

УДК 004.42

Е.Б. Грибанова

## Стохастические алгоритмы решения обратных задач экономического анализа с ограничениями

Предложены стохастические алгоритмы решения обратной задачи, учитывающие ограничения аргументов функции и коэффициенты относительной важности. Для их реализации был использован объектно-ориентированный подход, что позволяет выполнять модификацию системы, решать многоуровневые задачи, применять различные методы решения задач каждого уровня. Приведен пример решения задачи формирования прибыли.

**Ключевые слова:** обратные вычисления, оптимизация, случайный поиск, объектно-ориентированный подход.

**doi:** 10.21293/1818-0442-2016-19-4-112-116

Под «обратной задачей» понимают определение набора исходных величин, обеспечивающего достижение заданного результата. Первоначально решением задач подобного рода занимался академик А.Н. Тихонов, исследуя свойства объектов, недоступных или неудобных для непосредственного изучения, т.е. осуществляя восстановление причин при их заданном следствии.

Позже Б.Е. Одинцовым была опубликована теория обратных вычислений [1], где под обратной задачей понимается определение приростов аргументов прямой функции (переменных функции), на основании целевого предписания человека в виде ее прироста и дополнительной информации: начальных значений аргументов и функции, коэффициентов относительной важности аргументов (далее в статье будут рассмотрены обратные задачи такого рода). Таким образом, изменение целевой функции происходит в большей степени за счет аргументов, имеющих большее значение коэффициента относительной важности. Например, для двух аргументов функции  $f(x, z)$  система уравнений будет иметь вид

$$\begin{cases} y \pm \Delta y = f(x \pm \Delta x(\alpha), z \pm \Delta z(\beta)), \\ \frac{\Delta x}{\Delta z} = \frac{\alpha}{\beta}, \end{cases}$$

где  $y, x, z$  – исходные значения функции и аргументов;  $\Delta y$  – необходимый прирост функции;  $\alpha, \beta$  – коэффициенты относительной важности аргументов  $x$  и  $z$  соответственно.

Решая данную систему относительно  $\Delta x$  и  $\Delta z$ , получим необходимые приросты аргументов.

Простота данного аппарата и актуальность решаемых с его помощью задач обусловили его применение в различных сферах: экономике, образовании [2–4]. На основе данного метода был также разработан модифицированный метод обратных вычислений [5], с помощью которого можно вычислить новые значения аргументов функции на основе начальных значений аргументов и результата и коэффициентов относительной важности. При нахождении решения используется линейное уравнение связи между аргументами, поэтому данный метод при

компьютерной реализации не требует выполнения проверки соответствия установленных коэффициентов важности поставленной цели.

Кроме того Б.Е. Одинцовым были исследованы задачи, когда на значения аргументов наложены ограничения и, следовательно, возможно возникновение дефицита ресурса. В работе [6] предложена итерационная процедура, основанная на последовательном увеличении результирующей переменной. До полного исчерпания ресурсов либо достижения заданного значения целевой функции повторяются следующие шаги: нахождение решения с помощью обратных вычислений; проверка соответствия полученного решения заданным ограничениям и перерасчет приростов в случае обнаружения дефицита в некотором ресурсе.

При числе аргументов больше двух задача усложняется и возникает необходимость использования процедуры свертки.

Использование стохастических методов при решении задач подобного рода позволяет избежать сложных вычислений и найти приближенное решение с учетом коэффициентов важности, ограничений аргументов, в том числе рассматривать ситуации, когда переменные могут принимать только целые значения либо значения из заданного набора.

Поскольку обратная задача может быть представлена в виде задачи глобальной оптимизации, в которой нужно минимизировать разницу между заданным значением целевой функции и полученным решением, были рассмотрены существующие работы, посвященные минимизации функции.

### Алгоритмы случайного поиска

Появление идеи использования случайных величин при поиске решения связывают с именем У.Р. Эшби. В нашей стране исследования алгоритмов случайного поиска берут начало в работах Л.А. Расстригина [7, 8]. Алгоритмы поиска подразделяют на ненаправленные (все случайные испытания строят независимо друг от друга) и направленные (испытания связаны между собой).

Наиболее простым методом решения задач глобальной оптимизации является метод ненаправленного случайного поиска. Он заключается в получе-

нии случайных значений аргументов из заданного интервала, расчете целевой функции и сравнении её величины с наилучшим из вычисленных. Если новое рассчитанное значение результата оказалось меньше, то осуществляется запоминание полученного решения. Таким образом, для функции одного аргумента последовательность шагов будет следующая:

Шаг 1. Генерирование на интервале  $[r; R]$  равномерно распределенной случайной величины  $x$ .

Шаг 2. Если  $f(x) < f_{\min}$  ( $f_{\min}$  – минимальное найденное значение функции), то происходит запоминание новой точки в качестве текущего решения  $f_{\min} = f(x)$ ,  $x_{\min} = x$ .

Шаги повторяются в течение заданного числа реализаций либо до получения решения с указанной точностью. Такой способ нахождения решения является реализацией метода проб и ошибок.

Данный алгоритм может быть совмещен с локальным поиском, когда из случайно выбранных точек осуществляется локальный спуск в ближайший минимум. Из найденных локальных минимумов выбирается точка с наименьшим значением [7].

Благодаря своей простоте и гибкости данный метод получил широкое распространение при решении различных задач. Например, в статье [9] рассматривается использование метода ненаправленного случайного поиска для решения комбинаторной задачи выбора оптимального портфеля биржевых опционов, что позволило получить целочисленные значения искомым величин.

К алгоритмам направленного поиска относят алгоритм парной пробы, с возвратом при неудачном шаге, наилучшей пробы и т.д. [8].

В алгоритме парной пробы по обе стороны от исходной точки делаются два поисковых шага случайной величины. После этого осуществляется переход в новую точку в направлении наилучшего значения функции. В алгоритме с возвратом при неудачном шаге задается начальная точка  $x$  и случайным образом осуществляется моделирование приращения  $dx$ . Если значение функции в новой точке  $x + dx$  лучше, чем в точке  $x$ , то осуществляется переход в эту точку. В некоторых работах предлагается исключать то направление, которое не приводит к улучшению значения функции. В статье [10] описывается простой алгоритм оптимизации (SOPT), в котором случайная величина приращения имеет не равномерное распределение, а распределение Гаусса. Также предлагается многократно вызывать алгоритм для выбора наилучшего решения из полученных.

Недостатком приведенных алгоритмов направленного поиска является то, что они в качестве решения могут определить локальный минимум, а не глобальный. В связи с этим разрабатываются различные модификации. К ним, в частности, можно отнести адаптивные алгоритмы. Например, таким алгоритмом является случайный поиск ARSET (Adaptive Random Search Technique) [11] и динамический случайный поиск DRASET (Dynamic Random

Search Technique) [12]. В адаптивном случайном поиске в зависимости от значения целевой функции пространство поиска сужается (когда происходит поиск наилучшего значения) или расширяется (когда найдено решение с приемлемой точностью), таким образом, уменьшается вероятность нахождения локального минимума вместо глобального из-за недостаточного исследования отдельных участков. В алгоритме DRASET после нахождения решения дополнительно осуществляется локальный поиск вокруг найденной точки для получения более точного значения.

Также существуют алгоритмы поиска глобального минимума, которые используют процедуру случайного блуждания [7], например, метод «зашумления» градиента и метод сглаживания, который может быть использован в случае, если минимизируемая функция образована путем наложения на унимодальную функцию мелких отклонений; воспроизводят поведение живой и неживой природы [13, 14]; основаны на интервальных подходах [15, 16].

К недостаткам метода случайного поиска относят необходимость выполнения большого числа итераций для получения решения с заданной точностью, что требует затрат вычислительных ресурсов, а также существующую погрешность вычислений.

Данная работа посвящена разработке алгоритмов случайного поиска для решения обратной задачи. Для реализации алгоритмов предполагается использование объектно-ориентированного подхода.

### Стохастические алгоритмы решения обратной задачи

Взаимосвязь показателей может быть представлена в виде дерева, где на нулевом уровне расположено значение результирующей функции, а на нижних – аргументы. В свою очередь каждый лист этого дерева может быть результирующим показателем (рис. 1).

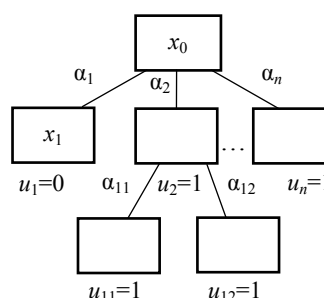


Рис. 1. Представление задачи в виде дерева

Узлы дерева имеют следующие характеристики:

- начальное значение  $x$ ;
- коэффициент относительной важности  $\alpha$  (сумма коэффициентов относительной важности аргументов одного уровня и относящихся к одному результирующему показателю должна быть равна единице);
- минимальное  $r$  и максимальное  $R$  значения, которые может принимать данный показатель;

• индикатор  $u$ , характеризующий возможность использования данного элемента, и принимающий два значения: 1 (использование возможно) и 0 (использование невозможно).

Коэффициент относительной важности  $\alpha$  указывает степень изменения результирующего показателя за счет данного аргумента. Он может отражать предпочтение исследователя в определении величин, а также рассчитываться на основе данных за предыдущие периоды и, таким образом, показывать наиболее вероятные значения аргументов для достижения результата.

Значение индикатора  $u$  становится равным нулю в случае, если изменение аргумента не может быть выполнено из-за существующего ограничения или отсутствия положительного изменения целевой функции. Также данный индикатор устанавливается равным нулю для величин-констант.

Общая задача может включать несколько подзадач, решение которых осуществляется последовательно путем обхода вершин дерева сверху вниз. Так, для задачи на рис. 2 сначала будут вычислены значения  $x_1, x_2, x_n$ , обеспечивающие заданную величину результата  $x_0$ , далее –  $x_{21}, x_{22}$  для получения значения  $x_2$ . Ниже будут описаны процедуры для решения одной подзадачи. При этом начальное значение корня рассматриваемого поддерева будем обозначать  $y = f(x)$ , а его значение, которое необходимо получить, –  $y^*$ .

Для решения обратной задачи были разработаны два алгоритма. Первый представляет собой модификацию случайного поиска, второй – итерационной процедуры, основанной на приращении функции.

Чтобы случайный поиск можно было использовать для решения обратной задачи, необходимо:

- учесть коэффициенты важности аргументов;
- представить обратную задачу в виде задачи глобальной оптимизации, где нужно минимизировать разницу между полученным решением и искомым  $y^*$ .

Для этого был использован интегральный показатель, который отражает степень достижения глобального минимума и соответствие приростов аргументов коэффициентам важности. Таким образом, алгоритм может быть представлен в виде следующих шагов:

Шаг 1. Генерирование на интервалах  $[r_i, R_i]$  равномерно распределенных случайных величин  $x_{t,i}$ , ( $i=1..n$ ,  $n$  – количество аргументов). Расчет значения функции  $y_t = f(x_t)$ .

Шаг 2. Вычисление интегрального показателя:

$$c = \frac{|y_t - y^*|}{|y - y^*|} + \sum_{i=1}^n \frac{|\Delta x_i|}{\sum_{i=1}^n |\Delta x_i|} - \alpha_i,$$

где  $\Delta x_i = x_{t,i} - x_i$ .

Первая часть слагаемого принимает минимальное значение, равное нулю, когда величина резуль-

тирующего показателя будет близка заданному  $y^*$ , вторая часть – при соответствии приростов коэффициентам важности.

Шаг 3. Сравнение с наилучшим значением интегрального показателя: если  $c < c_{\min}$ , то новое решение запоминается в качестве текущего  $y_{\min} = f(x_t)$ ,  $x_{\min i} = x_{t,i}$ .

Критерием останова является выполнение заданного числа итераций либо получение решения с указанной точностью.

Рассмотрим теперь алгоритм, основанный на моделировании приращения функции.

Устанавливается шаг приращения аргументов  $\Delta y$ .

Шаг 1. Установить новое значение результирующего показателя  $y_t = y_t + \Delta y$ .

Шаг 2. С помощью алгоритма моделирования полной группы несовместных событий выбрать узел из вершин-потомков, для которых значения индикатора равно 1, в соответствии с коэффициентами важности  $\alpha_1, \alpha_2, \dots, \alpha_n$ . Для этого выполняется расчет нормированных значений вероятностей по формуле

$$p_i^* = \alpha_i / \sum_{j=1}^n \alpha_j,$$

где  $j$  – номер вершины, для которой значения индикатора равно 1.

Если узел не найден, то осуществляется завершение работы алгоритма.

Шаг 3. Определяется значение  $x_{t,k}^*$  выбранной на предыдущем шаге вершины  $k$ , при фиксированных значениях остальных величин для получения заданного  $y_t$ .

Шаг 4. Проверка соответствия ограничению  $r_k \leq x_{t,k}^* \leq R_k$ . Если условие выполняется, то  $x_{t,k} = x_{t,k}^*$ , и всем вершинам, не являющимся константой, присваивается индикатор, равный единице, а иначе –  $f_k = 0$ . Переход к шагу 1.

Шаг 5. Проверка условия:  $y^* = y_t$ . Если условие выполняется, происходит завершение работы алгоритма, иначе – переход к шагу 1.

Полученные значения  $x_t$  будут решением задачи.

Для реализации представленных алгоритмов был использован объектно-ориентированный подход. На рис. 2 представлена диаграмма основных классов. Класс «Узел» представляет собой вершину дерева, которая связана с другими узлами. Класс «Метод» является абстрактным, его потомки «Приращение функции» и «Случайный поиск» предназначены для решения обратной задачи с использованием соответствующих стохастических алгоритмов. Класс «Модель» предназначен для хранения функций, связанных с расчетом результирующего показателя.

Данная структура, полученная путем модификации существующей [17], предоставляет возможность дальнейшего развития системы: добавления

новых методов, моделей; решения многоуровневых задач путем последовательного обхода всех уровней графа (сверху вниз), а также применения различных методов решения подзадач.



Рис. 2. Структура классов

В качестве примера рассмотрим модель формирования прибыли, приведенную в [5]:

$$profit = revenue - const\ costs - variable\ costs,$$

где *profit* – прибыль; *revenue* – выручка; *const costs* – постоянные затраты; *variable costs* – переменные затраты.

Исходные данные (тыс. руб.): *profit* = 200, *revenue* = 400, *const costs* = 50, *variable costs* = 150. Ставится задача определения уровня выручки, переменных и постоянных затрат для увеличения прибыли на 150 тыс. руб. При этом коэффициенты относительной значимости равны:  $\alpha=0,8$ ;  $\beta=0,1$ ;  $\gamma=0,1$ .

Данная задача может иметь более одного решения в зависимости от направления изменения величин. Например, увеличение прибыли может произойти за счет увеличения выручки, увеличения постоянных и переменных затрат, а также за счет увеличения выручки и уменьшения затрат. В табл. 1 представлены результаты для случая, когда увеличение выручки сопровождается снижением затрат.

Таблица 1

#### Результаты решения задачи

Метод поиска решения	Значение функции	Значения аргументов		
		Выручка	Постоянные затраты	Переменные затраты
Модифицированный метод обратных вычислений	350	520	35	135
Случайный поиск (1 млн. точек)	350,04	519,94	34,97	134,93
Приращение функции (шаг = 1)	350	521	36	135

В модифицированном методе обратных вычислений направление изменения аргументов указывается с помощью вида зависимости: прямая или обратная. Так, результаты табл. 1 получены для случая, когда между выручкой и затратами обратная зависимость, а между постоянными и переменными – прямая. В методе случайного поиска направление изменения можно получить путем установления соответствующих нижних и верхних границ (иначе решения в разных реализациях могут значительно отличаться). Для задачи были использованы следующие ограничения:

$$400 \leq revenue \leq 800,$$

$$10 \leq const\ costs \leq 50,$$

$$120 \leq variable\ costs \leq 150.$$

При использовании алгоритма на основе приращения функции возможно нахождение решения только для случая, когда происходит наименьшее изменение аргумента с наибольшим значением коэффициента относительной важности. Например, в табл. 2 представлены варианты увеличения результата в случае аддитивной модели двух аргументов. Для суммы двух аргументов увеличение результирующего показателя можно получить двумя способами: увеличив оба аргумента и увеличив более значительный аргумент с наибольшим значением коэффициента важности, уменьшив при этом аргумент с меньшим значением коэффициента относительной значимости. Метод приращения функции позволяет найти решение только для первого случая. Аналогично для второго варианта модели (разность аргументов) метод приращения функции выполнит поиск решения только в случае увеличения аргумента с наибольшим значением коэффициента важности при уменьшении значения аргумента с наименьшим значением коэффициента значимости.

Таблица 2

Варианты достижения цели		
Модель	Прирост результата	
$A(\alpha) + B(\beta), \alpha > \beta$	$A^+, B^+$	$A^+, B^-$
$A(\alpha) - B(\beta), \alpha > \beta$	$A^+, B^+$	$A^+, B^-$

#### Заключение

На основе простого случайного поиска и итерационной процедуры приращения функции разработаны алгоритмы решения обратной задачи. Их преимуществом является простота компьютерной реализации, в случае их использования отсутствует необходимость решения системы уравнений, применения процедуры свертки и многократного решения задачи с помощью обратных вычислений в случае, когда число аргументов больше двух. Кроме того, случайный поиск позволяет получить решение с учетом коэффициентов относительной важности и ограничений на величины аргументов, что невозможно получить только с помощью обратных вычислений. К недостаткам можно отнести существенную погрешность вычислений из-за использо-

вания стохастических методов, метод на основе приращения функции может быть использован лишь в некоторых вариантах изменения аргументов: когда сумма модулей приращений аргументов минимальна. Для реализации алгоритмов был использован объектно-ориентированный подход, структура представляется в виде дерева, на нижнем уровне которого приводятся аргументы, а на верхнем – рассчитываемый на их основе результирующий показатель. Данная структура предоставляет возможность изменения и дальнейшего развития системы, а также решения многоуровневых задач и использования различных методов решения подзадач. В качестве примера приводится задача формирования прибыли. Представленные алгоритмы и могут быть использованы при решении обратных задач в экономике.

#### Литература

1. Одинцов Б.Е. Обратные вычисления в формировании экономических решений. – М.: Финансы и статистика, 2004. – 256 с.
2. Виштак О.В. Использование технологии обратных вычислений при мониторинге качества дополнительного образования в вузе / О.В. Виштак, И.А. Штырова // Вестник Астрахан. гос. техн. ун-та. – 2014. – № 2. – С. 67–73.
3. Бармина Е.А. Мониторинг качества коммерческой организации. Структурирование показателей. Применение когнитивных карт / Е.А. Бармина, И.Ю. Квятковская // Вестник Астрахан. гос. техн. ун-та. – 2010. – № 2. – С. 15–20.
4. Мартянова А.В. Управление эффективностью банка на базе обратных вычислений // Вестник магистратуры. – 2015. – № 6(45). – С. 77–79.
5. Грибанова Е.Б. Решение обратных задач экономики с помощью модифицированного метода обратных вычислений // Проблемы управления. – 2016. – №5. – С. 35–40.
6. Одинцов Б.Е. Итерационный метод оптимизации управления предприятиями средствами обратных вычислений / Б.Е. Одинцов, А.Н. Романов // Вестник Финансового университета. – 2014. – № 2. – С. 60–73.
7. Растрингин Л.А. Адаптация сложных систем. – Рига: Зинатне, 1981. – 375 с.
8. Растрингин Л.А. Статистические методы поиска. – М.: Наука, 1968. – 376 с.
9. Мицель А.А. Комбинаторная модель опционного портфеля / А.А. Мицель, М.Е. Семенов, М.Э. Фатьянова // Финансовая аналитика: проблемы и решения. – 2016. – № 25(307). – С. 2–13.
10. Thomas J. Improved simple optimization algorithm for unconstrained non-linear optimization problems / J. Thomas, S. Mahapatra // Perspectives in Science. – 2016. – No. 8. – PP. 1–3.
11. Hamzacebi C. Continuous function minimization by dynamic random search / C. Hamzacebi, F. Kutay // Applied Mathematical Modeling. – 2007. – Vol. 31, No. 10. – PP. 2189–2198.
12. Hamzacebi C. A heuristic approach for finding the global minimum: adaptive random search technique / C. Hamzacebi, F. Kutay // Applied Mathematics and Computation. – 2006. – Vol. 173. – PP. 1323–1333.
13. Toksari M.D. Ant colony optimization for finding the global minimum // Applied Mathematics and computation. – 2006. – No. 176. – PP. 308–316.
14. Toksari M.D. A heuristic approach to find the global optimum of function // Journal of computational and Applied mathematics. – 2007. – Vol. 209, No. 2. – PP. 160–166.
15. Жиглявский А.А. Методы поиска глобального экстремума / А.А. Жиглявский, А.Г. Жилинскас. – М.: Наука, 1991. – 248 с.
16. Панов Н.В. Объединение стохастических и интервальных подходов для решения задач глобальной оптимизации функций // Вычислительные технологии. – 2009. – № 5(14). – С. 49–65.
17. Грибанова Е.Б. Информационная система рейтинговой оценки объектов экономики / Е.Б. Грибанова, А.Н. Алимханова, П.Э. Тугар-оол // Доклады Том. гос. ун-та систем управления и радиоэлектроники. – 2016. – № 2(19). – С. 51–55.

---

#### Грибанова Екатерина Борисовна

Канд. техн. наук, доцент каф. автоматизированных систем (АСУ) ТУСУРа  
Тел.: +7 (382-2) 70-15-36  
Эл. почта: katag@yandex.ru

Gribanova E.B.

#### Stochastic algorithms to solve the economic analysis inverse problems with constraints

Stochastic algorithms for solving the inverse problems are considered. These simple algorithms allow taking into account the limitations of the arguments of the function and the coefficients of relative importance. Object-oriented model implementation allows to modify the system, to solve multi-level problems, and to apply different methods for each level. The article presents a sample solution for profit generation.

**Keywords:** inverse computations, optimization, random search, object-oriented approach.