

УДК 519.688

«Жадный» адаптивный метод случайного поиска условного глобального экстремума функций

О.В. Рязанцева

Научный руководитель доктор физико-математических наук

А.В. Пантелеев

Аннотация

В настоящее время при решении задач проектирования аэрокосмических систем и ракетно-космических комплексов достаточно широко применяются метаэвристические методы глобальной оптимизации [1]. Примером такого метода является «жадный» адаптивный алгоритм случайного поиска глобального экстремума [2], который рассмотрен в данной работе. Сформировано алгоритмическое и программное обеспечение метода, эффективность которого продемонстрирована на модельных примерах со сложной структурой линий уровня.

Ключевые слова:

глобальный экстремум; метаэвристические методы; мультистарт; локальный поиск.

Введение

Необходимость решения разнообразных задач оптимизации параметров конструкций современных самолетов и космических аппаратов предъявляет повышенные требования к используемым методам оптимизации. К основным проблемам применения классических алгоритмов следует отнести «застывание» в окрестности локальных минимумов, большая чувствительность к росту размерности задачи, проблемы с обеспечением сходимости. Поэтому актуальной проблемой является развитие метаэвристических методов [1], которые позволяют получать приемлемое с точки зрения проектировщика решения за разумное время. Критериями качества в задачах проектирования летательных аппаратов обычно являются вес, дальность полета, точность выполнения задачи, стоимость создания нового

образца, различные аэродинамические характеристики и т.д. Оптимизируемые параметры задаются проектировщиком одновременно с ограничениями на их значения исходя из цели проекта.

Рассматривается проблема поиска условного глобального экстремума функций многих переменных на множестве допустимых решений, где допустимые значения каждой из переменных заданы отрезком.

Метод использует идею мультистарта, т.е. многократного поиска решения при различных начальных значениях, при этом каждая итерация алгоритма включает в себя две фазы: фазу конструирования и фазу локального поиска. В результате первой фазы (фазы конструирования) порождаются решения «хорошего качества», из которых начинается вторая фаза – локального поиска. Затем полученные точки берутся в качестве начальных для первой фазы и процедура продолжается. Наилучшее из найденных решений принимается за приближённое решение поставленной задачи.

Авторами сформированы детальный алгоритм решения поставленной задачи и соответствующее программное обеспечение, которое предоставляет возможность выбора целевой функции, задания количества переменных, диапазона поиска и всех необходимых параметров метода. Табличное и графическое представления промежуточных и итоговых расчётов позволяют наглядно продемонстрировать процесс вычислений.

В работе приведено решение для нескольких типовых функций (в том числе многоэкстремальных со сложными рельефами линий уровня), для которых известно аналитическое решение. На данных примерах показано, что алгоритм позволяет найти решение с заданной точностью при различных параметрах метода.

Постановка задачи

Пусть X – множество допустимых решений, f – заданная на множестве X непрерывная целевая функция.

Требуется найти такую точку x^* , что

$$f(x^*) = \min_{x \in X} f(x), \quad (1)$$

где

$$X = \{x \mid a_i \leq x_i \leq b_i, i = 1, \dots, n\}, \quad (2)$$

т.е. требуется найти условный глобальный минимум целевой функции.

Стратегия поиска решения

Метод использует идею мултистарта, т.е. многократного поиска решения, где каждая итерация включает две фазы:

- фазу конструирования;
- фазу локального поиска.

Наилучшее из найденных решений принимается за приближённое решение поставленной задачи.

В результате первой фазы (фазы конструирования) порождаются решения «хорошего качества», из которых начинается вторая фаза – локального поиска. Затем полученные точки берутся в качестве начальных для первой фазы и процедура продолжается.

Изначально на **фазе конструирования** все координаты вектора решения считаются нефиксированными. На этой фазе производится локальный поиск по направлениям, параллельным координатным осям, т.е. изменяется только одна координата. В результате такого поиска формируется ограниченный список кандидатов – наиболее перспективных направлений, из которых случайным образом выбирается одно. Найденная в результате поиска по этому направлению точка берётся в качестве начальной для продолжения решения, использованное направление исключается из списка возможных направлений, соответствующее значение координаты решения фиксируется. Описанная последовательность действий продолжается до тех пор, пока не будут зафиксированы все координаты вектора решения x .

На **фазе локального поиска** точка x улучшается с помощью поиска в $3^n - 1$ возможных направлениях. При $n = 2$ их число равно 8 (см. рис.1).

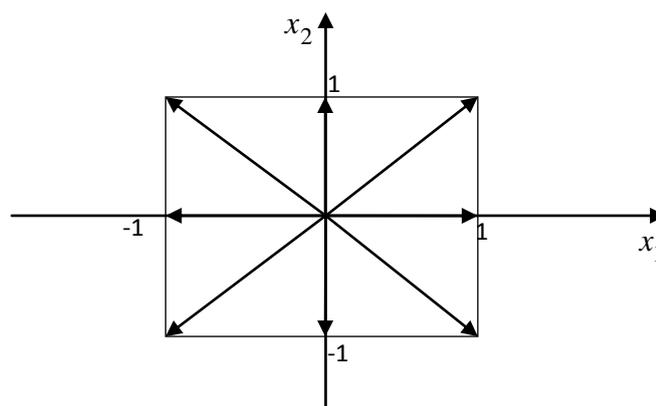


Рис.1. Направления поиска при $n = 2$

Возможные направления d задаются векторами $(1;0)^T$, $(0;1)^T$, $(-1;0)^T$, $(0;-1)^T$, $(1;1)^T$, $(-1;-1)^T$, $(1;-1)^T$, $(-1;1)^T$, координаты которых принадлежат множеству $\{-1,0,1\}$. Чтобы

определить вектор направления поиска для номера направления $r \in \{1, 2, \dots, 3^n - 1\}$, необходимо:

1) представить число r по основанию 3, т.е. в виде

$$r = r_k \cdot 3^k + \dots + r_2 \cdot 3^2 + r_1 \cdot 3^1 + r_0 \cdot 3^0. \quad (3)$$

Получить представление числа r по основанию 3 можно с помощью следующей процедуры:

- а) разделить r на 3, запомнить остаток;
 - б) частное от деления снова разделить на 3 и запомнить остаток; повторять шаг б) до тех пор, пока частное больше 3;
 - в) выписать ответ в виде: последнее частное, затем все полученные остатки в обратном порядке (начиная с последнего);
- 2) в представлении числа r по основанию 3 заменить цифру 2 на «-1».

Пример нахождения векторов возможных направлений поиска для $n = 2$ приведён в табл.1.

Таблица 1

| Представление r по основанию 3 | Вектор направления | | Представление r по основанию 3 | Вектор направления |
|----------------------------------|--------------------|--|----------------------------------|--------------------|
| 01 | $(0;1)^T$ | | 12 | $(1;-1)^T$ |
| 02 | $(0;-1)^T$ | | 20 | $(-1;0)^T$ |
| 10 | $(1;0)^T$ | | 21 | $(-1;1)^T$ |
| 11 | $(1;1)^T$ | | 22 | $(-1;-1)^T$ |

Так как количество направлений может быть достаточно большим, можно производить поиск только по количеству направлений, указанному пользователем. Это количество может быть меньше либо равно $3^n - 1$.

Алгоритм

Шаг 1. Задать N – максимальное количество итераций, M – максимальное количество итераций без улучшения значения целевой функции, S – максимальное количество стартов в мультистартовой процедуре, N_D – максимальное число исследуемых направлений, $\alpha \in [0;1]$ – параметр, n – количество координат целевой функции, $[a_i, b_i]$, $i = 1, \dots, n$ – промежутки изменения всех координат на множестве допустимых решений X .

Положить $f^{**} = +\infty$, $s = 1$ (число стартов).

Шаг 2. Генерировать начальную точку x с помощью равномерного распределения:

$x_i \sim R[a_i, b_i]$, $i = 1, \dots, n$. Положить $h = \frac{\max(b_i - a_i)}{8}$ (шаг сетки на множестве (2)), $m = 0$ (число

итераций без улучшения значения целевой функции), $j = 1$ (число итераций).

Шаг 3. Фаза конструирования.

Шаг 3.1. Сформировать множество $I = \{1, 2, \dots, n\}$ номеров координат вектора x .

Шаг 3.2. Положить $f_{\min} = +\infty$, $f_{\max} = -\infty$, $i = 1$ (номер координаты).

Шаг 3.3. Если $i \in I$, тогда перейти к шагу 3.4, иначе – к шагу 3.5.

Шаг 3.4. Решить задачу линейного поиска по координате i при фиксированных остальных координатах: вычислить значения целевой функции в узлах сетки $x_i^{(m)} = x_i + m \cdot h$, $m = 0, \pm 1, \pm 2, \dots$, причём $a_i \leq x_i^{(m)} \leq b_i$, значения остальных координат совпадают с соответствующими значениями вектора x . Среди полученных точек выбрать ту, которой соответствует наименьшее значение целевой функции. Обозначить её $z^{(i)}$, а соответствующее значение функции $g^{(i)}$.

Если $g^{(i)} > f_{\max}$, то положить $f_{\max} = g^{(i)}$.

Если $g^{(i)} < f_{\min}$, то положить $f_{\min} = g^{(i)}$.

Шаг 3.5. Если $i < n$, то положить $i = i + 1$ и перейти к шагу 3.3.

Шаг 3.6. Положить $RCL = \emptyset$ (ограниченный список кандидатов), $i = 1$.

Шаг 3.7. Если $i \in I$ и $g^{(i)} \leq (1 - \alpha) \cdot f_{\min} + \alpha \cdot f_{\max}$, то добавить номер i в RCL .

Шаг 3.8. Если $i < n$, то положить $i = i + 1$ и перейти к шагу 3.7.

Шаг 3.9. Выбрать случайным образом номер l из списка RCL , положить $x = z^{(l)}$, исключить число l из множества I .

Шаг 3.10. Если I не пусто, то перейти к шагу 3.2, иначе – к шагу 4.

Шаг 4. Фаза локального поиска.

Шаг 4.1. Положить $D = \emptyset$ (множество использованных номеров направлений поиска), $x^* = x$, $f^* = f(x)$, $N_d = \min\{3^n - 1, N_D\}$.

Шаг 4.2. Положить $B = 0$ (флаг улучшения).

Шаг 4.3. Если количество элементов в множестве D меньше или равно N_d и $B = 0$, то перейти к шагу 4.4, иначе – к шагу 4.8.

Шаг 4.4. Выбрать с помощью равномерного распределения $R[1;3^n - 1]$ номер направления r (если он принадлежит D , то генерировать заново), добавить номер r в множество D .

Шаг 4.5. Представить r по основанию 3 и, заменяя «2» на «-1» в этом представлении, получить соответствующий вектор направления поиска d . Вычислить $x = x^* + h \cdot d$.

Шаг 4.6. Если $a \leq x \leq b$ (покоординатно) и $f(x) < f^*$, то положить $x^* = x$, $f^* = f(x)$, $D = \emptyset$, $B = 1$.

Шаг 4.7. Перейти к шагу 4.3.

Шаг 4.8. Если $B = 1$, перейти к шагу 4.2, иначе – к шагу 5.

Шаг 5. Положить $x = x^*$. Если $f(x) < f^{**}$, то положить $x^{**} = x$, $f^{**} = f(x)$, $m = 0$, иначе: $m = m + 1$.

Шаг 6. Если $m > M$, то $h = \frac{h}{2}$, $m = 0$.

Шаг 7. Если $j < N$, то положить $j = j + 1$ и перейти к шагу 3.

Шаг 8. Если $s \leq S$, то положить $s = s + 1$ и перейти к шагу 2. Если $s > S$, то процедуру поиска завершить. Искомое приближённое решение: $x^{**}, f(x^{**})$.

Замечания.

1. Рекомендуемые значения параметров: $\alpha = 0,4$; $N_d = 30$; $M = 20$; $N = 200$; $S = 20$.
2. На шаге 3.4 проявляется «жадность» алгоритма, так как при первом же успехе (уменьшении функции) он переходит в «успешную» точку.
3. На шаге 3.9 введён элемент случайности, чтобы внести необходимое разнообразие на фазе конструирования.

Программное обеспечение. Анализ эффективности метода

Программная реализация метода выполнена с помощью Builder C++, MS Word 2007 и Htm2chm. Основным её назначением является:

- определение глобального минимума заданных целевых функций;
- сбор данных для анализа применимости, устойчивости и быстродействия алгоритма;
- демонстрация работы алгоритма в текстовом и графическом виде.

В функции программы входит:

- задание параметров метода;
- выбор целевой функции из предложенного списка;
- решение с выводом подробной информации в табличном виде;
- пошаговое решение;
- построение графиков: линий уровня с найденной точкой экстремума, значений целевой функции по стартам; эволюции наилучшего значения целевой функции по итогам реализованных стартов;
- предоставление подробного протокола решения;
- предоставление справочной информации.

Программа обладает интуитивно-понятным интерфейсом. При её разработке были учтены возможные неверные действия пользователя по вводу данных и установлены ограничения на длину и тип данных полей ввода, всплывающие подсказки, ограничения на доступ к кнопкам.

Интерфейс программы включает в себя три вкладки, каждая из которых отвечает за реализацию определённых функций.

На вкладке «Постановка задачи» (см. рис.2) пользователь имеет возможность:

- 1) ознакомиться с постановкой задачи;
- 2) выбрать целевую функцию;
- 3) задать границы области и параметры метода такие, как: число переменных (n),

максимальное количество итераций (N), максимальное количество итераций без улучшения значения целевой функции (M), максимальное количество стартов в мультистартовой процедуре (S), максимальное число исследуемых направлений (N_D) и параметр α .

Целевая функция выбирается из списка, при этом автоматически меняются рекомендуемые границы для каждой функции. Любой параметр можно поменять, включая количество переменных, т.е. имеется возможность решать задачи большой размерности. Всплывающие подсказки для каждого поля ввода указывают, какие значения может принимать соответствующий параметр.

После задания всех необходимых параметров можно нажать на кнопку «Решить» и автоматически перейти на вкладку «Решение».

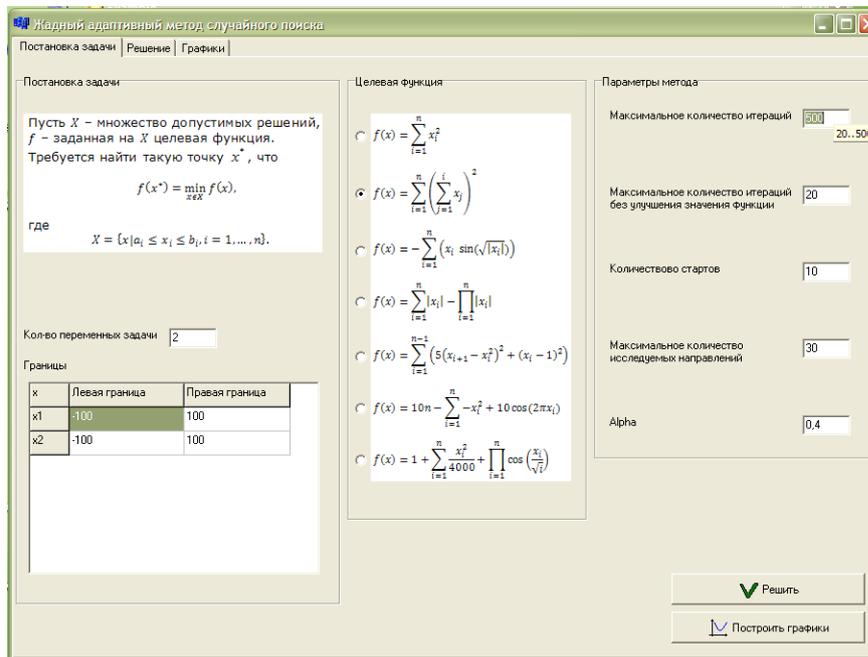


Рис.2. Вкладка «Постановка задачи»

На вкладке «Решение» (см. рис.3) расположены таблицы, в которые выводятся основные результаты, получаемые при решении задачи. Первая таблица содержит все итерации для каждого старта. Номер старта выбирается из выпадающего списка. В таблице фиксируются результаты фазы конструирования, локального поиска, признак улучшения значения целевой функции, а также шаг, с которым производится поиск (радиус поиска). Вторая таблица предназначена для отображения конечных результатов всех стартов. Она содержит начальную случайную точку, из которой стартует алгоритм, конечную точку – оптимальную на данном этапе, а также значение целевой функции в данной точке.

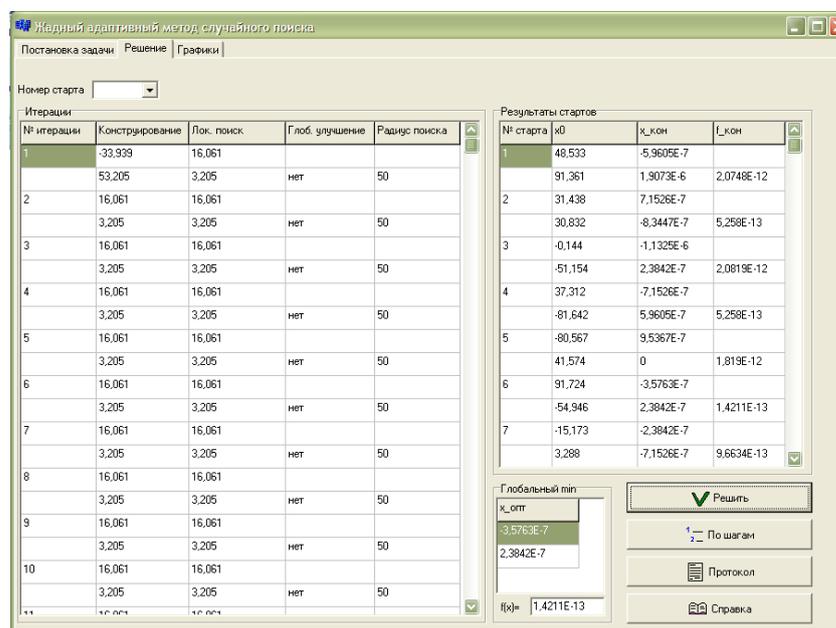


Рис.3. Вкладка «Решение»

Итоговый результат вычислений представлен в виде вектора координат и значения функции для этого вектора. Этот ответ является наиболее оптимальным из всех, которые были найдены за общее число стартов.

Все результаты вычислений отображены на графиках, которые расположены на вкладке «Графики» (см. рис.4). Первый график (слева) показывает линии уровня выбранной целевой функции, а также точку глобального минимума после решения задачи. Данный график отображается только в том случае, если размерность рассматриваемой задачи равна 2. Второй график (справа сверху) точками отображает оптимальные значения функции, полученные на разных стартах. По нему наглядно можно определить, на каком старте было получено наилучшее значение функции, то есть найдена искомая точка глобального минимума. Третий график отражает динамику изменения глобального минимума функции. Так как рассматривается задача на поиск минимума, очевидно, что данный график всегда будет представлять собой невозрастающую функцию, так как он строится по точкам глобальных минимумов, то есть меняет своё значение лишь тогда, когда полученное в конце очередного старта значение целевой функции меньше, чем уже имеющееся оптимальное значение функции.

Второй и третий графики строятся для задач любой размерности и являются наглядным изображением работы программы.

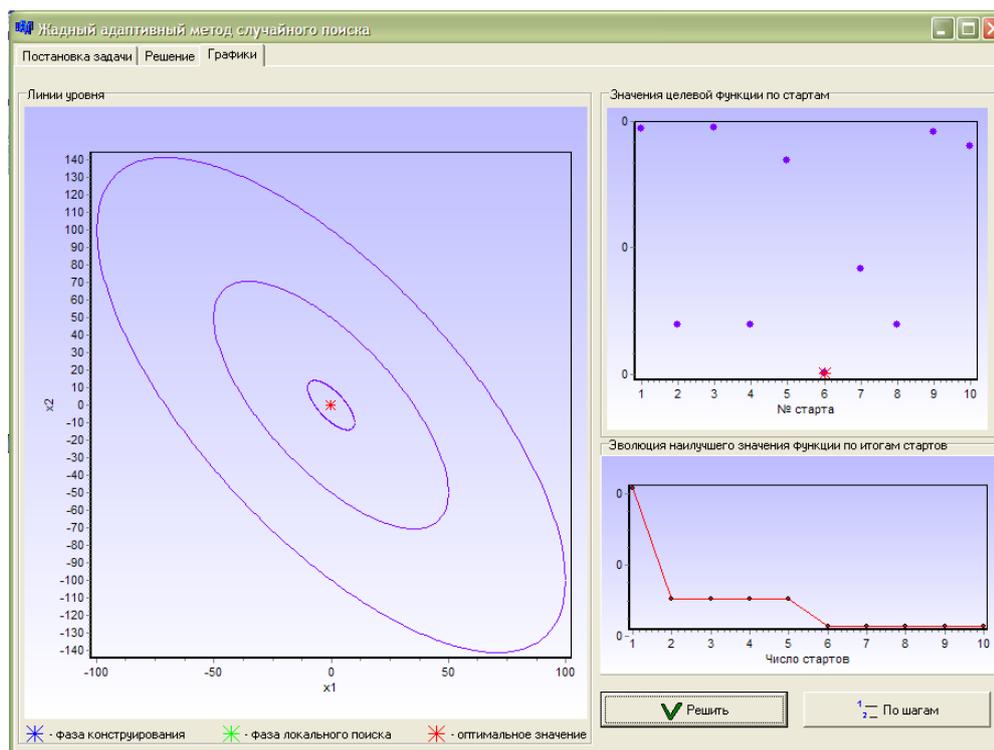


Рис.4. Вкладка «Графики».

При нажатии на кнопку «Решить» на первых двух вкладках реализуется поиск глобального минимума выбранной функции с заданными на вкладке «Постановка задачи» параметрами метода. Вся информация о процессе вычисления заносится в таблицы и отображается на графиках.

Кнопка «По шагам» имеет то же назначение, что и кнопка «Решить», но результаты вычислений выводятся не сразу, а постепенно, по шагам, как в алгоритме. Необходимые построения на графиках также отображаются динамически в процессе вычислений, что является наглядной реализацией решения. На каждом старте можно проследить, как происходит поиск (см. рис.5), а с помощью графиков значений функции отследить результаты всех стартов.

При нажатии на кнопку «Протокол» открывается протокол вычислений (см. рис.6), который включает полную информацию о каждом проведённом решении: номер решения, начальное значение функции, количество переменных и все остальные параметры метода (заданные на вкладке «Постановка задачи»), начальную точку для каждого старта, значения, полученные на каждой итерации каждого старта, итоговые значения по стартам, а также ответ – приближённое значение глобального минимума функции.

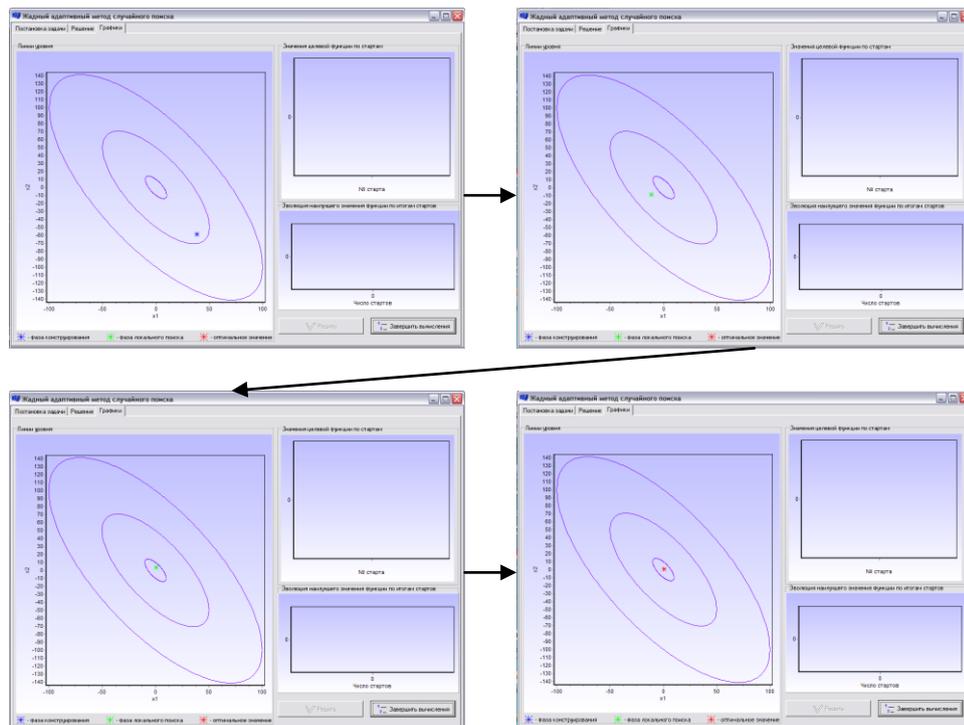


Рис.5. Пример пошагового выполнения на первом старте

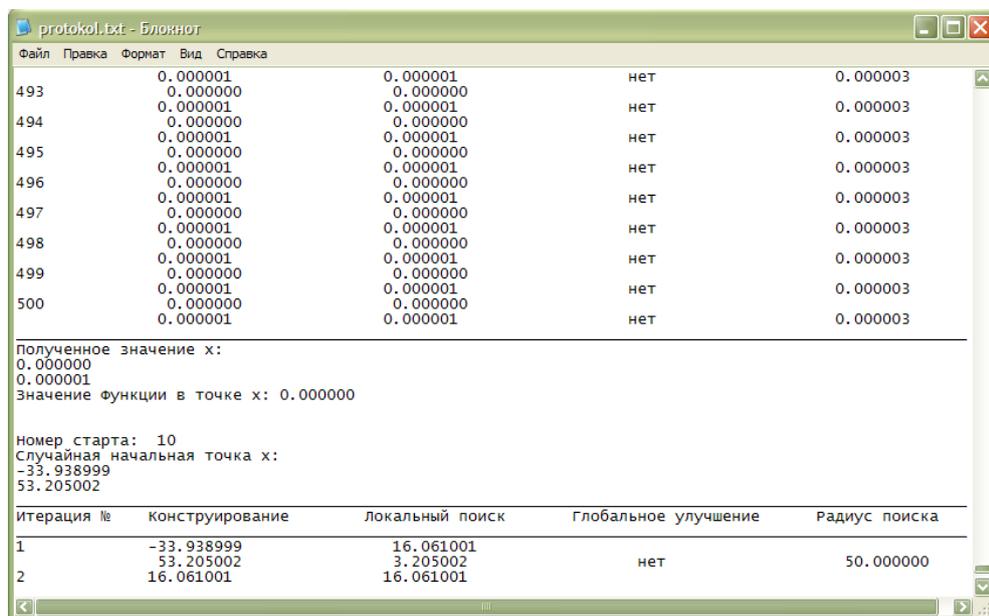


Рис.6. Протокол вычислений

При нажатии на кнопку «Справка» открывает файл справки (см. рис.7), которая предоставляет пользователю возможность ознакомиться с алгоритмом метода, а также даёт подробное описание самой программы.

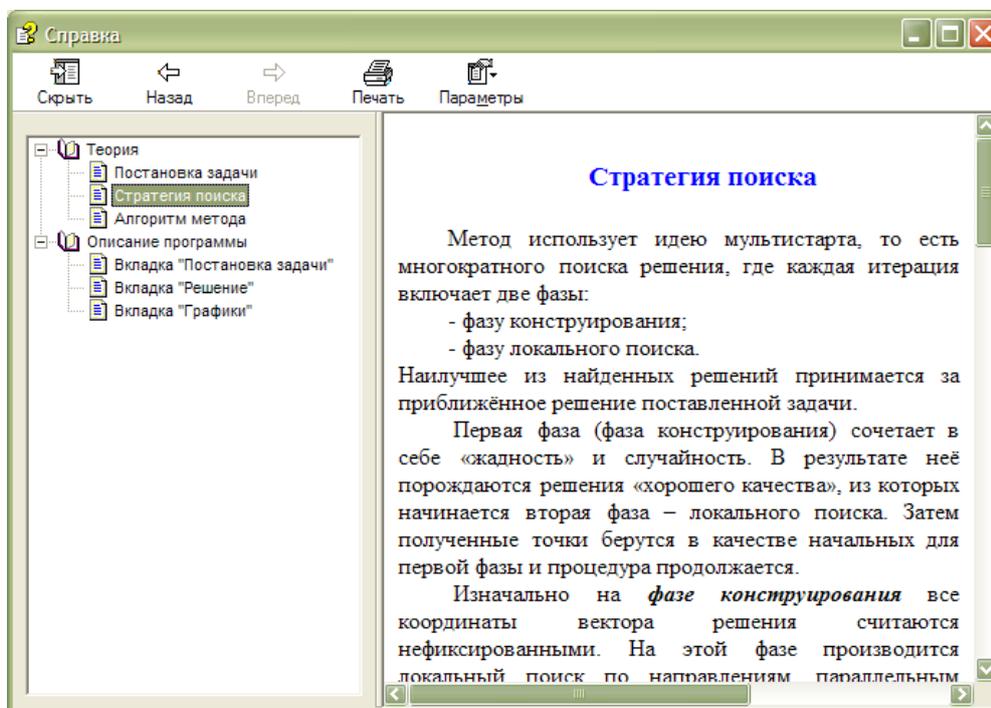


Рис.7. Окно справки

С помощью разработанной программы были получены результаты для минимизации различных тестовых функций. Основное влияние на результат, полученный с помощью программы, оказывают такие параметры, как количество итераций и количество стартов. Будем их изменять, оставляя неизменными значения остальных параметров: максимальное количество итераций без улучшения значения функции = 20, максимальное количество исследуемых направлений = 30, $\alpha = 0.4$, $n = 2$.

Пример 1. Целевая функция $f(x) = \sum_{i=1}^n x_i^2$, $a_i = -100, b_i = 100, i = 1..n$ (при $n = 2$ см.

рис. 8). Точное решение: $x_i^* = 0, i = 1..n, f(x^*) = 0$, а численное – приведено в табл. 2.

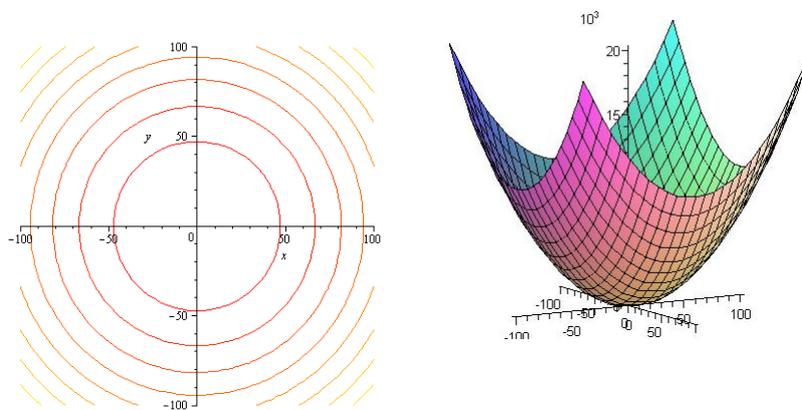


Рис. 8. Линии уровня и график функции $f(x) = \sum_{i=1}^2 x_i^2$

Таблица 2

| Кол-во итераций | Кол-во стартов | Полученное значение | | |
|-----------------|----------------|---------------------|-------------|------------|
| | | x_1 | x_2 | $f(x)$ |
| 100 | 10 | -0,371 | 0,351 | 0,26084 |
| 500 | 10 | 2,3842E-7 | 1,1921E-7 | 7,1054E-14 |
| 1000 | 10 | 0 | -7,1054E-15 | 5,0487E-29 |
| 500 | 50 | 1,1921E-7 | -1,1921E-7 | 2,8422E-14 |
| 500 | 100 | 0 | -1,1921E-7 | 1,4211E-14 |

Пример 2. Целевая функция $f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$, $a_i = -100, b_i = 100, i = 1..n$ (при $n = 2$

см. рис. 9). Аналитическое решение: $x_i^* = 0, i = 1..n, f(x^*) = 0$.

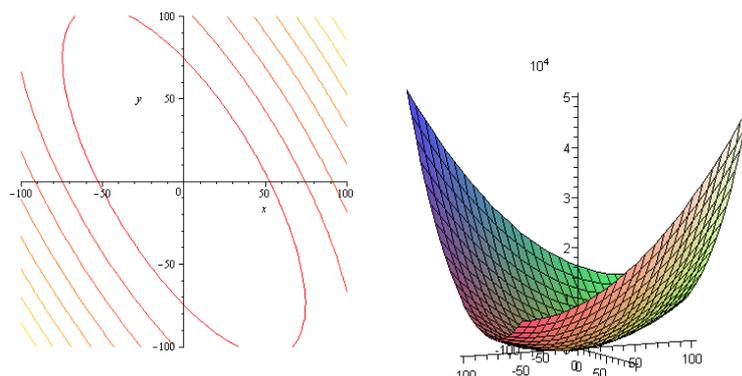


Рис. 9. Линии уровня и график функции $f(x) = \sum_{i=1}^2 (\sum_{j=1}^i x_j)^2$

Численное решение приведено в табл. 3.

Таблица 3

| Кол-во итераций | Кол-во стартов | Полученное значение | | |
|-----------------|----------------|---------------------|-------------|------------|
| | | x_1 | x_2 | $f(x)$ |
| 100 | 10 | -0,226 | 1,15 | 0,90486 |
| 500 | 10 | -3,5763E-7 | 1,1921E-7 | 1,8474E-13 |
| 1000 | 10 | 0 | -3,5527E-15 | 1,2622E-29 |
| 500 | 50 | 3,5763E-7 | -1,1921E-7 | 1,8474E-13 |
| 500 | 100 | 0 | 2,3842E-7 | 5,6843E-14 |

Пример 3. Целевая функция $f(x) = -\sum_{i=1}^n (x_i \sin(\sqrt{|x_i|}))$, $a_i = -500, b_i = 500, i = 1..n$ (при $n = 2$ см. рис. 10). Аналитическое решение: $x_i^* = 420,96, i = 1..n, f(x^*) = -837,97$.

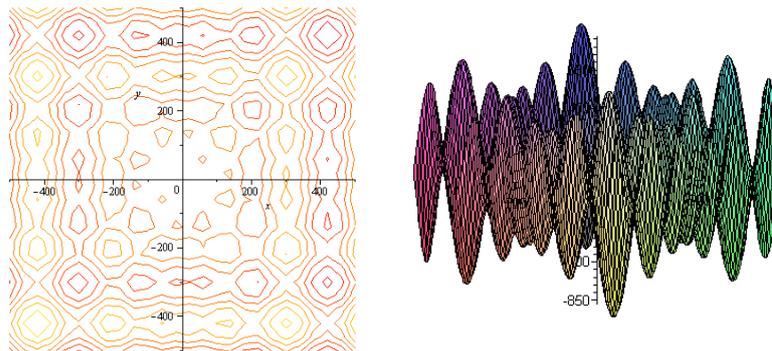


Рис. 10. Линии уровня и график функции $f(x) = -\sum_{i=1}^2 (x_i \sin(\sqrt{|x_i|}))$

Численное решение приведено в табл. 4.

Таблица 4

| Кол-во итераций | Кол-во стартов | Полученное значение | | |
|-----------------|----------------|---------------------|--------|---------|
| | | x_1 | x_2 | $f(x)$ |
| 100 | 10 | 421,1 | 420,42 | -837,93 |
| 500 | 10 | 420,98 | 420,97 | -837,97 |
| 1000 | 10 | 420,98 | 420,98 | -837,97 |
| 500 | 50 | 420,97 | 420,96 | -837,97 |
| 500 | 100 | 420,97 | 420,96 | -837,97 |

Пример 4. Целевая функция $f(x) = \sum_{i=1}^n |x_i| - \prod_{i=1}^n |x_i|$, $a_i = -10, b_i = 10, i = 1 \dots n$ (при $n = 2$ см. рис. 11). Аналитическое решение: $x_i^* = 0$, $i = 1 \dots n$, $f(x^*) = 0$. Численное решение приведено в табл. 5.

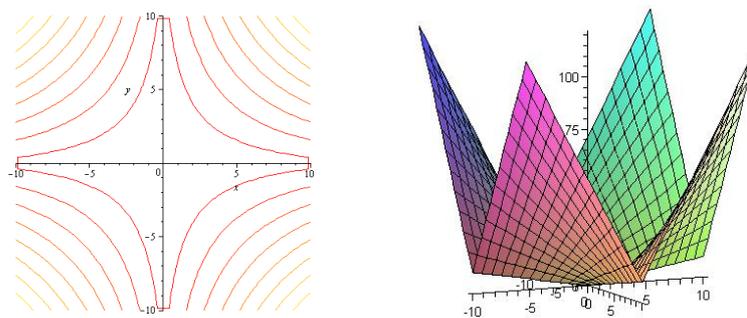


Рис. 11. Линии уровня и график функции $f(x) = \sum_{i=1}^2 |x_i| - \prod_{i=1}^2 |x_i|$

Таблица 5

| Кол-во итераций | Кол-во стартов | Полученное значение | | |
|-----------------|----------------|---------------------|---------|------------|
| | | x_1 | x_2 | $f(x)$ |
| 100 | 10 | 0,016 | -0,0975 | 0,11506 |
| 500 | 10 | 0 | 0 | 0 |
| 1000 | 10 | 3,5527E-15 | 0 | 3,5527E-15 |
| 500 | 50 | -2,9802E-8 | 0 | 2,9802E-8 |
| 500 | 100 | 0 | 0 | 0 |

Пример 5. Целевая функция $f(x) = \sum_{i=1}^{n-1} (5(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$, $a_i = -30, b_i = 30$, $i = 1 \dots n$ (при $n = 2$ см. рис. 12). Аналитическое решение: $x_i^* = 1$, $i = 1 \dots n$, $f(x^*) = 0$.

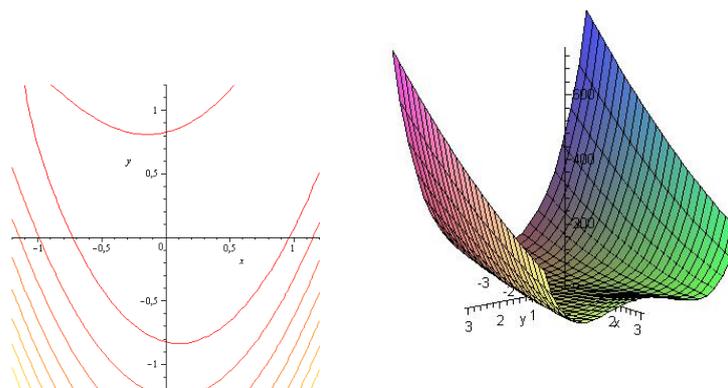


Рис. 12. Линии уровня и график функции $f(x) = 5(x_2 - x_1^2)^2 + (x_1 - 1)^2$
Численное решение приведено в табл. 6.

Таблица 6

| Кол-во итераций | Кол-во стартов | Полученное значение | | |
|-----------------|----------------|---------------------|-------|------------|
| | | x_1 | x_2 | $f(x)$ |
| 100 | 10 | 1,0145 | 0,597 | 0,93424 |
| 500 | 10 | 1 | 1 | 3,1974E-14 |
| 1000 | 10 | 1 | 1 | 0 |
| 500 | 50 | 1 | 1 | 1,4211E-14 |
| 500 | 100 | 1 | 1 | 0 |

Пример 6. Целевая функция $f(x) = 10n - \sum_{i=1}^n (-x_i^2 + 10 \cos(2\pi x_i))$, $a_i = -5.12, b_i = 5.12$, $i = 1..n$ (при $n = 2$ см. рис. 13). Аналитическое решение: $x_i^* = 0, i = 1..n, f(x^*) = 0$.

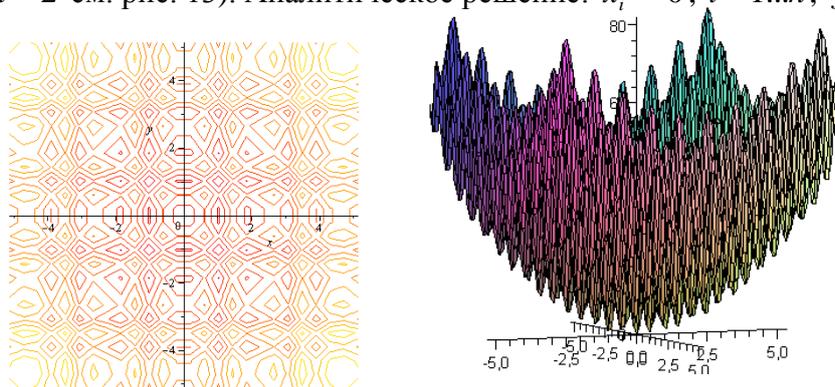


Рис. 13. Линии уровня и график функции $f(x) = 20 - \sum_{i=1}^2 (-x_i^2 + 10 \cos(2\pi x_i))$

Численное решение приведено в табл. 7.

Таблица 7

| Кол-во итераций | Кол-во стартов | Полученное значение | | |
|-----------------|----------------|---------------------|------------|---------|
| | | x_1 | x_2 | $f(x)$ |
| 100 | 10 | 0,025 | 0,022 | 0,21961 |
| 500 | 10 | -3,1114E-5 | 1,5087E-7 | 0 |
| 1000 | 10 | 1,2293E-7 | -6,2345E-5 | 0 |
| 500 | 50 | 3,1252E-5 | -1,0151E-7 | 0 |
| 500 | 100 | -1,5495E-5 | -3,099E-5 | 0 |

Пример 7. Целевая функция $f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} + \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$, $a_i = -600, b_i = 600$, $i = 1..n$ (при $n = 2$ см. рис. 14). Аналитическое решение: $x_i^* = 0, i = 1..n, f(x^*) = 0$.

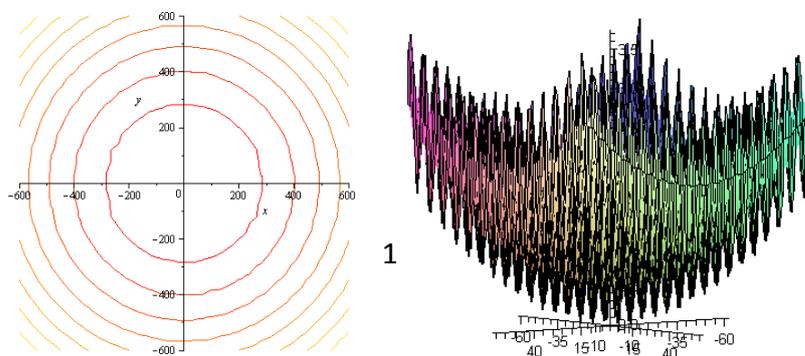


Рис. 14. Линии уровня и график функции $f(x) = 1 + \sum_{i=1}^2 \frac{x_i^2}{4000} + \prod_{i=1}^2 \cos\left(\frac{x_i}{\sqrt{i}}\right)$

Численное решение приведено в табл. 8.

Таблица 8

| Кол-во итераций | Кол-во стартов | Полученное значение | | |
|-----------------|----------------|---------------------|------------|------------|
| | | x_1 | x_2 | $f(x)$ |
| 100 | 10 | -3,416 | 4,163 | 0,06345 |
| 500 | 10 | 3,483E-6 | 0 | 7,5175E-15 |
| 1000 | 10 | 2,3283E-9 | -1,1642E-9 | 0 |
| 500 | 50 | 0 | 7,1526E-7 | 1,2794E-16 |
| 500 | 100 | 0 | -1,4305E-6 | 5,1164E-16 |

Анализируя полученные результаты, можно сделать вывод о том, что рассмотренный алгоритм применим для поиска глобального условного минимума функций многих переменных, так как он даёт результат, достаточно близкий к истинному, а с помощью изменения параметров метода этот результат можно ещё улучшить.

Заключение

В работе сформированы детальный алгоритм «жадного» адаптивного метода случайного поиска условного глобального экстремума и соответствующий комплекс программных средств. Их эффективность продемонстрирована при решении специальным образом сконструированных модельных примеров. Полученные результаты позволяют надеяться на успешное применение метода в задачах проектирования летательных аппаратов различных классов.

Библиографический список

1. Пантелеев А.В. Метаэвристические алгоритмы поиска глобального экстремума.- М.: Издательство МАИ-ПРИНТ, 2009.
2. M.J. Hirsch, C.N. Meneses, P.M. Pardalos, M.G.C. Resende Global optimization by continuous grasp // Optimization Letters, 2006.

Сведения об авторах

Рязанцева Ольга Вячеславовна; студентка Московского авиационного института
(государственного технического университета);
e-mail: ryazantseva_olga@mail.ru.