

УДК 007

Алгоритм поиска кукушки для решения задачи глобальной безусловной оптимизации

09, сентябрь 2012

Бенза Н.Н.

*Научный руководитель: д.т.н., профессор, Карпенко А.П.
МГТУ им. Н.Э.Баумана, Москва, Россия*

МГТУ им. Н.Э. Баумана
bauman@bmstu.ru

Введение

В последнее время появляется все большее число метаэвристических алгоритмов оптимизации, вдохновленных поведением живых существ в природе. Например, метод роя частиц авторов вдохновило поведение стай птиц и косяков рыб, алгоритм светлячков был инспирирован характером свечения тропических светлячков и так далее.

Предметом исследования является новый алгоритм глобальной оптимизации – алгоритм поиска кукушки, который разработали Xin-She Yang и Susan Deb в 2009 году [1]. На создание алгоритма авторов вдохновило поведение кукушек в процессе вынужденного гнездового паразитизма, когда некоторые виды кукушек подкладывают яйца в гнезда птиц других видов [2-4].

1 Постановка задачи и схема алгоритма

Рассматриваем задачу глобальной безусловной минимизации вида

$$\min_{x \in R^d} F(x) = F(x^*) = F^*, \quad (1)$$

x - d -мерный вектор варьируемых параметров, $F(x)$ - целевая функция.

В алгоритме поиска кукушки используются следующие основные предположения. Каждое яйцо в гнезде представляет собой решение, а яйцо кукушки - новое решение. Цель заключается в использовании новых и потенциально лучших решений (кукушкиных), чтобы заменить менее хорошие решения в гнездах. В простейшей форме алгоритма в каждом гнезде находится по одному яйцу. Алгоритм может быть расширен для более сложных случаев, когда в каждом из гнезд находится более одного яйца, представляющих некоторое множество решений.

Поиск кукушки основан на трех следующих правилах.

1) Каждая кукушка откладывает одно яйцо за один раз, и подкладывает его в гнездо, которое выбирается случайным образом.

2) Лучшие гнезда с яйцами высокого качества (пригодными решениями) переходят на следующее поколение.

3) Число доступных гнезд фиксировано, а яйцо кукушки может быть обнаружено хозяином гнезда с вероятностью $p_a \in (0,1)$. Обнаруженные решения исключаются из дальнейшего рассмотрения.

Основные этапы поиска кукушки можно представить в виде псевдокода следующим образом.

```

begin
  Генерация начальной популяции  $n$  гнезд  $x_j$ , ( $j=1,2,\dots,n$ );
  while (критерий останова)
    Случайным образом поместить кукушку в точку  $x_i$ ,
    выполняя полеты Леви;
    Случайным образом выбрать гнездо  $j$  среди  $n$  гнезд;
    if ( $F_i < F_j$ )
      Заменить  $x_j$ , на новое решение  $x_i$ ;
    end if
    Часть гнезд, обнаруженных с вероятностью  $p_a$ , удалить
    из популяции и построить такое же число новых;
    Сохранить лучшее решение (гнездо);
  end while
  Пост-обработка результатов и визуализация;
end

```

2 Полеты Леви

При создании новых решений $x(t+1) = x^{t+1}$ для кукушки i полеты Леви осуществляются по формуле

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Lévy(\lambda), \quad (2)$$

где $\alpha > 0$ - размер шага, который должен быть связан с масштабами задачи поиска. В большинстве случаев, можно использовать $\alpha=1$. Уравнение (2) представляет собой стохастическое уравнение случайных блужданий. В общем, случайное блуждание представляет собой цепь Маркова, для которой следующее положение зависит только от текущего местоположения (первое слагаемое в уравнении) и вероятности перехода (второе слагаемое). Случайную длину шага в уравнении (2) определяет распределение Леви

$$Lévy(\lambda) = t^{-\lambda}, \quad (1 < \lambda \leq 3). \quad (3)$$

Некоторые из новых решений должны быть порождены полетами Леви вокруг текущего лучшего решения, что ускорит сходимость алгоритма. Значительная часть новых решений должна быть сформирована в случайно выбранных точках пространства поиска, т.е. достаточно далеко от текущих лучших решений. Это позволит диверсифицировать поиск, повысить вероятность преждевременной сходимости.

3 Исследование эффективности алгоритма

В рамках исследований проводился анализ эффективности алгоритма поиска кукушки при изменении числа гнезд n и вероятности обнаружения гнезда p_a . В докладе представлены результаты численных экспериментов при поиске минимума функции Растригина, которая имеет вид

$$f(x) = 10d + \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i)). \quad (4)$$

Результаты исследования иллюстрируют рисунки 1 – 3 и таблицы 1, 2, на которых принято, что t_{end} - начало стагнации вычислительного процесса, n_f - число испытаний (вычислений значений целевой функции).

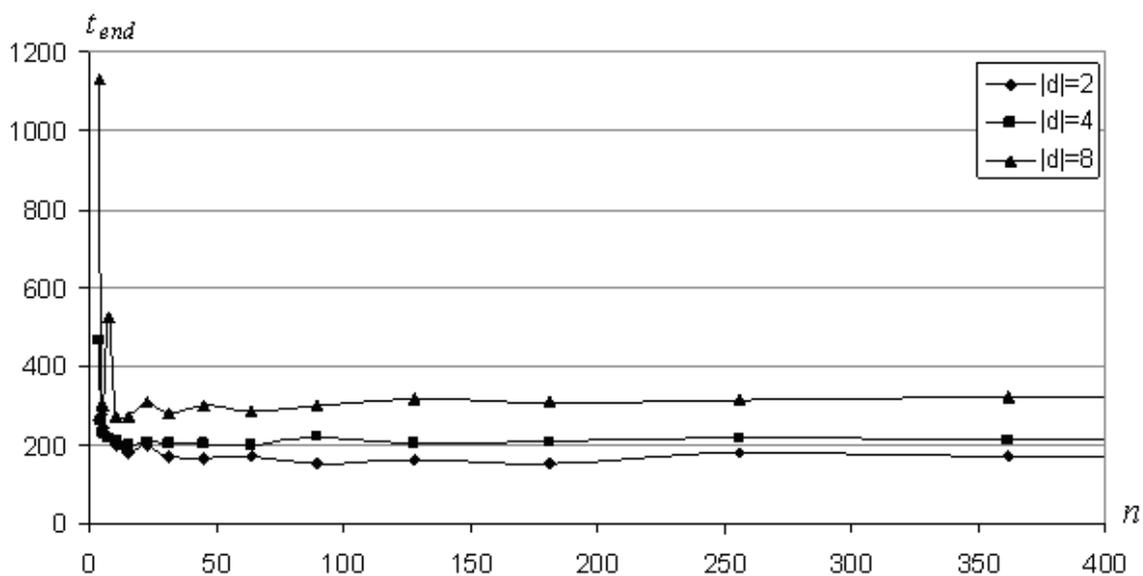


Рис. 1 Число итераций t_{end} в зависимости от числа гнезд n :
 функция Растригина; $p_a = 0,25$

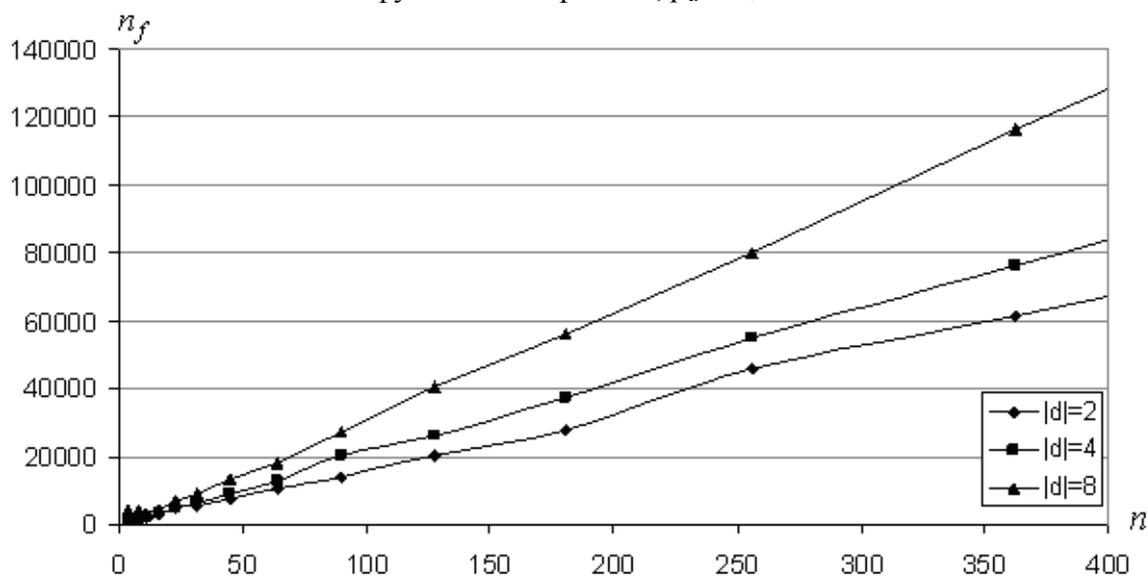


Рис. 2 Число испытаний n_f в зависимости от числа гнезд n :
 функция Растригина; $p_a = 0,25$

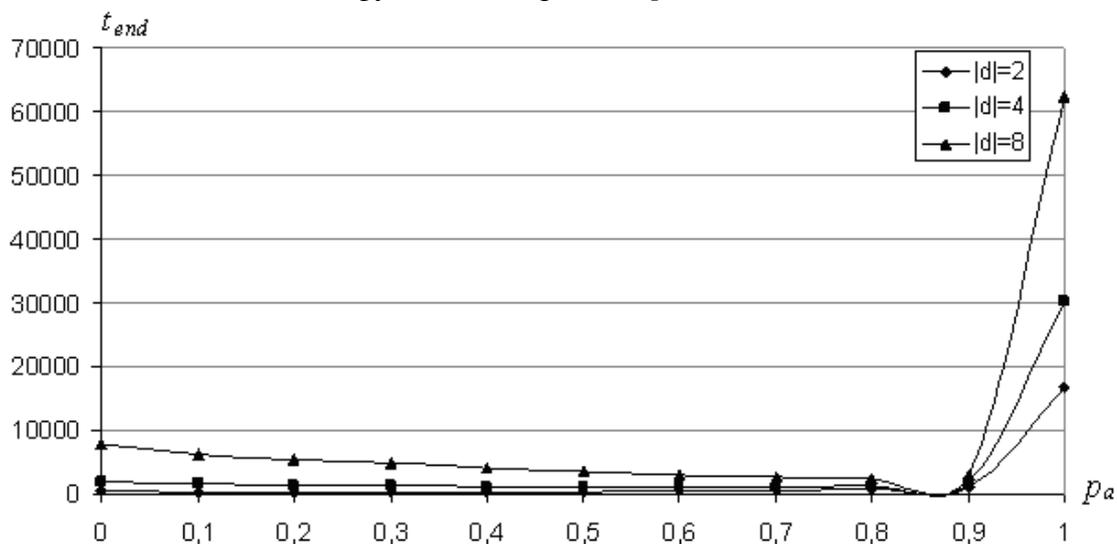


Рис. 3 Число итераций t_{end} в зависимости от вероятности p_a :

функция Растригина; $n = 32$

Таблица 1 – Вероятность локализации минимума функции Растригина при различных значениях параметров n, d

n	$d=2$	$d=4$	$d=8$
4	80	17	0
5	94	49	0
6	97	73	7
8	100	96	41
11	100	100	83
16	100	100	98
23	100	100	100
32	100	100	100
45	100	100	100
64	100	100	100
90	100	100	100
128	100	100	100
181	100	100	100
256	100	100	100

Таблица 2 – Вероятность локализации минимума функции Растригина при различных значениях параметров $p_a, d; n=32$

p_a	$d=2$	$d=4$	$d=8$
0	100	100	91
0,1	100	100	100
0,2	100	100	100
0,3	100	100	100
0,4	100	100	100
0,5	100	100	100
0,6	100	100	100
0,7	100	100	100
0,8	100	100	100
0,9	100	100	100
1	95	87	100

Рисунки показывают, что при увеличении числа гнезд n скорость сходимости алгоритма повышается и одновременно возрастает число вычислений значений целевой функции. При увеличении размерности пространства d заметно возрастает суммарное число испытаний.

На основе результатов, представленных в таблице 1, можно утверждать, что при $d=2$ вероятность нахождения минимума функции Растригина будет 100% при числе гнезд, большем 8; при $d=4$ следует выбирать значения n , большие 10; при $d=8$ значение n должно превосходить 20.

При исследовании влияния параметра p_a использовалось фиксированное значение числа гнезд $n=32$. Из рисунка 3 следует, что сходимость алгоритма не чувствительна к значениям этого параметра, если $p_a \in [0; 0,9]$. На основании данных таблицы 2 можно заключить, что значения величины p_a следует выбирать из интервала $[0,1; 0,9]$, поскольку такие значения обеспечивают 100% вероятность обнаружения минимума целевой функции.

Заключение

В работе рассматривается новый метаэвристический алгоритм глобальной оптимизации – алгоритм поиска кукушки, использующий полеты Леви. На примере

сложной многоэкстремальной функции Растригина выполнено исследование эффективности алгоритма в зависимости от его основных параметров. На основании результатов исследования можно сделать вывод, что алгоритм поиска кукушки обладает высоким потенциалом развития.

Литература

1. Yang X.-S., S. Deb S. Cuckoo search via L'evy flights / In: Proc. Of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), 2009, India, pp. 210-214.
2. Tuba M., Subotic M., Stanarevic N. Modified cuckoo search algorithm for unconstrained optimization problems / In: Proceedings of the European Computing Conference, 2010, pp. 263-268.
3. Valian Eh., Mohanna Sh., Tavakoli S. Improved cuckoo search algorithm for feedforward neural network training / In: International Journal of Artificial Intelligence & Applications (IJAI), 2011, Vol.2, No.3, pp. 36-43.
4. http://en.wikipedia.org/wiki/Cuckoo_search