

АЛГОРИТМ ИМИТАЦИИ ОТЖИГА ДЛЯ ЗАДАЧ НЕПРЕРЫВНОЙ ОПТИМИЗАЦИИ

А.В. Лисин

В статье описывается подход к использованию алгоритма имитации отжига к решению задач безусловной многомерной непрерывной оптимизации.

Ключевые слова: алгоритм имитации отжига, непрерывная оптимизация.

ВВЕДЕНИЕ

Пусть задана целевая функция $f: S \rightarrow R$, где S — континуальное множество и $S \subseteq R^n$. Требуется найти $x^* \in S$, такое что $f(x^*) \leq f(x)$, $\forall x \in R^n$ (в данной статье мы ограничимся рассмотрением задачи минимизации). Чтобы гарантировать существование x^* , будем также предполагать множество S компактным [1].

Впервые описанный в [2], алгоритм имитации отжига представляет собой метаэвристический алгоритм нахождения глобального минимума целевой функции. В общем виде алгоритм может быть описан следующим образом:

Шаг 0. Выбирается стартовая точка $x_0 \in S$ и параметр t_0 .

Шаг 1. Выбирается вектор $y_{k+1} \in S$ в соответствии с заданным правилом $D(\cdot, x_k)$.

Шаг 2. Выбирается случайная точка $p_k \in [0, 1]$ согласно закону равномерного распределения. Следующее приближение выбирается по правилу:

$$x_{k+1} = \begin{cases} y_{k+1}, & \text{если } p_k \leq A(x_k, y_{k+1}, t_k) \\ x_k & \text{иначе} \end{cases}$$

Шаг 3. Пересчитывается параметр $t_{k+1} = U(t_0 k)$.

Шаг 4. Проверяется условие остановки. Если оно не выполняется, устанавливается $k = k + 1$ и происходит возврат к шагу 1.

Основной особенностью алгоритма является его способность на каждом шаге принимать решения, увеличивающие значение

целевой функции, что позволяет ему «выбираться» из локальных минимумов, в отличие, например, от алгоритма градиентного спуска [3]. Другая важная особенность алгоритма — отсутствие требований к дифференцируемости целевой функции f .

Заметим, что в большей части работ, посвящённых алгоритму имитации отжига, рассматривается применение алгоритма к решению задач дискретной оптимизации. Несмотря на это, существует обоснованный интерес к возможности применения алгоритма в задачах непрерывной оптимизации [4,5]. В следующих разделах рассмотрен один из вариантов построения такого алгоритма, а также приводятся экспериментальные данные поиска глобального минимума некоторых многомерных непрерывных функций с помощью предложенного алгоритма.

ВЫБОР ФУНКЦИИ ПРИНЯТИЯ НОВОГО ПРИБЛИЖЕНИЯ

Как и для дискретного случая, в качестве функции, осуществляющей принятие решение о том, принимать ли новое значение в качестве решения, может быть выбрана функция Метрополиса [2]:

$$A(x, y, t) = \min \left[1, \exp \left(- \frac{f(y) - f(x)}{t} \right) \right]$$

Функция Метрополиса всегда принимает положительное решение о принятии очередного y_{k+1} в качестве нового приближения в случае, если оно уменьшает значение f , т.е.

$f(y_{k+1}) \leq f(x_k)$. В то же время с определённой вероятностью может быть принято решение, увеличивающее значение целевой функции. Это свойство позволяет алгоритму

не «застрять» в локальном минимуме. Вероятность принятия положительного решения в случае увеличения значения целевой функции зависит от параметра t , представляющего «температуру» системы. Легко заметить, что в случае когда $\{t_k\} \rightarrow 0$, $A(x, y, t) \rightarrow 0$ при $k \rightarrow \infty$. Таким образом вероятность принятия нового приближения, увеличивающего значение f уменьшается с течением времени.

Другим возможным вариантом функции принятия нового приближения может являться, т.н. критерий Баркера [4]:

$$A(x, y, t) = \left[1 + \exp\left(\frac{f(y) - f(x)}{t}\right) \right]^{-1}.$$

Важным отличием критерия Баркера от функции Метрополиса является то, что первый может не принимать решения, уменьшающие значение целевой функции, особенно те, что незначительно улучшают его. Однако с уменьшением параметра t вероятность принятия лучших решений увеличивается.

Предлагаемый в работе вариант алгоритма использует функцию Метрополиса в качестве функции принятия новых приближений.

ПОИСК СЛЕДУЮЩЕГО ПРИБЛИЖЕНИЯ

В случае дискретной оптимизации, следующее приближение, как правило, получается путём различных перестановок элементов вектора текущего решения [6].

Для нахождения нового приближения в задачах непрерывной оптимизации можно воспользоваться подходом аналогичным применяемому в методах градиентного спуска. Зададимся некоторым вектором v_n , который может быть выбран относительно произвольно, тогда новое решение может быть определено по формуле [3]:

$$y_{k+1} = x_n + \theta v_n.$$

При этом θ_k — случайно сгенерированный вектор, такой что $\|\theta\| = 1$. Очевидно, что, если целевая функция f непрерывна на S ,

обязательно найдётся такой θ , что $f(y_{n+1}) < f(x_n)$ при $y_{n+1} \neq x^*$.

Вектор v_n может выбираться различными способами. В предлагаемом в статье варианте алгоритма предполагается, что вектор v_n фиксированный и не зависит от номера итерации.

Заметим, что выбор фиксированного v_n позволяет однозначно определить функцию $N(x_m)$ «соседних» решений для $x_m \in S$, которые могут быть достигнуты за одну итерацию. Таким образом, можно оценить минимальное число итераций, необходимое алгоритму для достижения x^* . Пусть x_0 — выбранное начальное приближение. Тогда минимальное число итераций, необходимое для нахождения x^* можно выразить как

$$Q_{min} = d(x^*, x_0) / v_k,$$

где d — Евклидово расстояние.

Оценим также вероятность принятия решений, увеличивающих значение f , на каждом шаге алгоритма, предполагая $v_n = const$. Поскольку $\|\theta\| = 1$, то в худшем случае $f(y_{k+1}) - f(x_k) = v_n$. Тогда для k -го шага алгоритма вероятность выбора худшего решения можно записать как

$$P_k = \exp(-v_k / t_k).$$

СХЕМА ОХЛАЖДЕНИЯ

Параметр t_k , также называемый температурой, зависит от номера итерации. Функция $U(t_0, k)$, зависящая от начальной температуры t_0 и номера итерации k , служит для пересчёта значения температуры на каждом шаге алгоритма.

Существует несколько подходов к задаче функции U . В самом простом варианте функция охлаждения зависит только от номера итерации и начального значения температуры. В некоторых источниках [4] предлагает-

АЛГОРИТМ ИМИТАЦИИ ОТЖИГА ДЛЯ ЗАДАЧ НЕПРЕРЫВНОЙ ОПТИМИЗАЦИИ

ся определять температуру в каждой точке индивидуально, т.е. $t_{k+1} = U(t_0, k, x_k)$.

В соответствии с определением функции $A(x, y, t)$, функция определения температуры должна быть монотонно убывающей. Исторически [2], первой схемой охлаждения была, т.н. схема Больцмана:

$$t_{k+1} = t_0 / \ln(1 + k).$$

Основной недостаток схемы Больцмана — медленное убывание температуры, что может приводить к медленной сходимости алгоритма.

Альтернативной схемой охлаждения является схема Коши:

$$t_{k+1} = t_0 / (1 + k).$$

На рисунке 1 приводится график изменения температуры от $t_0 = 100$ за первые 200 итераций.

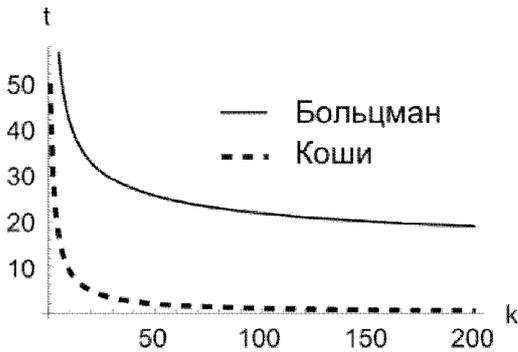


Рисунок 1 – Изменение температуры от 100 градусов при использовании схем Больцмана и Коши

В работе исследовались оба варианта схем охлаждения.

ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ

Предложенный в статье вариант алгоритма имитации отжига для задач безусловной непрерывной оптимизации был реализован на языке Wolfram Mathematica. Его свойства были протестированы на различных унимодальных и мультимодальных функциях.

Таблица 1 – Тестовые функции

Функция	S
<i>Унимодальные функции</i>	
$F_1 = \sum_{i=1}^n x_i^2$	$[-100, 100]$
$F_2 = (1 - x)^2 + 100(y - x^2)^2$	$[-30, 30]^2$
$F_3 = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$
$F_4 = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]$
<i>Мультимодальные функции</i>	
$F_5 = \sum_{i=1}^n -x_i \sin(\sqrt{x_i})$	$[-500, 500]$
$F_6 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] +$	$[-5, 12, 5, 12]$

Алгоритм тестировался с использованием схем охлаждения Коши и Больцмана. Начальное значение температуры в случае охлаждения Коши было выбрано равным 300, для охлаждения Больцмана 10. Максимальное количество итераций было установлено на 2000. Вектор v_n выбран единичным.

В таблице 2 приведены результаты эксперимента. Для сравнения указаны результаты работы другого метаэвристического алгоритма — метода роя частиц (МРЧ). Выбор МРЧ в качестве алгоритма для сравнения обусловлен двумя причинами: во-первых, МРЧ изначально подходит для задач непрерывной оптимизации; во-вторых, метод хорошо изучен, и существует большое количество работ, посвящённых выбору параметров МРЧ.

Таблица 2 – Численные результаты, $n = 30$

Тестовая функция	МРЧ	ИО (К)	ИО (Б)
1	2	3	4
F_1	$1,8 \times 10^{-3}$	1,052	$7,3 \times 10^{-1}$
F_2	$5,6 \times 10^{-3}$	0,059	0,124
F_3	$2,2 \times 10^{-3}$	$1,97 \times 10^{-6}$	$4,8 \times 10^{-7}$
F_4	$6,6 \times 10^{-3}$	$8,5 \times 10^{-1}$	$8,1 \times 10^{-1}$

Продолжение таблицы 2

1	2	3	4
F_5	$-9,8 \times 10^3$	$-6,4 \times 10^3$	$-6,1 \times 10^3$
F_6	56,6	$2,2 \times 10^2$	$4,1 \times 10^2$

Из таблицы 2 видно, что предложенная версия алгоритма имитации отжига показывает схожие результаты с МРЧ. Также для некоторых тестовых функций алгоритм имитации отжига показывает большую скорость сходимости к глобальному минимуму. Показательно, что одной из таких функций является тестовая функция Розенброка (F_2).

Также видно, что для данных тестовых функций, выбор схемы Коши или Больцмана не оказывает значительного влияния: для некоторых функций (F_2, F_6) использование схемы Коши обеспечивает лучший результат, для других (F_1, F_3, F_4, F_5) схема Больцмана улучшает решение.

На рисунке 2 приведены графики значений тестовых функций на каждом шаге алгоритма. Пунктирной линией изображены значения, полученные при использовании схемы Коши, сплошной — Больцмана.

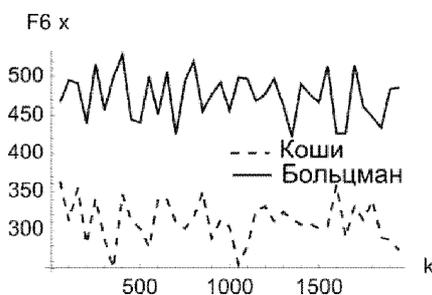
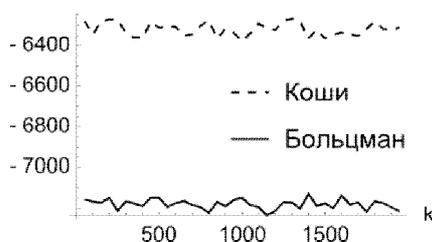
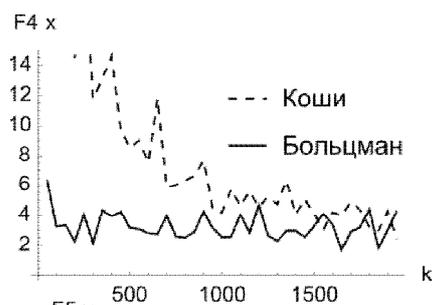
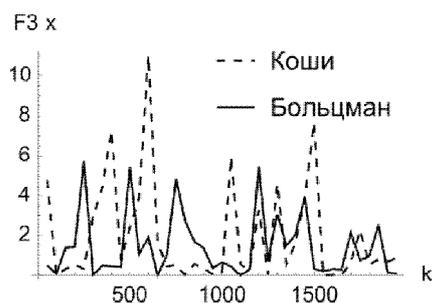
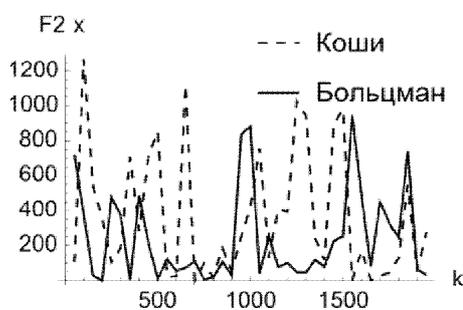
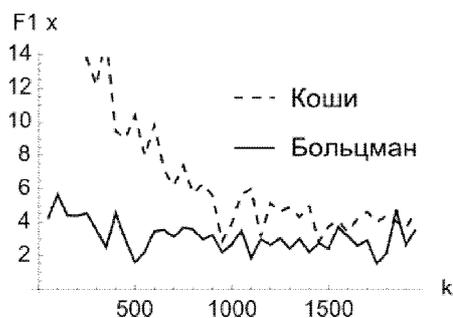


Рисунок 2 – Значения тестовых функций на каждом шаге алгоритма

ЗАКЛЮЧЕНИЕ

В статье описан вариант алгоритма имитации отжига, который может быть использован в задачах безусловной непрерывной оптимизации. Рассмотрены вопросы выбора следующего приближения, схемы охлаждения и принятия нового приближения.

В ходе численного эксперимента исследовано поведение алгоритма при поиске глобальных минимумов унимодальных и мультимодальных функций. Произведено сравнение полученных результатов с результатами метода роя частиц для тех же тестовых

функций, а также сравнение результатов при использовании различных схем охлаждения.

Дальнейшими направлениями исследования могут быть:

- использование алгоритма для случая, когда на целевую функцию $f: S \rightarrow R$ накладывается ряд ограничений $g_i(x) = 0$;

- обобщение алгоритма на случай многокритериальной оптимизации, т.е. $f: R^m \rightarrow R^n$.

Некоторые сведения о таких модификациях алгоритма могут быть найдены в работах [7,8].

СПИСОК ЛИТЕРАТУРЫ

1. Поляк, В.Т. Введение в оптимизацию / В.Т. Поляк. - М.: Наука. Главная редакция физико-математической литературы, 1983. – 384 с.
2. Kirkpatrick, S. Optimization by Simulated Annealing / S. Kirkpatrick, C.D. Gelatt Jr, M.P. Vecchi // Science. – 1983. – Vol. 220 (4598). – p. 671-680.
3. Устинов, С.М. Вычислительная математика / С.М. Устинов, В.А. Зимницкий. – СПб.: БХВ-Петербург, 2009. – 336 с.

4. Gendreau, M. Handbook of Metaheuristics (International Series in Operations Research and Management Science) / M. Gendreau, J.-Y. Potvin. – Springer, 2010. – 675 p.

5. Ларионов, В.Н. Применение современных методов оптимизации для определения параметров математических моделей электроустановок / В.Н Ларионов, Е.А. Акиншин // Ползуновский вестник. – 2002. – № 1. – с. 17-20.

6. Chen, S.-M. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques / S.-M. Chen, C.-Y. Chien // Expert Systems with Applications. – 2011. – Vol. 38. – p. 14439-14450.

7. Bertsimas, D. Simulated Annealing / D. Bertsimas, J. Tsitsiklis // Statistical Science. – Vol. 8, № 1. – p. 10-15

8. Tsallis, C. Generalized simulated annealing / C. Tsallis, D.A. Stariolo // Physica A. – 1996. – № 233. – p. 395-406.

Лисин А.В. — аспирант, Омский государственный технический университет, каф. «Комплексная защита информации», E-mail: andrey.lisin@gmail.com.