнительных вычислительных экспериментов с использованием пакета SLP-IOR.

Введение. Хорошо известным специальным видом двухэтапной стохастической задачи является модель с простой рекурсией [1]. Но эффективные алгоритмы решения и вычислительные результаты касались только моделей, когда вектор правых частей ограничений второго этапа был случайным [2]. В работе [3] был предложен подход к решению задач для случайной технологической матрицы ограничений второго этапа с применением r-алгоритмов [4]. В предлагаемой работе, которая является продолжением работы [3], представлены результаты вычислительных экспериментов эффективности применения r-алгоритмов для решения задач такого класса. Рассмотрим следующую постановку задачи:

$$\min_x [cx + Q(x)],$$
$$x \in X. \tag{1}$$

Здесь $X = \{x \in R_+^n : Ax \le b\}$ – детерминированные ограничения, ограничения на неотрицательность для переменных первого этапа; $Q(x) = E_\omega[v(h(\omega) - \mathrm{T}(\omega)x)]$ является функцией ожидаемых затрат, где ω – случайный вектор, $v(z)$ – целевая функция для задачи второго этапа:

$$\min_y [q^+ y^+ + q^- y^-],$$
$$y^+ - y^- = z, \quad z \in R^m,$$
$$y^+ \in R_+^m,\ y^- \in R_+^m. \tag{2}$$

Предположим, что $q_i^+ + q_i^- \geq 0,\ i = 1,\dots,m$, откуда функция v является ограниченной и выпуклой [2].

Обозначим отклонения $\eta_i(x,\omega) = T_i(\omega)x - h_i(\omega),\ i \in I$. Для каждого индивидуального отклонения штрафы q_i^+ и q_i^- назначаются, соответственно, за излишки $\eta_i(x,\omega)^+ = \max\{0, \eta_i(x,\omega)\}$ и дефицит $\eta_i(x,\omega)^- = \max\{0, -\eta_i(x,\omega)\}$. Целевую функцию можно представить в виде

$$cx + E_\omega\left[\sum_{i\in I}\left(q_i^+\eta_i(x,\omega)^+ + q_i^-\eta_i(x,\omega)^-\right)\right]. \tag{3}$$

Воспользуемся сепарабельностью функции ожидаемых затрат Q. Пусть ω_i обозначает компоненты вектора ω, от которых фактически зависят $T_i(\omega)$ и $h_i(\omega)$. Тогда $(T_i(\omega), h_i(\omega)) = (T_i(\omega_i), h_i(\omega_i))$, $i = 1,...,m$, а функция Q может быть записана в виде

$$Q(x) = \sum_{i=1}^{m} Q_i(x), \tag{4}$$

где

$$Q_i(x) = q_i^+ E_{\omega_i}[(h_i(\omega_i) - T_i(\omega_i)x)^+] + q_i^- E_{\omega_i}[(h_i(\omega_i) - T_i(\omega_i)x)^-],\ i = 1,\dots,m.$$

Здесь математическое ожидание берется относительно распределения ω_i. Учитывая, что $(z)^+ - (z)^- = z$, и, используя ранее введенное обозначение $\eta_i(x,\omega_i) = T_i(\omega_i)x - h_i(\omega_i)$, получаем следующее выражение:

$$\begin{aligned} Q_i(x) &= q_i^+ E_{\omega_i}[\eta_i(x,\omega_i)^-] + q_i^- E_{\omega_i}[\eta_i(x,\omega_i)^+] = \\ &= q_i^-(\bar{t}_i x - \bar{h}_i) + (q_i^+ + q_i^-)E_{\omega_i}[(\eta_i(x,\omega_i)^-], \end{aligned} \tag{5}$$

где $\bar{t}_i = E_{\omega_i}[T_i(\omega_i)]$, $\bar{h}_i = E_{\omega_i}[h_i(\omega_i)]$.

Предположим, что ω, следовательно, и ω_i имеют дискретное распределение $\Pr\{\omega_i = \omega_i^s\} = p_i^s$, $s \in S_i$. В этом случае получаем:

$$E_{\omega_i}[\eta(x,\omega_i)^-] = \sum_{s\in S_i}\max\{0, -p^s\eta(x,\omega_i^s)\},$$

$$\begin{aligned} Q_i(x) &= q_i^+ E_{\omega_i}[\eta_i(x,\omega_i)^-] + q_i^- E_{\omega_i}[\eta_i(x,\omega_i)^+] = \\ &= q_i^-(\bar{t}_i x - \bar{h}_i) + (q_i^+ + q_i^-)\sum_{s\in S_i}\max\{0, -p^s\eta(x,\omega_i^s)\}. \end{aligned} \tag{6}$$

Для учета ограничений первого этапа $x \in X$ можно использовать, например метод негладких штрафных функций [4]. Тогда, чтобы решить задачу (1) для случая дискретного распределения, можно применить алгоритмы субградиентного типа. Предлагаем в этом случае воспользоваться *r*-алгоритмами [4].

Есть несколько преимуществ в использовании данного подхода.

1. Исходя из сепарабельности функции Q, можно рассмотреть только $\sum_{i=1}^{m} S_i$ подзадач второго этапа вместо $\prod_{i=1}^{m} S_i$ (при условии, что строки являются независимыми), чтобы вычислить значение Q и субградиент для каждого текущего решения x_k.

2. Вместо решения каждой подзадачи второго этапа как задачи линейного программирования можно использовать аналитические формулы для вычисления $Q(x_k)$ и субградиента целевой функции в точке x_k.

Для проверки предлагаемого алгоритма была разработана программа SIMPLE_T на языке программирования С++. Вычислительные эксперименты проводились с использованием известной системы моделирования для задач стохастического программирования SLP-IOR [2]. Специализированных программ для задач с простой рекурсией и случайной технологической матрицей второго этапа, с которыми можно было бы сравнить результаты, в распоряжении не было, поэтому было проведено сравнение времени решения с программой для задач с фиксированной рекурсией DAPPROX v1.0 (Kall & Mayer, 2001) из пакета SLP-IOR, которая имела среди всех программ, входящих в этот пакет, лучшее быстродействие.

Характеристики компьютера, на котором проводились исследования:

- компьютер IBM PC;
- процессор CPU Pentium 750MHz;
- оперативная память 256MB;
- операционная система Windows 98.

Предлагаемый алгоритм был проверен на разнообразных тестовых задачах. Они представляют собой задачи линейного стохастического программирования с простой рекурсией и случайной технологической матрицей (правыми частями ограничений) второго этапа. Тестовые задачи генерировались с помощью средств системы моделирования SLP-IOR. Каждый пример refineD5–refineD20 генерировался на основе задачи REFINE из [1] с помощью аппроксимации всех непрерывных распределений исходной задачи дискретными распределениями с количеством реализаций соответственно равными $k = \{5,10,15,20\}$. При этом общее число реализаций для каждого примера, которые просматривала программа DAPPROX, равнялось k^6, а число реализаций, которые учитывались программой SIMPLE_T, равнялось k^3. Аналогично примеры mixD2–mixD12_5 генерировались на основе задачи Product Mix из [5], при этом для примеров mixD12_2 и mixD12_5 число случайных параметров равнялось 12, количество дискретных реализаций 2 и 5 соответственно, для остальных примеров из этой группы число случайных параметров равнялось 10, количество реализаций для дискретных аппроксимаций было соответственно $k = \{2,3,4,13\}$.

Параметры *r*-алгоритма для решения тестовых задач были выбраны следующими: $\alpha = 2$ – коэффициент растяжения пространства; $\|x_{k+1} - x_k\| \le 10^{-6}$ – точность при завершении по аргументу (k – номер итерации); $h = 10$ – величина начального шага . Параметры программы DAPPROX взяты по умолчанию.

В табл. 1 приведены параметры этих тестовых задач.

ТАБЛИЦА 1

Тестовая задача	m1	n1	m2	n2	NRand	S_i	Nrealiz
MixD2	1	4	2	4	10	2^5	2^{10}
MixD3	1	4	2	4	10	3^5	3^{10}
MixD4	1	4	2	4	10	4^5	4^{10}
MixD13	1	4	2	4	10	13^5	13^{10}
refine5	1	2	2	4	6	5^3	5^6
refine10	1	2	2	4	6	10^3	10^6
refine15	1	2	2	4	6	15^3	15^6
refine20	1	2	2	4	6	20^3	20^6
mixD12_2	1	5	2	4	12	2^6	2^{12}
mixD12_5	1	5	2	4	12	5^6	5^{12}

Здесь
m1 – число ограничений первого этапа;
n1 – число переменных первого этапа;
m2 – число ограничений второго этапа;
n2 – число переменных второго этапа;
S_i – число случайных реализаций для *i*-го ограничения второго этапа;
Nrand – число случайных параметров;
Nrealiz – общее число случайных реализаций.

В табл. 2 представлено время решения для некоторых тестовых задач, в табл. 3 приведены значения функционала при завершении выполнения соответствующей программы. Здесь введены следующие обозначения:
* – остановка программы из-за недостатка памяти;
F – задача не была решена.

ТАБЛИЦА 2

МОДЕЛИ	Время решения для задач с простой рекурсией (сек.)	
	SIMPLE_T	DAPPROX
mixD2	0.02	0.84
mixD3	0.03	2.3
mixD4	0.09	3.24
mixD13	24.09	28.1
refine15	0.26	173.46*
refine5	0.01	3.74
refine10	0.08	49.03
refine20	0.6	537.94*
mixD12_2	0.11	F
mixD12_5	2.193	F

ТАБЛИЦА 3

МОДЕЛИ	Значение функционала при завершении	
	SIMPLE_T	DAPPROX
mixD2	-22026.32	-22026.325
mixD3	-17857.756	-17857.756
mixD4	-27153.412	-27153.411
mixD13	-18018.69	-18018.698
refine15	115.877	115.876*
refine5	111.65	111.65
refine10	115.063	115.06
refine20	116.175	116.17*
mixD12_2	1363.82	F
mixD12_5	170.33	F

Заключение. Все тестовые задачи были успешно решены программой SIMPLE_T, реализующей предложенный алгоритм. Отметим, что программа SIMPLE_T эффективно справлялась с примерами mixD12_2 и mixD12_5, которые превышали возможности программы DAPPROX. Полученные результаты показывают на данных тестовых примерах, что специализированная программа, учитывающая сепарабельность модели, является более эффективным средством для решения данного класса задач стохастического программирования, чем существующие программы, предназначенные для решения задач с фиксированной рекурсией.

О.П. Лиховид

ПРО ВИКОРИСТАННЯ R-АЛГОРИТМУ ДЛЯ РІШЕННЯ ОДНОГО КЛАСУ ЗАДАЧ СТОХАСТИЧНОГО ПРОГРАМУВАННЯ

Розглядається дослідження ефективності використання r-алгоритмів для рішення стохастичних задач з простою рекурсією та випадковою технологічною матрицею обмежень другого етапу. Приведені результати обчислювальних експериментів з використанням пакету SLP-IOR.

A.P. Likhovid

USING THE R-ALGORITHM TO SOLVE ONE STOCHASTIC PROBLEM CLASS

The paper investigates the efficiency of r-algorithms used to solve stochastic problems with a simple recourse and a random technology matrix of second-stage constraints. The computational results are given for the case, when the SLP-IOR package is used.

1. *Kall P., Wallace S.W.* Stochastic programming. – London: John Wiley and Sons, 1994. – 225 p.
2. *Mayer J.* Stochastic Linear Programming Algorithms: A Comparison Based on a Model Manegement System. – Amsterdam: ODP, 1998. – 153 p.
3. *Лиховид А.П.* О реализации алгоритма решения одного класса двухэтапных задач с простой рекурсией // Теория оптимальных решений. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 2001. – С. 3–9.
4. *Shor N.Z.* Nondifferentiable Optimization and Polynomial Problems. – Boston Dordrecht. – London: Kluwer Academic Publishers, 1998. – 412 p.
5. *King A.J.* Stochastic programming problems: examples from the literature. In Yu. Ermoliev and R.J-B. Wets, editors, Numerical techniques for stochastic optimization. Springer-Verlag, Berlin, 1988. – **30**, P. 543–567.

Получено 25.03.2005