УДК 519.8

Анализ эффективности биоинспирированных методов глобальной оптимизации

Орловская Н.М.

Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, А-80, ГСП-3,125993, Россия

e-mail: orlovskaya.nataly@yandex.ru

Аннотация

Рассмотрены биоинспирированные методы глобальной оптимизации: методы, имитирующие поведение лягушек, кукушек, светлячков, и метод, имитирующий распространение сорняков. Основная особенность всех четырех методов состоит в возможности поиска глобального экстремума многоэкстремальных целевых функций с большим числом переменных. Целью работы является анализ эффективности методов, который был проведен путем применения их для поиска глобального условного максимума нескольких типовых функций. Для каждого метода разработан пошаговый алгоритм решения поставленной задачи оптимизации, а также создано программное обеспечение.

Ключевые слова: оптимизация, глобальный экстремум, популяция, приспособленность, целевая функция, биоинспирированные методы.

Введение

Важным этапом процесса проектирования авиационно-космических комплексов является осуществление расчетов по оптимизации технических и экономических характеристик. В результате такого анализа вырабатываются требования, определяющие выбор наилучшего сочетания характерных параметров. Этот этап имеет особое значение, так как позволяет увеличивать эффективность эксплуатации летательных аппаратов и их качество при минимальных затратах и времени конструирования, что обуславливает важность выбора эффективного метода оптимизации.

В статье рассматривается задача поиска глобального экстремума функций многих переменных с помощью четырех биоинспирированных методов глобальной оптимизации: метода, имитирующего поведение лягушек [1-3], кукушек [4-9], светлячков [10] и метода, имитирующего распространение сорняков [11].

Для каждого из методов сформированы пошаговые алгоритмы решения задачи, а также создано программное обеспечение, позволяющее исследовать эффективность работы алгоритмов на тестовых примерах. Анализ работы методов был проведен путем применения их для нахождения глобального максимума типовых многоэкстремальных функций двух переменных, для которых известно аналитическое решение.

Биоинспирированные методы глобальной оптимизации [12] – группа метаэвристических методов [13,14], имитирующих процессы в природной среде и поведение некоторых видов животных и растений. Эти методы становятся все более попу-

лярными, так как позволяют находить решение таких задач оптимизации, для которых нахождение решений с помощью традиционных численных методов поиска экстремума оказывается неэффективным или вообще невозможным. Они не гарантируют сходимости к глобальному решению задачи оптимизации, однако позволяют получить хорошее решение за приемлемое с практической точки зрения время.

Метод, имитирующий поведение кукушек (Cuckoo Search) [4-9], и метод, имитирующий распространение сорняков (Weed Colonization) [11], относятся к эволюционным методам. В их основе лежит идея формирования новой популяции, когда в результате применения некоторых операций удаляются особи (возможные решения) с наихудшей приспособленностью и заменяются лучшими.

Методы, имитирующие поведение лягушек (Shuffled Frog-Leaping Algorithm) [1-3] и светлячков (Glowworm Swarm Optimization) [10], относятся к методам «роевого» интеллекта [13,14]. Основная идея этих методов – это взаимодействие множества агентов системы «роевого» интеллекта между собой, которые обмениваются информацией с целью приближения к оптимальному решению.

Все четыре метода объединяет идея имитации биологических процессов и поведения животных и растений. Как методы, относящиеся к классу метаэвристических, они способны искать оптимальное решение, не «застревая» в окрестности локальных экстремумов, позволяя исследовать большее пространство поиска.

Постановка задачи

Дана целевая функция $f(x) = f\left(x_1, x_2, \ldots, x_n\right)$, определенная на множестве допустимых решений $D \subseteq R^n$. Требуется найти глобальный условный максимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что

$$f\left(x^*\right) = \max_{x \in D} f(x), \tag{1}$$

где $x = \left(x_1, x_2, \ldots, x_n\right)^T$, $D = \left\{x \mid x_i \in \left[a_i, b_i\right], i = 1, 2, \ldots, n\right\}$.

Метод, имитирующий поведение лягушек

Стратегия поиска решения. Алгоритм, имитирующий поведение лягушек (shuffled frog leaping algorithm) [1-3], основан на учете поведения прыгающих лягушек при поиске пищи. В методе реализуется итерационный переход от одной *популяции* к другой. Популяция представляется множеством из P лягушек (решений), разделенным на M непересекающихся подмножеств, содержащих одинаковое число лягушек, называемых *стаями*.

Внутри каждой стаи производится *локальный поиск*. При этом внутри стаи находится наилучшее решение $\left(x_{best}\right)$ и наихудшее $\left(x_{worst}\right)$. Затем положение *наихудшей лягушки* (с наихудшим значением целевой функции) изменяется по правилу:

$$x_{worst}^{new} = x_{worst} + C \cdot rand \square \left[x_{best} - x_{worst}\right], \tag{2}$$

где $rand$ – случайный вектор, элементы которого равномерно распределены на отрезке $\left[0;1\right]$; $\square$ – поэлементное произведение векторов по Адамару; C – коэффици-

ент. Если полученное решение x_{worst}^{new} лучше, чем наихудшее x_{worst}, то оно заменяет его. Если – нет, то находится новое положение наихудшей лягушки по формуле (2), где вместо наилучшего решения x_{best} внутри стаи используется наилучшее среди всех лягушек в популяции решение x_{gb}. Если опять полученное решение не лучше наихудшего, то наихудшее решение заменяется полученным случайным образом с помощью равномерного распределения на множестве допустимых решений. Внутри каждой стаи производится одно и то же количество итераций локального поиска. Локальный поиск аналогичен применяемому в методе частиц в стае (стремление к наилучшему варианту из достигнутых в стае или в популяции в целом).

После завершения локального поиска *деление популяции* на стаи производится заново. При этом все решения упорядочиваются по убыванию значения целевой функции. Порядок формирования стай: наилучшее решение помещается в первую стаю, следующее – во вторую и т.д. M-е в M-ю стаю, $(M+1)$-е снова в первую стаю, $(M+2)$-е во вторую стаю и т.д. Описанный процесс соответствует обмену информацией между стаями с целью эффективного приближения к глобальному экстремуму. Процедура поиска завершается при достижении максимального числа итераций.

Алгоритм поиска решения.

Шаг 1. Задать: размер популяции P; число стай (memeplexes) M; параметр C; максимальное число итераций локального поиска в каждой стае IT; максимальное число глобальных итераций $ITER$.

Положить $j = 1$ (счетчик числа глобальных итераций).

Шаг 2. Генерировать начальную популяцию лягушек (решений) на множестве D с помощью равномерного распределения. Для каждой лягушки подсчитать значение целевой функции.

Шаг 3. Упорядочить элементы популяции по убыванию значений целевой функции.

Шаг 4. Сформировать из популяции M стай (групп) лягушек: наилучшее (с наибольшим значением целевой функции) решение поместить в первую стаю, следующее – во вторую и т.д., M -е поместить в M -ю стаю, $(M+1)$-е снова в первую стаю и т.д. Результатом являются M стай, содержащих по m лягушек каждая, так что $P = M \cdot m$.

Шаг 5. В рамках каждой стаи провести заданное число IT итераций локального поиска (прыжков лягушки):

Шаг 5.1. Положить $Num = 1$ (номер стаи).

Шаг 5.2. Положить $it = 1$ (счетчик числа итераций локального поиска).

Шаг 5.3. В рамках стаи с номером Num найти:

– наилучшее решение x_{best} (положение лягушки) в стае, которому соответствует наибольшее значение целевой функции;

– наихудшее решение x_{worst}, которому соответствует наименьшее значение целевой функции.

Найти наилучшее глобальное решение в популяции x_{gb}.

Шаг 5.4. Найти новое положение наихудшей лягушки:

$$x_{worst}^{new} = x_{worst} + C \cdot rand \square [x_{best} - x_{worst}]. \quad (3)$$

Шаг 5.5. Если $f\left(x_{worst}^{new}\right) > f\left(x_{worst}\right)$, то заменить решение x_{worst} на x_{worst}^{new} и перейти к шагу 5.7. Иначе снова найти новое положение наихудшей лягушки:

$$x_{worst}^{new} = x_{worst} + C \cdot rand \square [x_{gb} - x_{worst}]. \quad (4)$$

Шаг 5.6. Если $f\left(x_{worst}^{new}\right) > f\left(x_{worst}\right)$, то заменить решение x_{worst} на x_{worst}^{new} и перейти к п. 5.7. Иначе генерировать решение x_{worst}^{new} случайно на множестве D с помощью равномерного распределения, заменить им решение x_{worst} и перейти к шагу 5.7.

Шаг 5.7. Если $it = IT$, то локальный поиск в стае завершить. Сравнить номер Num с числом M:

– если $Num = M$, то завершить локальный поиск во всех стаях и перейти к шагу 6;

– если $Num < M$, то положить $Num = Num + 1$ и перейти к шагу 5.2.

Если $it < IT$, то положить $it = it + 1$ и перейти к шагу 5.3.

Шаг 6. Отменить деление популяции на стаи, считая всех лягушек элементами популяции с номером j.

Шаг 7. Если $j = ITER$, процесс завершить, в качестве ответа принять решение x_{gb}. Если $j < ITER$, положить $j = j + 1$ и перейти к шагу 3.

Метод, имитирующий поведение светлячков

Стратегия поиска решения. Светлячки порождают свет с интенсивностью I, обратно пропорциональной квадрату расстояния [10]. Это обстоятельство свиде-

тельствует о том, что светлячок виден на ограниченной дистанции. Свет помогает ему притягивать пищу и партнеров. Вводятся допущения:

а) привлекательность светлячка пропорциональна яркости, поэтому в произвольной паре светлячков менее яркий будет стремиться к более яркому; привлекательность (или яркость) уменьшается с увеличением расстояния. Если нет более яркого светлячка, то он будет двигаться случайным образом;

б) яркость отождествляется с величиной целевой функции.

Привлекательность определяется двумя факторами: величиной целевой функции и расстоянием до светлячка. Яркость в зависимости от расстояния до источника определяется согласно закону

$$I(r) = \frac{I_0}{r^2}, \tag{5}$$

где I_0 - интенсивность источника, r - расстояние до него. Чтобы исключить деление на нуль при $r = 0$, используются две зависимости, приближенно совпадающие (с погрешностью $O(r^3)$) с приведенной при малых r:

$$I(r) = \frac{I_0}{1 + \gamma r^2}, \tag{6}$$

$$I(r) = I_0 e^{-\gamma r^2}, \tag{7}$$

где γ - коэффициент.

Движение светлячка с номером j к светлячку с номером m (более яркому) определяется формулой

$$x^j = x^m + I_0 e^{-\gamma r_{jm}^2}(x^m - x^j) + \alpha \cdot S \square \left(rand - \frac{1}{2} \right), \tag{8}$$

где $rand \sim R[0,1]$ – случайный вектор, все элементы которого равномерно распределены на отрезке $[0;1]$; $\square$ – поэлементное произведение векторов по Адамару, $I_0 = 1,\ \alpha \in [0,1],\ \gamma > 0,\ r_{jm} = \sqrt{\sum_{i=1}^{n} \left(r_i^j - r_i^m \right)^2}$ - расстояние между светлячками; S - вектор, учитывающий шкалу изменения каждой координаты x_i.

Алгоритм поиска решения.

Шаг 1. Задать параметры метода: α – коэффициент рандомизации; γ – коэффициент ослабления интенсивности; NP – число светлячков; N – максимальное число итераций; S – вектор масштаба по всем переменным.

Шаг 2. Генерировать NP светлячков на множестве D с помощью равномерного распределения: $x^{1,0},\ldots,x^{NP,0}$. Подсчитать значения целевой функции $f(x^{1,0}),\ldots,f(x^{NP,0})$. Положить $k = 0$ (счетчик числа итераций).

Шаг 3. Положить $j = 1$.

Шаг 4. Положить $m = 1$.

Шаг 5. Если $f\left(x^{m,k}\right) > f\left(x^{j,k}\right)$, то передвинуть светлячка с номером j к светлячку с номером m:

$$\hat{x}^{j,k} = x^{j,k} + e^{-\gamma r_{jm}^2}\left(x^{m,k} - x^{j,k}\right) + \alpha \cdot S \square \left(rand - \frac{1}{2}\right), \qquad x^{j,k} = \hat{x}^{j,k}. \qquad (9)$$

Вычислить $f\left(x^{j,k}\right)$. Если $m = NP$, перейти к шагу 7. Если $m < NP$, перейти к шагу 6.

Шаг 6. Положить $m = m + 1$ и перейти к шагу 5.

Шаг 7. Если $j = NP$, перейти к шагу 8. Если $j < NP$, положить $j = j + 1$ и перейти к шагу 4.

Шаг 8. Положить $x^{j,k+1} = x^{j,k}, \quad j = 1,\ldots, NP$. Среди всех решений выбрать наилучшее: x^b.

Шаг 9. Если $k = N - 1$, процесс завершить и в качестве решения выбрать x^b.

Если $k < N - 1$, положить $k = k + 1$ и перейти к шагу 3.

Метод, имитирующий поведение кукушек

Стратегия поиска решения. Кукушки относятся к особенным птицам не только в силу специфичного пения, но и агрессивной стратегии размножения. Они могут откладывать яйца в чужие гнезда. Если хозяйка гнезда обнаруживает чужие яйца, она может выбросить их из гнезда или покинуть гнездо и свить его в другом месте. Некоторые виды кукушек способны менять цвет яиц и их форму так, чтобы они были похожи на яйца, отложенные хозяйкой, что увеличивает вероятность их сохранности. Алгоритм, имитирующий поведение кукушек (cuckoo search algorithm) [4-9], использует наблюдения за поведением кукушки в процессе поиска гнезд других птиц, в которые она стремится отложить свои яйца вместо уже имеющихся.

Принимаются следующие допущения:

а) каждая кукушка откладывает только одно яйцо в случайно выбранное гнездо (при генерации нового поколения);

б) наилучшие гнезда с яйцами хорошего качества (решения с наилучшими значениями целевой функции) сохраняются при переходе к следующему поколению;

в) число доступных кукушке гнезд фиксировано. Яйцо, отложенное кукушкой, исследуется хозяйкой с вероятностью p_d.

Для простоты реализации последнего допущения часть гнезд, равная p_d, из общего числа N гнезд заменяется новыми гнездами (генерируемыми случайным образом). Каждое яйцо в гнезде представляет решение, а яйцо кукушки – новое решение. Целью процесса поиска является замещение не очень хороших решений лучшими. Новое решение (положение кукушки) генерируется с помощью распределения Леви:

$$x_i^{(j+1)} = x_i^{(j)} + \alpha \cdot Levy(\lambda), \quad i = 1,\ldots,n, \tag{10}$$

где $x_i^{(j)}$ – координата положения j-й кукушки, α – величина шага, $Levy(\lambda)$ – случайная величина, генерируемая с помощью распределения Леви ($u = t^{-\lambda}$):

$$p(x_i) = x_i^{-\lambda}, \quad \lambda \in (1,3]. \tag{11}$$

Заметим, что согласно (10) возможна достаточно большая величина приращения по каждой координате вектора x. Ее следует контролировать, проверяя принадлежность нового решения множеству допустимых решений: $x_i \in [a_i, b_i]$.

Алгоритм поиска решения.

Шаг 1. Задать параметры метода: λ – параметр распределения Леви; p_d – доля удаляемых гнезд; α – параметр изменения шага; N – число гнезд; $ITER$ – максимальное число итераций.

Положить $j = 0$ (счетчик числа глобальных итераций).

Шаг 2. Генерировать на множестве начальную популяцию из N гнезд с помощью равномерного распределения.

Шаг 3. Генерировать положение кукушки:

а) найти положение $x^{(j)}$ наилучшего гнезда с наибольшим значением целевой функции из текущей популяции;

б) найти положение кукушки:

$$x_i^{(j+1)} = x_i^{(j)} + \frac{\alpha}{j+1} \cdot Levy(\lambda), \quad i = 1,\ldots,n; \tag{12}$$

если не выполнено условие $x_i^{(j+1)} \in [a_i, b_i]$, процесс повторить;

в) подсчитать значение целевой функции $f(x^{(j+1)})$.

Шаг 4. Выбрать среди N гнезд одно случайным образом (с номером k), которому соответствует значение целевой функции f_k.

Шаг 5. Если $f(x^{(j+1)}) > f_k$, то заменить гнездо с номером k на новое гнездо $x^{(j+1)}$. Перейти к шагу 6.

Если $f(x^{(j+1)}) \le f_k$, то замену не производить и перейти к шагу 6.

Шаг 6. Упорядочить гнезда по убыванию значений целевой функции и удалить из них $p_d \cdot N$ гнезд.

Шаг 7. Генерировать на множестве D допустимых решений $p_d \cdot N$ новых гнезд (решений) с помощью равномерного распределения. При этом общее число гнезд сохраняется равным N (текущая популяция).

Шаг 8. Если $j = ITER - 1$, процесс завершить. В качестве приближенного решения из текущей популяции выбрать решение, соответствующее наибольшему значению целевой функции. Если $j < ITER - 1$, положить $j = j + 1$ и перейти к шагу 3.

Метод, имитирующий распространение сорняков

Стратегия поиска решения. Алгоритм, имитирующий распространение сорняков, использует наблюдения за процессом распространения сорняков на новых площадях (weed colonization) [11]. Сорняки относятся к растениям, чьи способности к росту и распространению наносят существенный вред культивируемым растениям. Они размножаются семенами. Предполагается, что в начале семена рассеиваются на множестве допустимых решений D, образуя начальную популяцию. Каждое семечко (решение) превращается в растение, которое затем вырабатывает количество семян, пропорциональное достигнутому состоянию (величине целевой функции). Выработанные семена случайным образом рассеиваются на исследуемом множестве D и вырастают в новые растения. Растения с плохой приспособленностью, которым соответствуют наихудшие значения целевой функции, удаляются (погибают в процессе соревновательного отбора).

Количество семян, рассеиваемое растением, определяется линейной зависимостью и находится в диапазоне $\left[s_{\min,} s_{\max} \right]$. Чем больше значение целевой функции, тем больше число рассеиваемых семян. Эти семена рассеиваются вокруг растения согласно нормальному распределению (положение растения задает математическое ожидание). Среднеквадратическое отклонение определяется числом k выполненных итераций, начальным $\sigma_{initial}$ и конечным σ_{final} значениями. Считается, что семена по-

рождают дочерние растения. Вместе с родительскими растениями они образуют новую популяцию. Если ее размер меньше допустимой величины $NPMAX$, то процесс рассеивания продолжается. Если ее размер больше $NPMAX$, то элементы популяции ранжируются по величине целевой функции. Из них $NPMAX$ лучших растений образуют новую популяцию, а остальные отбрасываются (погибают в процессе естественного отбора). Процесс поиска заканчивается по достижении максимального числа итераций $ITER$.

Алгоритм поиска решения.

Шаг 1. Задать параметры метода: NP – число семян в начальной популяции; $ITER$ – максимальное число итераций; $NPMAX$ – максимальное число элементов в популяции ($NP < NPMAX$); $s_{\max}$ – максимальное число рассеиваемых семян от одного растения; $s_{\min}$ – минимальное число рассеиваемых семян от одного растения; $\sigma_{initial}$ – начальное среднеквадратическое отклонение; σ_{final} – конечное среднеквадратическое отклонение; p – параметр.

Положить $k = 0$ (счетчик числа итераций).

Шаг 2. Генерировать начальную популяцию $\left\{x^{j,0}\right\}_{j=1}^{NP}$ из NP семян (решений) на множестве D, используя равномерное распределение: $x^{1,0},\ldots,x^{NP,0}$.

Подсчитать значения целевой функции $f(x^{1,0}),\ldots,f(x^{NP,0})$. Среди них найти максимальное $f_{\max}$ (наилучшее) и минимальное $f_{\min}$ (наихудшее).

Шаг 3. Для каждого растения, полученного из NP семян, найти число рассеиваемых семян:

$$s^j = s_{\min} + \frac{s_{\max} - s_{\min}}{f_{\max} - f_{\min}} \cdot \left(f^j - f_{\min} \right),\ j = 1,\ldots,NP, \tag{13}$$

где f^j – значение целевой функции для j-го растения.

Шаг 4. Для каждого j-го растения из популяции получить s^j семян, используя нормальное распределение с математическим ожиданием, определяемым положением растения, и среднеквадратическим отклонением

$$\sigma_{iter} = \left(\frac{ITER - k}{ITER} \right)^p \cdot \left(\sigma_{initial} - \sigma_{final} \right) + \sigma_{final}. \tag{14}$$

Вырабатываемые семена должны принадлежать множеству D (если семечко не принадлежит D, процесс генерации продолжается).

Шаг 5. Подсчитать: общее число N растений (родителей) и распределяемых ими семян; значения целевой функции для полученных семян (решений).

Если $N \le NPMAX$, положить $NP = N$, ранжировать все решения по убыванию целевой функции (найти $f^j,\quad j = 1,\ldots,NP;\ f_{\max},\ f_{\min}$), перейти к шагу 6.

Если $N > NPMAX$, ранжировать все решения по убыванию целевой функции, оставить только $NPMAX$ лучших решений, отбросив все остальные. Положить $NP = NPMAX$, найти $f^j,\quad j = 1,\ldots,NP;\ f_{\max},\ f_{\min}$, перейти к шагу 6.

Шаг 6. Если $k = ITER - 1$, процесс завершить. В качестве решения задачи поиска условного глобального максимума выбрать решение, соответствующее значению $f_{\max}$. Если $k < ITER - 1$, положить $k = k + 1$ и перейти к шагу 3.

Программное обеспечение

В среде разработки Microsoft Visual Studio 2010 на языке программирования C# создана программа поиска глобального максимума функций с помощью биоинспирированных методов глобальной оптимизации.

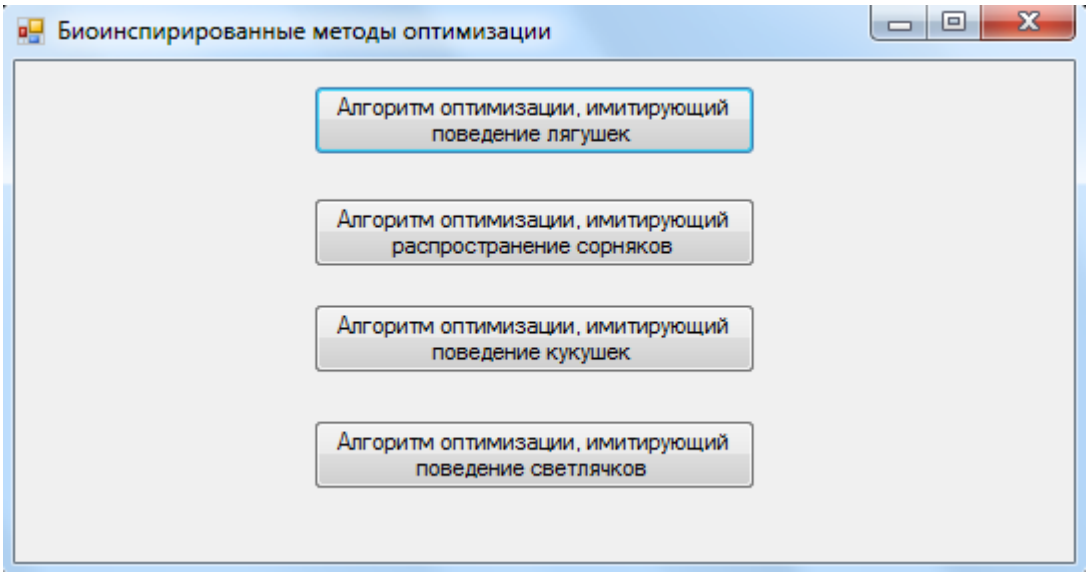

Рис. 1. Главная форма программы

После выбора метода оптимизации в главной форме программы пользователь попадает на основную форму метода, где может выбрать вид целевой функции, задать множество допустимых значений и параметры алгоритма. При нажатии кнопки «получить ответ» происходит поиск точки глобального максимума с помощью соответствующего метода оптимизации. Результат работы метода можно увидеть как на основной форме, так и в протоколе работы метода и в форме «Дополнительная информация», доступных при нажатии соответствующих кнопок, что позволяет исследовать работу метода более детально. Кроме того, на графике в поле «Изображение популяции и линии уровня функции» можно посмотреть графическую интерпретацию результата работы метода.

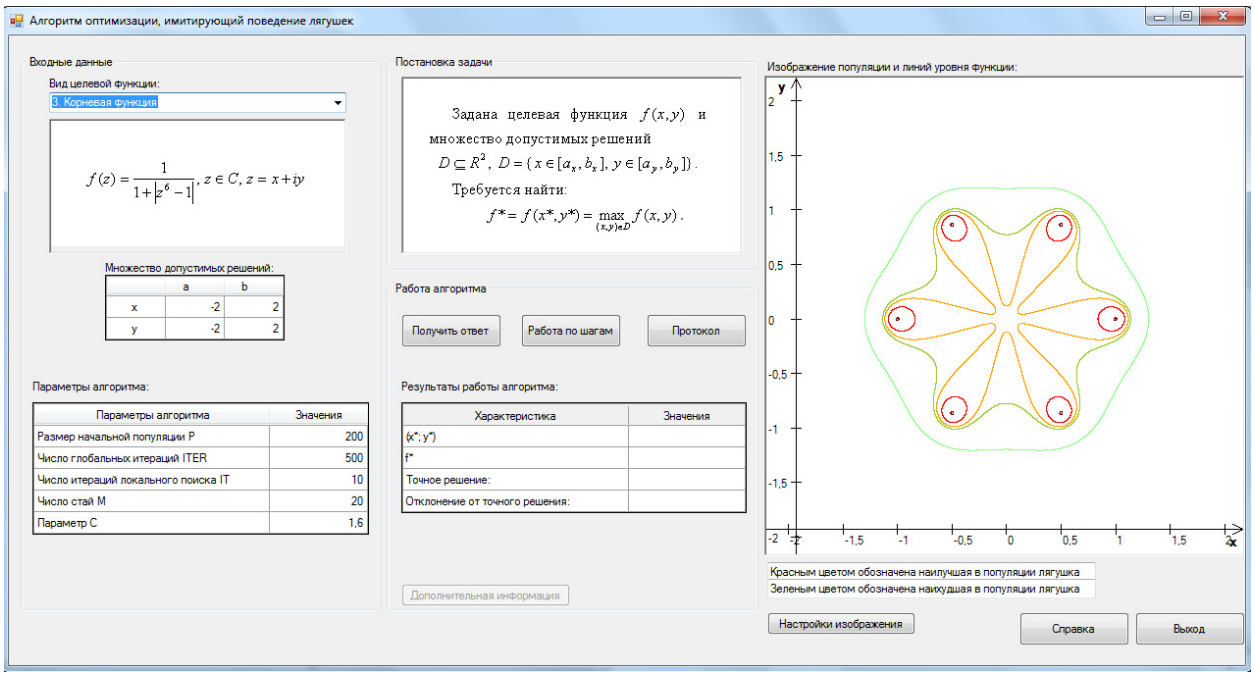

Рис. 2. Основная форма метода

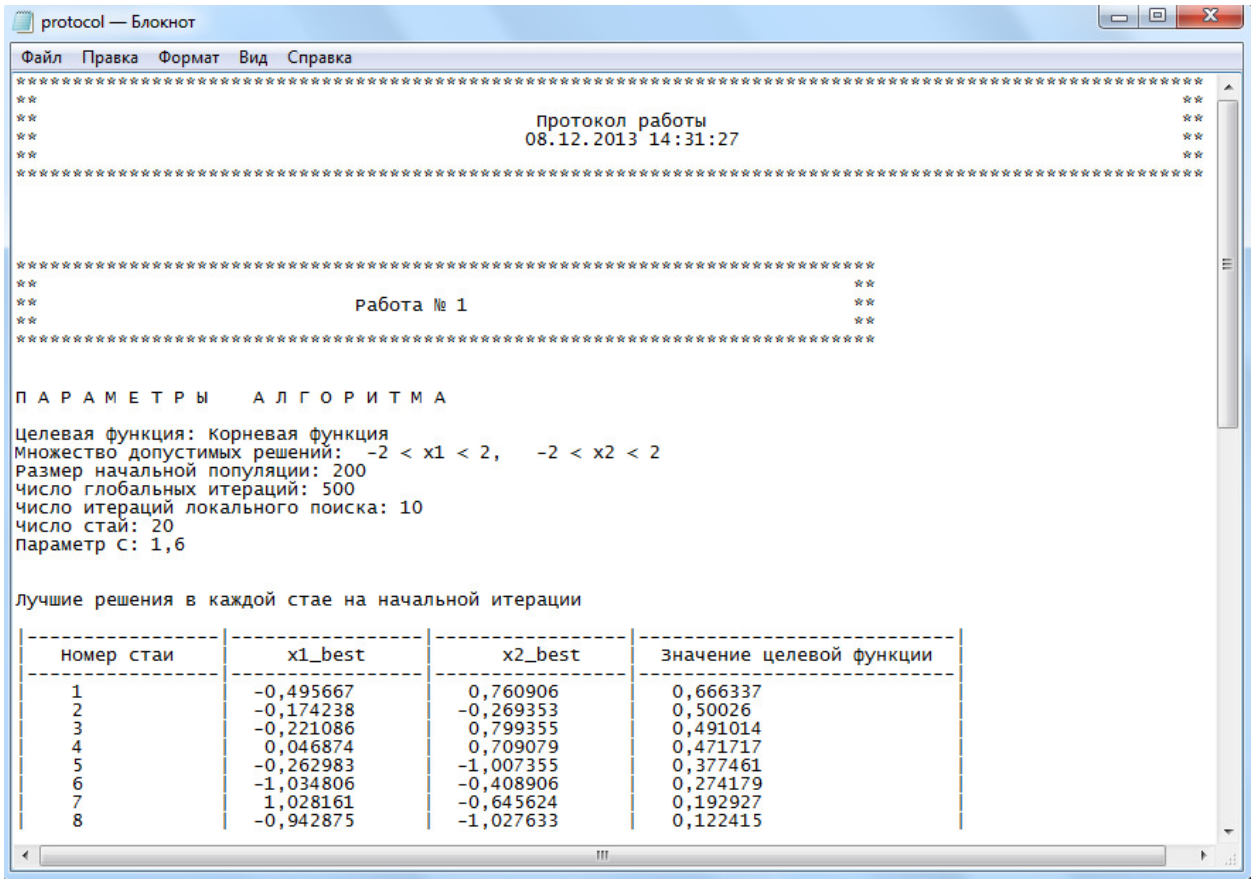

Рис. 3. Фрагмент протокола работы метода

Разработанная программа предусматривает также возможность анализа работы методов по шагам. При нажатии на основной форме на кнопку «Работа по шагам» пользователь попадает на форму вида:

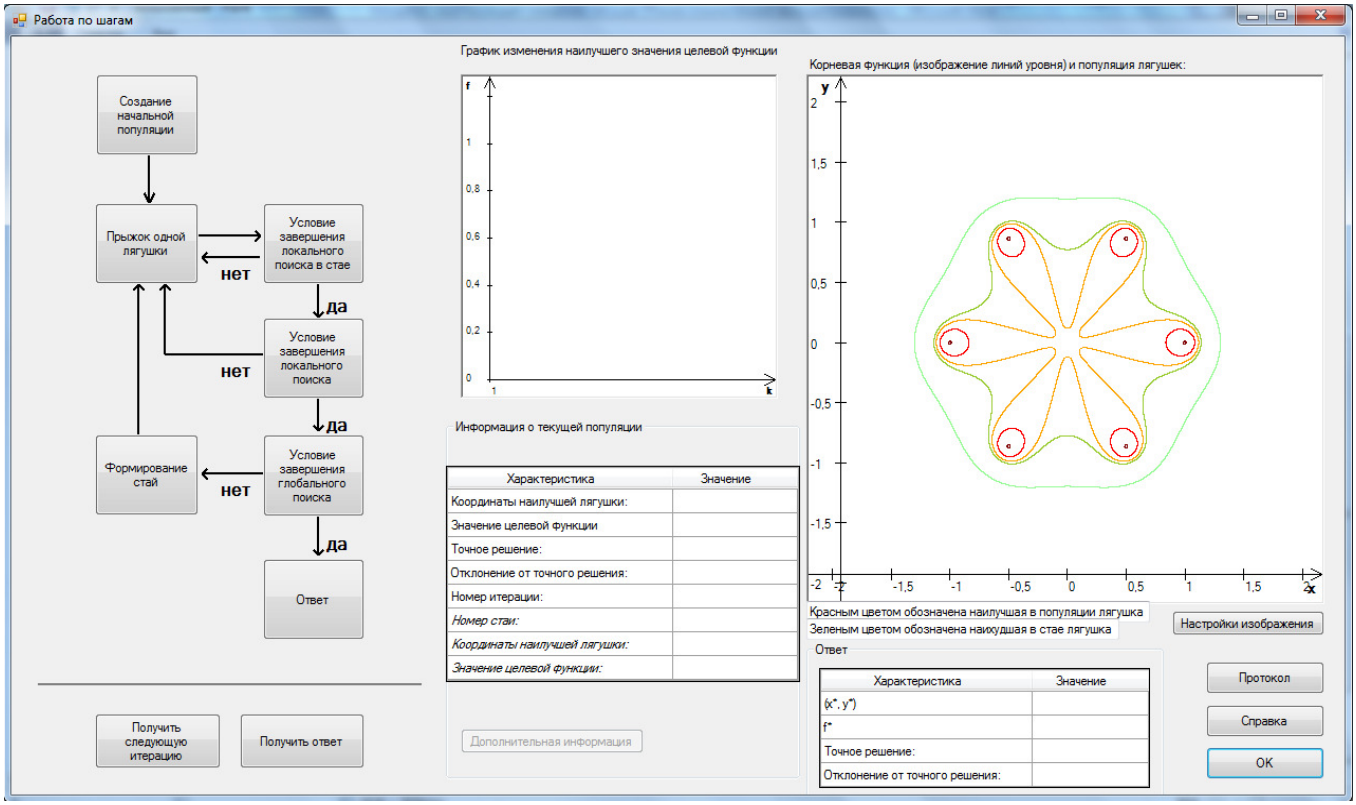

Рис. 4. Форма пошагового выполнения алгоритма

На форме изображена схема работы метода, отображаются результаты работы каждого шага, график наилучшего значения целевой функции, а также окончательный результат работы метода.

Тестовые примеры.

Пример 1. Рассмотрим работу программы на примере поиска глобального максимума корневой функции: $f(z)=\dfrac{1}{1+\left|z^6-1\right|}, z\in C, z=x+iy$. Зададим множество допустимых решений $x, y\in[-2;2]$. Глобальный максимум $f\left(x^*, y^*\right)=1$.

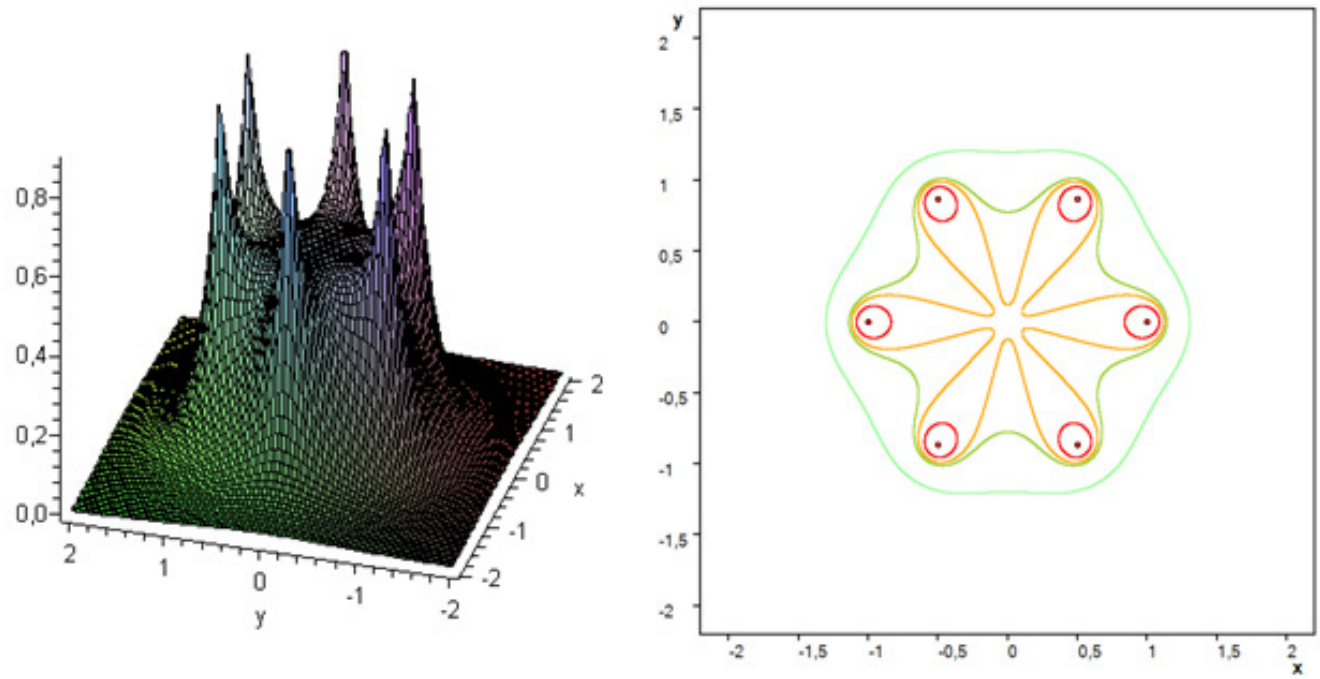

Рис. 5. Изображения поверхности и линий уровня корневой функции

Результаты работы методов Таблица 1

Название метода	Значения координат	Значение целевой функции
Метод, имитирующий поведение лягушек	$(x^*, y^*) = (0{,}5; 0{,}866)$	$f(x^*, y^*) = 1$
Метод, имитирующий распространение сорняков	$(x^*, y^*) = (0{,}5; -0{,}866)$	$f(x^*, y^*) = 1$
Метод, имитирующий поведение кукушек	$(x^*, y^*) = (-0{,}5008; 0{,}866)$	$f(x^*, y^*) = 0{,}9951$
Метод, имитирующий поведение светлячков	$(x^*, y^*) = (0{,}5006; -0{,}8657)$	$f(x^*, y^*) = 0{,}9957$

Пример 2. Рассмотрим работу программы на примере поиска глобального максимума функции Розенброка: $f(x, y) = -a\left(y - x^2\right)^2 - (1 - x)^2$. Зададим множество допустимых решений $x \in [-3; 3]$, $y \in [-1; 5]$. Глобальный максимум $f(x^*, y^*) = 0$.

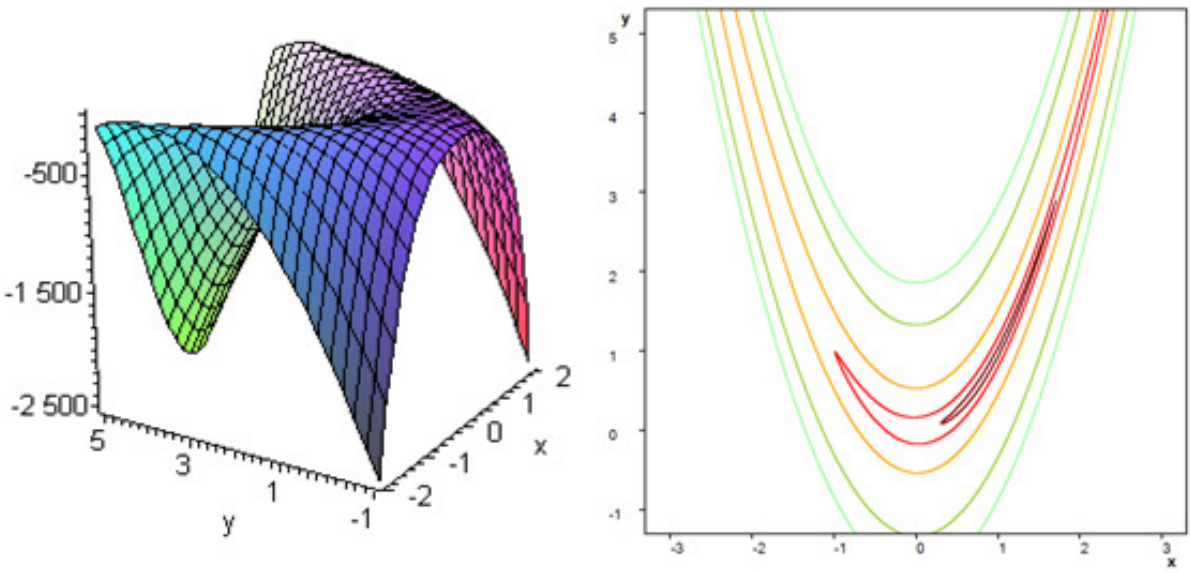

Рис. 6. Изображения поверхности и линий уровня функции Розенброка

Результаты работы методов Таблица 2

Название метода	Значения координат	Значение целевой функции
Метод, имитирующий поведение лягушек	$(x^*, y^*) = (1;1)$	$f(x^*, y^*) = 0$
Метод, имитирующий распространение сорняков	$(x^*, y^*) = (1,0001;1,0002)$	$f(x^*, y^*) = 0$
Метод, имитирующий поведение кукушек	$(x^*, y^*) = (0,9997;0,9995)$	$f(x^*, y^*) = 0$
Метод, имитирующий поведение светлячков	$(x^*, y^*) = (1,0144;1,0301)$	$f(x^*, y^*) = -0,0003$

Заключение

Основным результатом данной работы является создание алгоритмического и программного обеспечения для решения задач оптимизации многоэкстремальных функций. Продемонстрирована применимость всех четырех рассмотренных биоинспирированных методов для нахождения глобального максимума функций со сложным рельефом линий уровня.

Библиографический список

1. Eusuff, M.M., Lansey, K.E. Optimization of water distribution network design using the shuffled frog leaping algorithm // Journal of Water Resources Planning and Management, 2003, no. 3, pp. 210-225.

2. Elbeltagi, E., Hegazy, T., Grierson, D. Comparison among five evolutionary-based optimization algorithms // J. Advanced Engineering Informatics, 2005, no. 19, pp. 43–53.

3. Elbeltagi, E., Hegazy, T., Grierson, D. A modified shuffled frog-leaping optimization algorithm: applications to project management // Structure and Infrastructure Engineering, 2007, no 1, pp. 53–60.

4. Yang X.-S., Deb S. Cuckoo search via Levy flights // Proceedings of world congress on Nature & Biologically Inspired computing, 2009, pp. 210–214.

5. Yang X. S., Deb S. Engineering Optimization by Cuckoo Search // Int. J. Mathematical Modelling and Numerical Optimization, 2010, vol.1, no 4, pp. 330–340.

6. Valian E., Mohanna H., Tavakoli S. Improved Cuckoo search Algorithm for global optimization // International Journal of Communications and Information Technology, 2011, vol. 1, no. 1, pp. 31– 44.

7. Richard Inglis, Chris Taylor Numerical approximation of Levy Flight // Mathematics, http://math.stackexchange.com/questions/52869/numerical-approximation-of-levy-flight, 2011.

8. Andrew M. Edwards Using likelihood to test for Lévy flight search patterns and for general power-law distributions in nature // Journal of Animal Ecology, 2008, no. 77, pp. 1212 – 1222.

9. David W. Sims, Emily J. Southall, Nicolas E. Humphries Scaling laws of marine predator search behavior // NATURE, 28 February 2008, no. 451, pp. 1098–1102.

10. Mehrabian A. R., Lucas C. A novel numerical optimization algorithm inspired from weed colonization // Ecological Informatics, 2006, no. 1, pp. 355–366.

11. X.S. Yang Firefly algorithms for multimodal optimization // Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences. 2009, vol. 5792, pp. 169-178.

12. Гладков В.А., Курейчик В.В. Биоинспирированные методы в оптимизации.- М.: Физматлит, 2006. – 384 с.

13. Пантелеев А.В., Метлицкая Д.В., Алешина Е.А. Методы глобальной оптимизации. Метаэвристические стратегии и алгоритмы .– М.: Изд-во Вузовская книга, 2013. – 248 с.

14. Пантелеев А.В. Применение эволюционных методов глобальной оптимизации в задачах оптимального управления детерминированными системами.– М.: Изд-во МАИ, 2013. – 160 с.