

ИСПОЛЬЗОВАНИЕ МОДИФИЦИРОВАННОГО АЛГОРИТМА DIRECT ДЛЯ РЕШЕНИЯ ОБРАТНОЙ ЗАДАЧИ БКЗ

Нечаев О. В., Эпов М.И.

Институт нефтегазовой геологии и геофизики СО РАН

В данной работе для поиска области эквивалентных решений обратной задачи БКЗ рассматривается модифицированный алгоритм глобальной оптимизации DIRECT. Главным достоинством семейства алгоритмов DIRECT является сбалансированное сочетание глобального и локального поиска. Эффективность предложенного алгоритма численно исследуется на ряде модельных задач.

Ключевые слова: БКЗ, обратная задача, алгоритм глобальной оптимизации DIRECT.

ВВЕДЕНИЕ

Решение поставленной по результатам бокового каротажного зондирования задачи идентификации параметров модели околоскважинного пространства, может быть сформулировано как задача минимизации некоторой целевой функции. В силу неточности задания модели околоскважинной среды, возникает необходимость в качестве результата минимизации целевой функции рассматривать не отдельные точки, доставляющие минимум, а область, возможно многосвязную, внутри которой целевая функция принимает значения меньше некоторого заданного порога — область эквивалентных решений. Наиболее распространенным на данный момент численным методом поиска области эквивалентных решений является прямой перебор параметров целевой функции на заданной сетке. Такой подход обладает экспоненциальной сложностью относительно размерности обратной задачи и его применение для практического решения многомерных обратных задач требует больших вычислительных затрат. В данной статье для решения обратной задачи БКЗ (Бокового Каротажного Зондирования) предлагается и численно исследуется модифицированный алгоритм DIRECT. Впервые алгоритм DIRECT для глобальной минимизации целевой функции был предложен в [Jones D.R., Perttunen C.D., and Stuckman B.E. Lipschitzian optimization without the Lipschitz constant // Journal of Optimization Theory and Applications, 1993, v. 79, № 1, p. 157-181.]. В последующем рядом авторов было разработано несколько его модификаций. Преимуществом данного семейства алгоритмов является то, что для нахождения глобального экстремума целевой функции не требуется знать значение производной данной функции. Высокая эффективность алгоритмов DIRECT заключается в сбалансированном сочетании глобального и локального поиска. Еще одним преимуществом данного подхода является хорошая распараллеливаемость алгоритма DIRECT.

Под гиперинтервалом $[\vec{a}, \vec{b}]$ будем понимать множество точек $\vec{x} \in \mathbb{R}^n$ таких, что $a_i \leq x_i \leq b_i$, $i = 1, \dots, n$, где \vec{a} и \vec{b} являются нижней и верхней точками гиперинтервала. Под размером гиперинтервала будем понимать длину его главной диагонали (расстояние между точками a и b). Например, при $n = 1$ гиперинтервал является отрезком, при $n = 2$ - прямоугольником, при $n = 3$ - параллелепипедом. Введем операцию дробления гиперинтервала $[\vec{a}, \vec{b}]$ на три новых гиперинтервала: $[\vec{a}, \vec{c}]$, $[\vec{d}, \vec{e}]$, $[\vec{f}, \vec{b}]$. Координаты точек определяются следующим образом:

$$\begin{aligned}\vec{a} &= (a_1, \dots, a_i, \dots, a_n), \\ \vec{c} &= \left(b_1, \dots, a_i + \frac{1}{3}(b_i - a_i), \dots, b_n \right), \\ \vec{d} &= \left(a_1, \dots, a_i + \frac{1}{3}(b_i - a_i), \dots, a_n \right), \\ \vec{e} &= \left(b_1, \dots, a_i + \frac{2}{3}(b_i - a_i), \dots, b_n \right),\end{aligned}$$

$$\vec{f} = \left(a_1, \dots, a_i + \frac{2}{3}(b_i - a_i), \dots, a_n \right),$$

$$\vec{b} = (b_1, \dots, b_i, \dots, b_n),$$

где $i = \arg \max_{j=1, \dots, n} (b_j - a_j)$, таким образом, производится дробление стороны гиперинтервала, которая обладает максимальной длиной. Пример дробления двумерного гиперинтервала (прямоугольника) приведен на рис.

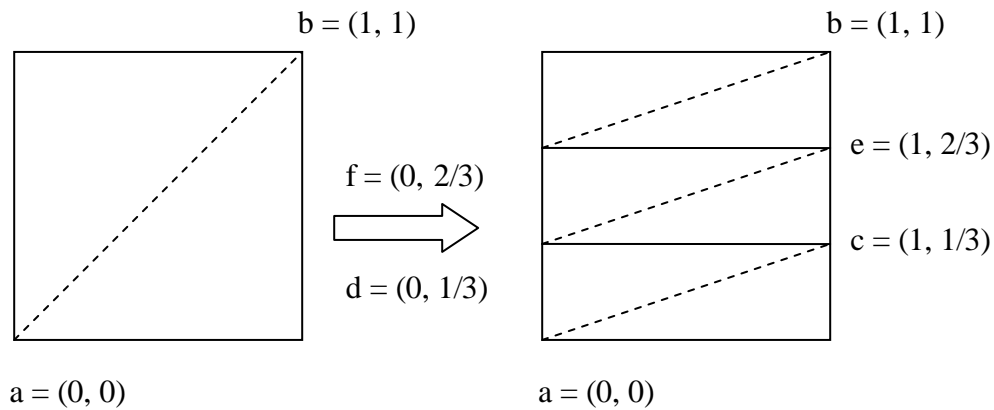


Рис. Дробление двумерного гиперинтервала.

Рассмотрим задачу минимизации функции $f(\vec{x})$ при $\vec{x} \in [\vec{a}, \vec{b}]$. Выполним процедуру разбиения гиперинтервала $[\vec{a}, \vec{b}]$ на три новых. Каждому из трех гиперинтервалов поставим в соответствие значение функции $f(\vec{x})$, вычисленное в центре соответствующего гиперинтервала. Таким образом, мы получили начальное разбиение области поиска. Минимизацию нашей функции будем осуществлять с использованием следующего модифицированного алгоритма DIRECT:

1. Разделить множество всех гиперинтервалов на подмножества, содержащие гиперинтервалы одинакового размера.
2. В каждом из полученных подмножеств найти гиперинтервал с минимальным значением функции в центре.
3. Выполнить дробление найденных в пункте 2 гиперинтервалов.
4. Установить в качестве текущего минимума функции f_{\min} наименьшее из значений функции $f(\vec{x})$, вычисленных в центрах всех имеющихся на данный момент гиперинтервалов.
4. Установить в качестве верхнего порога области эквивалентности решений число $f_{\min} \cdot C$, где C - некоторая заранее заданная константа, т.е. \vec{x} принадлежит области эквивалентности, если $f(\vec{x}) < f_{\min} \cdot C$.
6. Если за последние N (параметр N задан заранее) итераций не было найдено новое значение f_{\min} или новая точка \vec{x} , принадлежащая области эквивалентности, прекратить выполнение алгоритма, иначе перейти на пункт 1.

Согласно пункту 3 приведенного алгоритма, будем получать гиперинтервалы все меньшего и меньшего размера, что приведет к увеличению количества подмножеств, содержащих гиперинтервалы одинакового размера и к росту количества вычислений функции $f(\vec{x})$ на каждой итерации алгоритма. Для сокращения вычислительных затрат введем минимально возможный размер гиперинтервала. Заменим пункт 3 алгоритма на следующий

3. Выполнить дробление найденных в пункте 2 гиперинтервалов, если их размер больше установленного минимально возможного размера.

Для выполнения пункта 3 алгоритма необходимо вычислить значения функции $f(\bar{x})$ в центрах новых гиперинтервалов. Данные вычисления могут быть выполнены параллельно.

Результатом работы алгоритма является множество точек с вычисленными в них значениями функции $f(\bar{x})$, причем плотность точек в окрестностях минимумов функции $f(\bar{x})$ будет возрастать. В предельном случае работа алгоритма остановится, когда все полученные гиперинтервалы достигнут минимально возможного размера. В этом случае множество центров гиперинтервалов, в которых вычисляется значения функции $f(\bar{x})$, будут образовывать регулярную сетку и результаты работы алгоритма будут эквивалентны поиску минимума функции $f(\bar{x})$ на данной регулярной сетке. Фактически, алгоритм DIRECT задает порядок перебора узлов при поиске минимума функции на сетке. Поскольку алгоритм на каждой итерации выбирает наиболее подходящие участки для поиска минимума, это дает возможность не осуществлять полного перебора всех точек сетки.

Пусть в результате работы алгоритма сформировано множество точек с вычисленными в них значениями функции $f(\bar{x})$. Отбросим все точки, значения $f(\bar{x})$ в которых больше заранее заданного $f_{\min} \cdot C$. Оставшиеся точки будут принадлежать области эквивалентных решений. На основании полученного множества точек можно построить (например, применив аппарат кластерного анализа) набор гиперинтервалов, покрывающих с некоторой точностью область эквивалентности решения. Это становится возможным благодаря тому, что семейство алгоритмов DIRECT производит глобальную оптимизацию функции и позволяет найти все глобальные минимумы функции.

Эффективность разработанного модифицированного алгоритма глобальной оптимизации DIRECT исследовалась на ряде тестовых задач. Для каждой тестовой задачи была заранее задана модель околоскважинной среды, для которой при помощи математического моделирования были получены данные каротажного зондирования БКЗ. Результаты прямого моделирования использовались для построения целевой функции $f(\bar{x})$. Для сокращения вычислительных затрат в пространстве поиска использовались логарифмические координаты. Параметры алгоритма задавались следующим образом: $C = 1.2$, $N = 4 \cdot \{\text{количество идентифицируемых параметров}\}$. Минимально возможный размер гиперинтервала (см. пункт 3 алгоритма) выбирался так, чтобы сетка, в узлах которой производился поиск решения, состояла из 27 узлов по каждому из измерений.

Рассмотрим тестовую задачу. В тестовой задаче задавались следующие параметры скважины: УЭС скважины - 2 Ом, радиус скважины — 0.1 м.

Введем следующие условные обозначения: УЭС_{зп} - удельное электрическое сопротивление зоны проникновения, Ширина_{зп} - ширина зоны проникновения, УЭС_{оз} - удельное электрическое сопротивление окаймляющей зоны, Ширина_{оз} - ширина окаймляющей зоны, УЭС_{пл} - удельное электрическое сопротивление пласта.

В табл. 1 приведены параметры искомой модели, а так же верхняя и нижняя точки гиперинтервала, в которых выполнялась минимизация целевой функции.

Таблица 1. Параметры искомой модели.

	УЭС _{зп} , Ом	Ширина _{зп} , м	УЭС _{оз} , Ом	Ширина _{оз} , м	УЭС _{пл} , Ом
Известное решение	15	0.5	-	-	5
Верхняя граница поиска	28	1.0	28	1.0	28
Нижняя граница поиска	2	0.1	2	0.1	2

В приведенном тесте известная модель околоскважинной среды состоит из зоны проникновения и пласта, в то время как при решении обратной задачи используется модель, состоящая из зоны проникновения, окаймляющей зоны и пласта. Таким образом, мы априори имеем большую зону эквивалентности решений. Для решения приведенной задачи алгоритм выполнил 4015 решений прямой задачи, что позволило существенно сократить вычислительные затраты, т.к. при использовании для решения тестовой задачи прямого перебора на секте требуется решить 27^5 прямых задач.

В табл. 2 для каждого из найденных гиперинтервалов, покрывающих область эквивалентности решения, приведены наилучшие решения и соответствующие им значения целевой функции.

Таблица 2. Решения обратной задачи.

$f(\bar{x})$	УЭС _{зп} , Омм	Ширина _{зп} , м	УЭС _{оз} , Омм	Ширина _{оз} , м	УЭС _{пл} , Омм
0.025	14.8	0.484	5.58	0.958	4.59
0.027	14.8	0.626	2.81	0.225	4.59
0.028	14.8	0.408	9.09	0.344	4.59
0.028	14.8	0.681	2.32	0.375	4.59
0.029	14.1	0.550	5.31	0.464	4.37
0.029	14.8	0.245	13.45	0.375	4.59

Эффективность модифицированного алгоритма DIRECT для решения обратной задачи БКЗ обусловлена линейной вычислительной сложностью одной итерации алгоритма от количества идентифицируемых параметров обратной задачи. Количество итераций, выполняемых алгоритмом, а значит и время решения задачи, зависит от размера области эквивалентных решений (от количества узлов сетки, попадающих в данную область). Точность полученного решения с одной стороны зависит от размера ячеек сетки, на которой осуществляется поиск решения, а с другой стороны – от параметра алгоритма N . Выбор значения параметра N влияет на общее время работы алгоритма и точность определения границ области эквивалентных решений. С ростом значения N увеличивается количество исследованных узлов сетки и время работы алгоритма. Как показало проведенное численное исследование, модифицированный алгоритм DIRECT, предложенный для решения обратной задачи БКЗ, с одной стороны позволяет находить область эквивалентности решений, а с другой – его вычислительная эффективность гораздо выше метода полного перебора параметров на сетке.